TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Systémy automatického rozpoznávania reči pre aplikácie s detskými používateľmi

Diplomová práca

2024 **Bc. Richard Ševc**

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Systémy automatického rozpoznávania reči pre aplikácie s detskými používateľmi

Diplomová práca

Študijný program:	Počítačové siete
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra elektroniky a multimediálnych telekomunikácií
Školiteľ:	doc. Ing. Stanislav Ondáš, PhD.

2024 Košice **Bc. Richard Ševc**

Abstrakt

Táto práca sa zaoberá špecifikami a výzvami rozpoznávania detskej reči v slovenskom jazyku pomocou automatického rozpoznávania reči (ASR). Sústredí sa na adaptáciu a vylepšenie modelu XLS-R prostredníctvom augmentačných metód a trénovacích stratégií, aby bol model schopný efektívnejšie rozpoznávať detskú reč. Práca rozširuje existujúci dataset detskej reči v slovenskom jazyku o nové nahrávky a porovnáva rôzne prístupy k trénovaniu s cieľom zvýšiť presnosť modelu. Výsledný model je následne implementovaný vo webovej aplikácii, ktorá umožňuje jednoduchú transkripciu detskej reči v slovenčine.

Kľúčové slova

Automatické rozpoznanie reči, detská reč, model wav2vec 2.0, model XLS-R, dataset detskej reči v slovenčine, dátová augmentácia, jazykový model, webová aplikácia pre tvorbu transkripcie

Abstract

This work deals with the specifics and challenges of recognizing children's speech in Slovak language using automatic speech recognition (ASR). It focuses on the adaptation and improvement of the XLS-R model through augmentation methods and training strategies to make the model capable of recognizing children's speech more efficiently. The work extends the existing dataset of children's speech in Slovak language with new recordings and compares different training approaches in order to improve the accuracy of the model. The resulting model is then implemented in a web application that enables straightforward transcription of children's speech in Slovak.

Key words

Automatic speech recognition, children's speech, wav2vec 2.0 model, XLS-R model, children's speech dataset in Slovak, data augmentation, language model, web application for transcription

72416

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra elektroniky a multimediálnych telekomunikácií

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**

Študijný program: **Počítačové siete**

Názov práce:

Systémy automatického rozpoznávania reči pre aplikácie s detskými používateľmi

Automatic speech recognition systems for applications with child users

Študent: **Bc. Richard Ševc**

Školiteľ: **doc. Ing. Stanislav Ondáš, PhD.**

Školiace pracovisko: **Katedra elektroniky a multimediálnych telekomunikácií**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Naštudovať problematiku automatického rozpoznávania reči a trénovania modelov pre tieto systémy.
2. Rozšíriť databázu detskej reči v slovenčine vyhľadaním, spracovaním a prepisom nahrávok detskej reči.
3. Pripraviť rozšírenú databázu detskej reči na trénovanie a natrénovať modely pre systém automatického rozpoznávania reči pre detských používateľov.
4. Vypracovať záverečnú prácu s opisom teoretickej a praktickej časti práce podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 19.04.2024

Dátum zadania diplomovej práce: 31.10.2023

v. z. .....

prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 19. apríla 2024 .....

vlastnoručný podpis

Poďakovanie

Chcem sa poďakovať svojmu vedúcemu práce doc. Ing. Stanislavovi Ondášovi, PhD. za odborné vedenie, pomoc a užitočné pripomienky pri riešení diplomovej práce. Taktiež by som chcel poďakovať všetkým výskumníkom, ktorí riešia podobnú problematiku a takto rozširujú študijné materiály. Nakoniec by som sa chcel poďakovať rodine a blízkym za podporu počas celého môjho štúdia.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam symbolov a skratiek

ASR – Automatic Speech Recognition

CNN – Connectionist Temporal Classification

CTC – Connectionist Temporal Classification

GPU – Graphics Processing Unit

LM – Language Model

NLP – Natural Language Processing

VLTN – Vocal Tract Length Normalization

WER – Word Error Rate

Úvod

Automatické rozpoznávanie reči (ASR) predstavuje v súčasnosti jednu z kľúčových technológií v rámci spracovania prirodzeného jazyka. Jeho aplikácie nachádzajú uplatnenie v rôznych oblastiach. Napriek rozsiahlemu pokroku v technológiách ASR sa väčšina výskumov zameriava na rozpoznávanie reči dospelých. Reč detí, ktorá sa vyznačuje špecifickými charakteristikami ako sú vyššia tónová výška a nestálosť, zostáva často prehliadaná. Táto práca sa preto zameriava na špecifiká rozpoznávania detskej reči a snahu adaptovať existujúce modely ASR tak, aby boli schopné efektívnejšie rozpoznávať detskú reč v slovenskom jazyku.

V teoretickej časti práce je vysvetlený prehľad modelu wav2vec 2.0 a jeho predtrénovaných verzií XLSR-53 a XLS-R, ktoré boli špeciálne vyvinuté na zlepšenie rozpoznávania reči naprieč rôznymi jazykmi. Významná časť praktickej časti bola venovaná adaptácii a dotrénovaniu modelu XLS-R s cieľom dosiahnutia čo najlepšej presnosti pri rozpoznávaní detskej reči v slovenčine.

V rámci práce sú taktiež popísané a využité augmentačné metódy, ktoré v predošlom výskume preukázali pozitívne výsledky na zlepšenie presnosti rozpoznávania detskej reči v slovenčine pri aplikácií na trénovacie dáta pre trénovanie XLS-R modelu.

Jedným z hlavných cieľov práce je rozšírenie existujúceho datasetu detskej slovenskej reči, ktorý zahŕňa nahrávky z televíznej relácie Táraninky o nahrávky z nového zdroja, konkrétne z rádiovej relácie Rozhlasové Leporelo určenej pre deti. Toto rozšírenie umožní pokryť širšie spektrum výslovností a intonácií v detskej reči a zvýšiť tak diverzitu trénovacích dát a vytvorenie komplexnejšieho datasetu detskej reči v slovenčine.

Experimentálna časť práce sa zameriava na aplikáciu a porovnanie rôznych augmentačných a trénovacích stratégií pri dotrénovaní modelu XLS-R. Následne sú analyzované výsledky dotrénovaných modelov bez a s využitím jazykového modelu. Najlepší model je po analýze výsledkov implementovaný v rámci webovej aplikácie s intuitívnym používateľským rozhraním, ktorá umožňuje jednoduchú transkripciu detskej slovenskej reči pomocou mikrofónu alebo nahraním zvukovej nahrávky.

1. Automatické rozpoznávanie reči

Automatické rozpoznávanie reči (ASR) je technológia, ktorá umožňuje strojom identifikovať ľudskú reč a previesť ju na čitateľný text. Podstatou systémov ASR je umožnenie interakcie medzi prirodzenou ľudskou komunikáciou a digitálnym svetom, čo uľahčuje interakcie, ktoré sú intuitívnejšie a efektívnejšie ako tradičné metódy zadávania údajov, ako je písanie na klávesnici. Tieto systémy sú kľúčové v rôznych aplikáciách, od hlasom riadených inteligentných zariadeniach a pomôcok až po podporu automatickej transkripcie a asistenčných technológií pre osoby, ktoré nemôžu používať bežné vstupné zariadenia. [1]

Hlavnou výzvou v oblasti ASR systémov je presná interpretácia ľudskej reči, ktorá sa u jednotlivých ľudí a v rôznych kontextoch značne líši. K zložitosti rozpoznávania reči prispievajú faktory ako prízvuk, dialekty, rýchlosť reči a výslovnosť. Túto úlohu ďalej komplikuje okolité prostredie, ktoré prináša premenné, ako je šum v pozadí a ozvena, ktoré môžu ovplyvniť výkonnosť systému. [2]

V posledných rokoch došlo v oblasti automatického rozpoznávania reči k zásadnému pokroku, pričom sa prechádza od tradičných hybridných modelov závislých na oddelených komponentoch pre akustické, výslovnostné a jazykové spracovanie k modernejším a integrovanejším end-to-end systémom. End-to-end modely, ako napríklad wav2vec 2.0, spracovávajú zvukový záznam priamo, prepisujúc ho do textu prostredníctvom hlbokých neurónových sietí. Tento prístup nielenže zjednodušuje pipeline ASR tým, že eliminuje potrebu viacerých odlišných komponentov, ale tiež zlepšuje schopnosť modelu efektívne sa učiť z obrovského množstva neanotovaných dát, ktoré sú dostupnejšie než anotované dáta. Takéto modely predstavujú významný pokrok v technológii ASR, zdôrazňujúc efektivitu a využitie dát na zlepšenie presnosti a dostupnosti technológií rozpoznávania reči. [3] [4]

1.1. Výzvy rozpoznávania detskej reči

V minulosti sa väčšina výskumu a vývoja ASR zameriavala na reč dospelých. Reč dospelých je v porovnaní s rečou detí jednotnejšia z hľadiska výšky tónu, rýchlosti a artikulácie. V dôsledku toho je väčšina existujúcich systémov ASR optimalizovaná pre dospelých používateľov a využíva veľké datasety obsahujúce reč dospelých na trénovanie modelov, ktoré dokážu rozpoznávať a prepisovať hovorené slová s vysokou presnosťou. Toto zameranie na reč dospelých však zanechalo v tejto oblasti významnú medzeru: rozpoznávanie detskej reči. Detská reč sa od reči dospelých líši v niekoľkých kľúčových aspektoch [2]:

- **Výška a tón:** detské hlasy majú vo všeobecnosti vyššiu výšku a v porovnaní s dospelými môžu v rámci jedného výroku viac kolísať.
- **Výslovnosť a artikulácia:** Ich reč je často menej presná, takže je premenlivejšia a niekedy ťažšie zrozumiteľná.
- **Štruktúra reči a správanie:** Deti môžu mať menej predvídateľnú rečovú štruktúru (angl. speech pattern) vrátane páuz, váhania a neštandardného používania gramatiky, čo odráža ich vývojové štádium.

Navyše od rozdielnosti je zber a anotácia dát detskej reči náročnou úlohou v porovnaní s dátami dospelej reči, ktoré možno získať z rôznych zdrojov ako sú filmy, spravodajské relácie, audioknihy, internet a iné. Aj keď sa detská reč dá z takýchto zdrojov získať, zabezpečenie presných anotácií je oveľa náročnejšie oproti anotácií reči dospelých. Na základe týchto výziev sú datasety obsahujúce dospelú reč oveľa početnejšie a objemnejšie ako datasety detskej reči.

1.2. Stratégie experimentov na zlepšenie rozpoznávania detskej reči

V posledných rokoch sa na zvýšenie účinnosti automatizovaných systémov na rozpoznávanie detskej reči využíva celý rad stratégií [5]. Väčšina týchto stratégií sa zameriava na využívanie rôznych techník na rozširovanie dát, čím sa rozširuje súbor dostupných trénovacích dát. Techniky, ako je rozšírenie založené na prevode textu na reč (TTS), ako bolo pôvodne navrhované v niektorých štúdiách [6] [7], kde sa modely automatického rozpoznávania reči zdokonaľujú pomocou umelo vytvorených dát, zatiaľ nepriniesli podstatné zlepšenie presnosti rozpoznávania detskej reči. Medzi ďalšie významné metódy augmentácií patria perturbácia dĺžky hlasového traktu (VTLP) [8], normalizácia vlastností základnej frekvencie [9], augmentácia dát dospelej reči prostredníctvom stochastického mapovania vlastností (SFM) [10], a augmentácie založené na spracovaní dát [11] ako perturbácia rýchlosti, perturbácia výšky tónu, perturbácia tempa, perturbácia hlasitosti, perturbácia reverberácie a perturbácia spektra. Zvýšenie výkonu ASR bolo pozorované aj pri využití augmentácie spektrogramu (SpecAugment) [12] [13], čo zdôrazňuje jej potenciál. [14]

1.3. Hodnotiaca metrika WER

Miera chybovosti slov (WER) je najčastejšie používanou metrikou v oblasti rozpoznávania reči a spracovania prirodzeného jazyka. Slúži ako kvantitatívny nástroj na posúdenie presnosti ASR systému pomocou porovnania predikovanej transkripcie v porovnaní s referenčnou transkripciou rovnakej zvukovej nahrávky. Podstata WER spočíva v schopnosti merať presnosť systémov rozpoznávania reči výpočtom podielu chýb, ktoré sa vyskytli v rámci transkripcie. [15]

Hodnota WER sa určuje porovnávaním vygenerovanej transkripcie (hypotézy) s referenčnou transkripciou, pričom sa identifikujú a kvantifikujú tri typy chýb:

- **Substitúcia** (angl. substitution - **S**) sa vyskytuje vtedy, keď je slovo v hypotéze nesprávne, ale zaujíma správnu pozíciu vo vzťahu k iným slovám.
- **Vynechanie** (angl. deletion - **D**) sa identifikuje vtedy, keď v hypotéze chýba slovo z referenčného textu.
- **Vloženie** (angl. insertion - **I**) nastane, keď hypotéza obsahuje ďalšie slová, ktoré sa v referenčnom texte nenachádzajú.

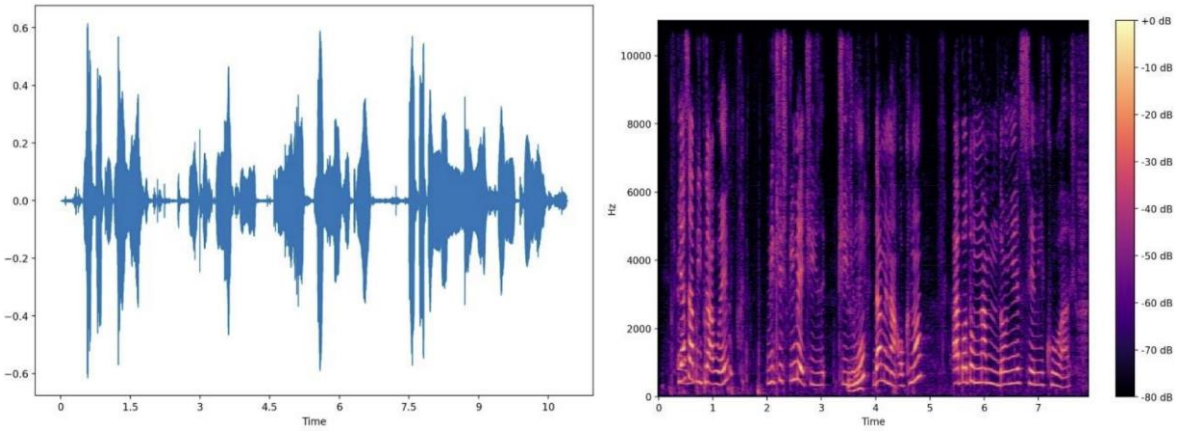
Vzorec na výpočet hodnoty WER je nasledujúci:

$$WER = \frac{S + D + I}{Počet\ slov\ v\ referencii}$$

2. Použité metódy augmentácie zvukových dát

Augmentácia zvukových dát je kľúčovou technikou v oblasti strojového učenia, najmä pri spracovaní a porozumení zvukových signálov. Jej hlavným cieľom je umelo zvýšiť rozmanitosť trénovacích datasetov zavedením rôznych foriem perturbácií alebo transformácií. To pomáha nielen zlepšiť odolnosť modelov voči šumu a variabilite dát z reálneho sveta, ale aj zabrániť nadmernému preučeniu sa, čím sa zvyšujú zovšeobecňujúce schopnosti modelov.

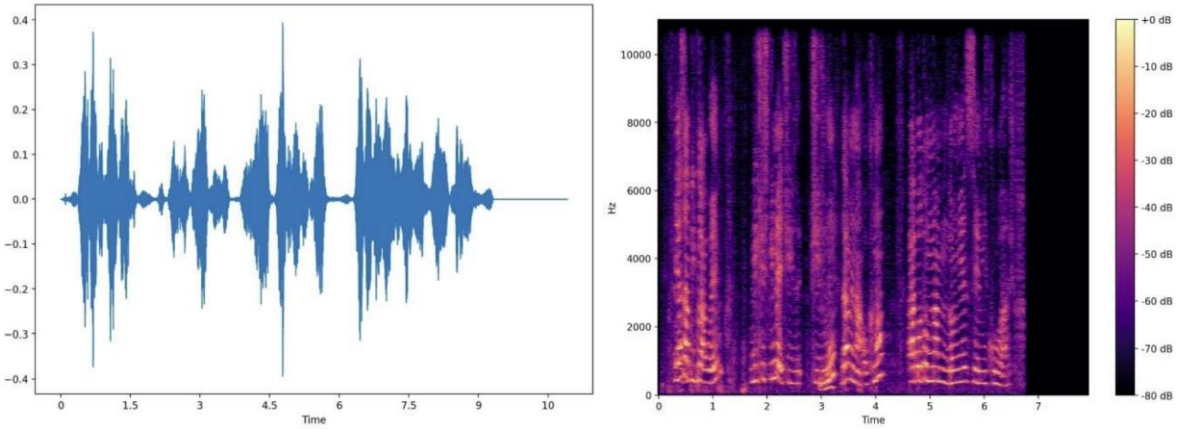
Na základe výsledkov experimentov s augmentačnými metódami predošlej práce [16] na podobných dátach bolo rozhodnuté, že v rámci práce boli vykonané experimenty len s augmentáciami, ktoré dosiahli najlepšie výsledky a to perturbácia rýchlosti a metóda SpecAugment. Pre porovnanie vplyvov jednotlivých augmentácií Obr. 1 zobrazuje časový priebeh a spektrogram originálnej neaugmentovanej nahrávky.



Obr. 1 Časový priebeh a spektrogram originálnej zvukovej nahrávky [16]

2.1.1. Perturbácia rýchlosti

Perturbácia rýchlosti je jednoduchá, ale účinná technika rozšírenia dát používaná pri spracovaní zvuku. Táto metóda zahŕňa zmenu rýchlosti prehrávania zvukového klipu, čo následne mení výšku a trvanie zvukového signálu. Zrýchlenie prehrávania má za následok kratší zvukový klip s vyšším tónom, zatiaľ čo spomalenie prehrávania predlžuje trvanie a znižuje výšku zvukového signálu. Perturbácia rýchlosti sa vo veľkej miere používa v systémoch rozpoznávania reči, aby boli odolnejšie voči zmenám v rýchlosti a výške tónu. Tréningom modelov na zvukových dátach, ktoré boli zrýchlené aj spomalené, sa model dokáže lepšie vysporiadať s rôznymi rýchlosťami reči a tónmi v reálnych scenároch. Časový priebeh a spektrogram nahrávky zrýchlenej pomocou využitia perturbácie rýchlosti zobrazuje Obr. 2.



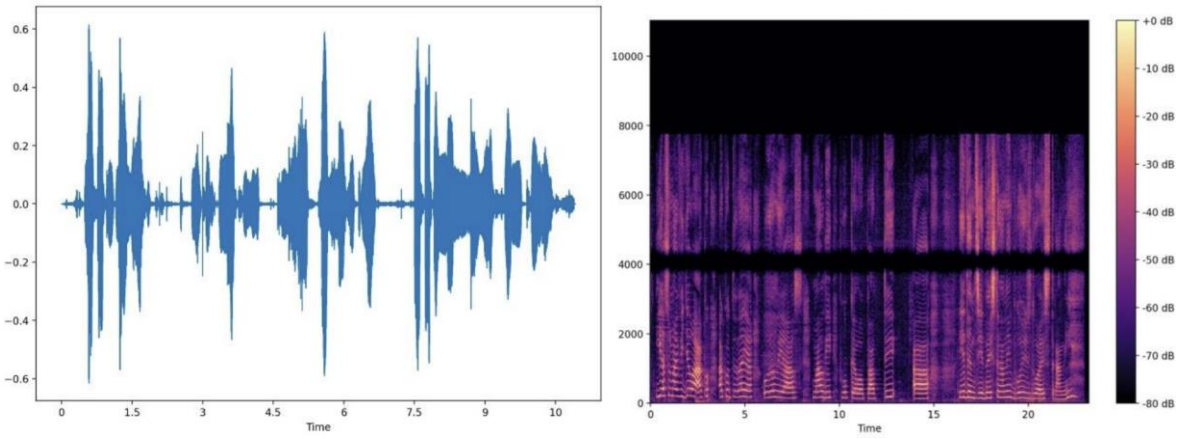
Obr. 2 Časový priebeh a spektrogram nahrávky po zrýchlení pomocou perturbácie rýchlosti [16]

2.1.2. Metóda SpecAugment

SpecAugment je efektívna metóda rozšírenia zvukových dát, ktorá zvyšuje robustnosť a zovšeobecňujúce schopnosti modelov strojového učenia, najmä v oblasti automatického rozpoznávania reči. Táto metóda modifikuje spektrogram zvukových signálov použitím špecifických masiek na časové a frekvenčné dimenzie. Týmto spôsobom vnáša do dát variabilitu, čo pomáha modelom dobre fungovať aj v rôznorodých a zašumených podmienkach reálneho sveta. [12] Metóda SpecAugment použitá v rámci práce pomocou knižnice Transformers od spoločnosti Hugging Face podporuje nasledujúce operácie:

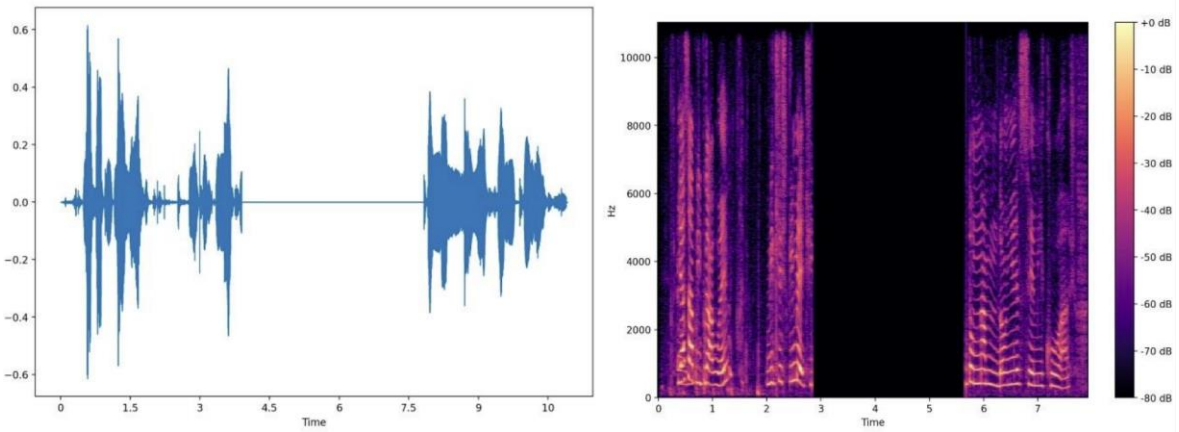
- Frekvenčné maskovanie (angl. frequency masking)
- Časové maskovanie (angl. time masking)

Frekvenčné maskovanie maskuje úsek f po sebe nasledujúcich mel frekvenčných kanálov, označený ako $[f_0, f_0 + f)$. Hodnota f sa vyberá z rovnomernej distribúcie v rozsahu od 0 do maximálneho parametra frekvenčnej masky F. Počiatočný frekvenčný kanál f_0 sa potom vyberá náhodne z rozsahu intervalu $[0, \nu - f)$, kde ν predstavuje celkový počet mel frekvenčných kanálov. Táto metóda pomáha zabezpečiť, aby sa model ASR nadmerne nespoliehal na špecifické frekvenčné pásma. [12] Časový priebeh a spektrogram nahrávky po frekvenčnom maskovaní zobrazuje Obr. 3.



Obr. 3 Časový priebeh a spektrogram nahrávky po frekvenčnom maskovaní [16]

Časové maskovanie maskuje t po sebe nasledujúcich časových krokov, reprezentovaných ako $[t_0, t_0 + t)$, kde hodnota t je vybraná z rovnomernej distribúcie od 0 po maximálny parameter časového maskovania T. Počiatočný časový krok, t_0 sa náhodne vyberie z intervalu [0, $\tau - t$), kde τ označuje celkový počet časových krokov v spektrograme. Časové maskovanie núti model využívať menej časových informácií a využívať širší kontext reči, vďaka čomu je efektívnejší pri riešení scenárov, v ktorých môžu časti zvuku obsahovať šum alebo chýbať. [12] Časový priebeh a spektrogram nahrávky po časovom maskovaní zobrazuje Obr. 4.



Obr. 4 Časový priebeh a spektrogram nahrávky po časovom maskovaní [16]

3. Prehľad modelu wav2vec 2.0

Model wav2vec 2.0 (ďalej len wav2vec2), ktorý vyvinula spoločnosť Facebook AI, predstavuje významný pokrok v oblasti rozpoznávania reči. Stavia na úspechu svojho predchodcu, modelu wav2vec a prináša nový prístup k učeniu z neupravených zvukových dát (angl. raw audio files) bez anotácií prostredníctvom samo-kontrolovaného učenia (angl. self-supervised learning). Na rozdiel od tradičných modelov rozpoznávania reči, ktoré sa vo veľkej miere spoliehajú na anotované dáta a ručne vytvorené funkcie, wav2vec2 vie využívať veľké množstvo neanotovaných zvukových dát rôznych jazykov na naučenie sa reprezentácií, ktoré sú užitočné pre úlohy rozpoznávania reči. Takýto spôsob učenia napodobňuje proces osvojovania si prirodzeného jazyka u ľudí. Je to podobné učeniu sa detí, ktoré si neustálym počúvaním dospelých okolo seba osvojujú danú reč bez explicitných inštrukcií. [3]

Samo-kontrolované učenie vo wav2vec2: Jadrom modelu wav2vec2 je koncept samo-kontrolovaného učenia, ktorý umožňuje modelu učiť sa priamo z dát bez nutnosti využitia anotovaných dát. Dosahuje sa to trénovaním modelu, aby predpovedal časti audio vstupu, ktoré ešte nevidel, na základe častí, ktoré už videl. Konkrétne, wav2vec2 maskuje časti audio vstupu a trénuje model predpovedať maskované časti z nemaskovaného kontextu. Tento prístup umožňuje modelu učiť sa bohaté reprezentácie zvukových vlastností, ktoré zachytávajú lingvistický obsah, charakteristiky hovoriaceho a iné detaily ľudskej reči. [3]

Výhody pre oblasť rozpoznávania reči: Schopnosť učiť sa z neanotovaných dát rieši jednu z hlavných výziev v oblasti rozpoznávania reči: nedostatok anotovaných dát, najmä pre nedostatočne zastúpené jazyky, dialekty a v prípade tejto práce pre slovenskú detskú reč. Schopnosť modelu wav2vec2 učiť sa z neanotovaných dát predstavuje významný pokrok vpred a ponúka škálovateľné riešenia na rozpoznávanie reči, ktoré sú efektívne a zároveň prispôsobiteľné jazykovej rozmanitosti ľudskej reči. [3]

Je dôležité zdôrazniť, že napriek tomu, že sa model wav2vec2 dokáže učiť veľmi všeobecné a prispôsobivé reprezentácie z neanotovaných dát, jeho doladenie (angl. fine-tuning) na anotovaných dátach špecifických pre danú úlohu je kľúčové pre optimalizáciu jeho výkonu. Doladenie umožňuje prispôsobiť model na špecifické požiadavky úlohy, ako sú tvorba transkripcie, rozpoznávanie hlasových príkazov alebo iná aplikácia. Tento proces je rozhodujúci pre maximalizáciu presnosti modelu a umožňuje mu lepšie zvládnuť konkrétne charakteristiky úlohy. [3]

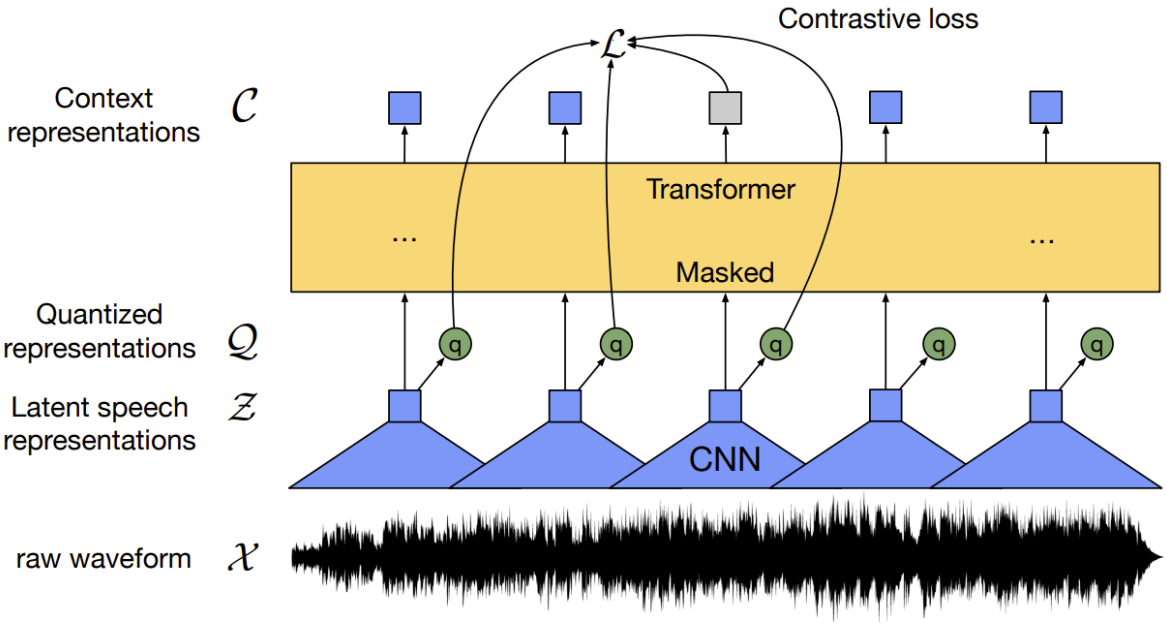
3.1. Architektúra wav2vec2 modelu

Model wav2vec2 má sofistikovanú architektúru, ktorá umožňuje efektívne spracovanie neupravených zvukových dát na extrakciu zmysluplných funkcií na rozpoznávanie reči. Jadro modelu tvoria tri hlavné komponenty: kódovač príznakov, kontextová sieť a kvantizačný modul. Architektúru modelu wav2vec2 pre trénovanie na neanotovaných dátach zobrazuje Obr. 5.

Kódovač príznakov (angl. feature encoder), $f: X \rightarrow Z$: Tento kódovač spracováva zvukové vstupy a prevádza ich na sériu latentných reprezentácií $z_1, \ldots, z_T$, pre T časových krokov. Tieto reprezentácie zachytávajú akustické vlastnosti zvukového signálu, ktoré ešte nie sú kontextualizované. Kódovač využíva na spracovanie zvukového signálu sériu konvolučných vrstiev, normalizáciu a aktivačné funkcie. Výstup Z pozostáva zo spojitých, latentných rečových reprezentácií. [3]

Kontextová sieť (angl. context network), $g: Z \rightarrow C$: Latentné reprezentácie Z vytvorené kódovačom príznakov sa potom odovzdajú kontextovej sieti, ktorá je založená na architektúre transformátora (angl. transformer). [17] Tento krok je kľúčový pre vytvorenie kontextových reprezentácií $c_1, \ldots, c_T$, ktoré zachytávajú informácie celej sekvencie zvukových dát. Kontextová sieť modeluje závislosti celej sekvencie latentných reprezentácií, pričom využíva mechanizmy vlastnej pozornosti (angl. self-attention) na vytvorenie komplexného porozumenia hovoreného jazyka v jeho kontexte. [3]

Kvantizačný modul (angl. quantization module), $Z \rightarrow Q$: Pre dosiahnutie cieľa samo-kontrolovaného učenia sa výstup kódovača príznakov diskretizuje na kvantizované reprezentácie Q pomocou kvantizačného modulu. Tento modul aplikuje kvantizáciu na latentné reprezentácie Z, čím ich transformuje na konečnú množinu diskrétnych jednotiek. Tieto kvantizované reprezentácie slúžia ako ciele v rámci samo-kontrolovaného učenia, kde sa model učí predpovedať tieto diskrétne verzie zvukového vstupu. [3]



Obr. 5 Architektúra modelu wav2vec2 pre trénovanie na neanotovaných dátach [3]

3.2. Predtrénovanie wav2vec2 modelu

Predtrénovanie (angl. pre-training) modelu wav2vec2 sa vykonáva prostredníctvom nového prístupu, ktorý čerpá inšpiráciu z modelovania maskovaného jazyka, podobne ako technika používaná v BERT. [18] Tento proces zahŕňa maskovanie určitého podielu časových krokov v latentných reprezentáciách vytvorených kódovačom príznakov a následné trénovanie modelu, aby identifikoval správnu kvantizovanú latentnú zvukovú reprezentáciu z množiny distraktorov pre každý maskovaný časový krok. Po predtrénovaní na neanotovaných dátach sa model podrobí doladeniu pomocou anotovaných dát, aby sa jeho schopnosti prispôsobili konkrétnym úlohám rozpoznávania reči. [3]

3.2.1. Maskovacia stratégia

Model maskuje časť výstupov z kódovača príznakov a nahrádza ich natrénovaným vektorom príznakov, ktorý je konzistentný vo všetkých maskovaných časových krokoch. Tento prístup nemení vstupy do kvantizačného modulu. Maskovanie sa realizuje náhodným výberom určitej časti p všetkých časových krokov, ktoré slúžia ako počiatočné indexy, z ktorých sa maskuje M po sebe nasledujúcich časových krokov. Tieto maskované úseky sa môžu prekrývať, čím sa zabezpečí komplexné pokrytie zvukovej sekvencie na robustné trénovanie. [3]

3.2.2. Ciele predtrénovania

Hlavným cieľom počas predtrénovania je riešenie kontrastívnej úlohy (angl. contrastive task) L_m, ktorej cieľom je porovnať výstup kontextovej siete pre maskovaný časový krok s jej skutočnou kvantifikovanou latentnou rečovou reprezentáciou v rámci množiny distraktorov. Túto

úlohu dopĺňa diverzná strata L_d, na podporu rovnomerného používania položiek kódovej knihy (angl. codebook). Celková strata je vyjadrená ako [3]:

$$L = L_m + \alpha L_d$$

kde α je hyperparameter určený ladením.

Kontrastívna strata (angl. contrastive loss): Pre daný maskovaný časový krok t musí model identifikovať skutočnú kvantizovanú reprezentáciu q_t z množiny $K+1$ kvantizovaných reprezentácií kandidátov, ktorá zahŕňa q_t a K distraktorov vybraných z iných maskovaných časových krokov v rámci tej istej výpovede. [19] Výpočet kontrastívnej straty je vyjadrený ako [3]:

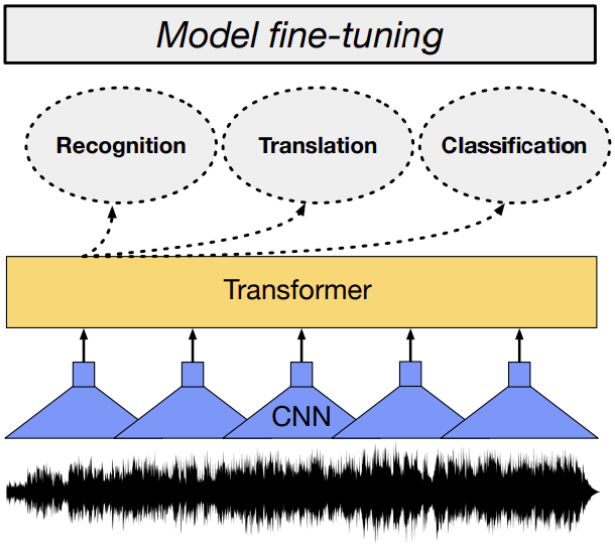
$$L_m = -log \frac{\exp(sim(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \sim Q_t} \exp(sim(c_t, \tilde{q})/\kappa)}$$

kde sa vypočítava kosínusová podobnosť (angl. cosine similarity) medzi kontextovou reprezentáciou a každou kandidátskou kvantizovanou reprezentáciou, normalizovanou parametrom teploty κ. To podporuje model, aby sa naučil rozlišovať správnu reprezentáciu od distraktorov. [3]

Diverzná strata (angl. diversity loss): Diverzná strata L_d je navrhnutá tak, aby sa zabezpečilo, že model bude rovnomerne využívať všetky záznamy kvantizovaných kódových kníh. Toto je dôležité pre povzbudenie modelu, aby sa nadmerne neprispôsoboval len na určitú podmnožinu položiek kódových kníh, ale namiesto toho sa naučil distribuovať ich používanie v celom rozsahu, čo vedie k robustnejším reprezentáciám. [3]

3.3. Doladenie predtrénovaných modelov

Doladenie (resp. dotrénovanie) je kritickou fázou pri aplikácii vopred natrénovaných modelov na konkrétne úlohy. V prípade modelu XLS-R, ktorý znázorňuje Obr. 6 je možné doladiť na rôzne jazykové úlohy ako je rozpoznávanie reči, preklad a klasifikácia zvuku. V kontexte rozpoznávania reči sa dolaďovaním prispôsobuje všeobecný model, ktorý je vopred natrénovaný na rozsiahlom korpuse neanotovaných zvukových dát, aby presne fungoval na konkrétnom datasete. [3]



Obr. 6 Možnosti doladenia XLS-R modelu [4]

Lineárna projekčná vrstva: Prvý krok v procese dolaďovania pre rozpoznávanie reči zahŕňa rozšírenie modelu o lineárnu projekčnú vrstvu (angl. linear projection layer). Táto vrstva je náhodne inicializovaná a slúži ako mechanizmus mapovania z kontextových reprezentácií vytvorených kontextovou sieťou modelu na súbor tried, ktoré zodpovedajú slovníku špecifického pre danú úlohu. Napríklad pre dataset LibriSpeech je slovník reprezentovaný 29 znakovými tokenmi spolu s ďalším tokenom označujúcim hranice slov. [3]

Stratová funkcia CTC: Model je optimalizovaný pomocou stratovej funkcie CTC (Connectionist Temporal Classification), ktorá je vhodná najmä pre úlohy, pri ktorých nie je vopred známe zarovnanie medzi vstupnými zvukovými dátami a výstupnou transkripciou. Strata CTC umožňuje modelu naučiť sa toto zarovnanie implicitne, čím poskytuje robustný spôsob na trénovanie modelov rozpoznávania reči na sekvenčných dátach. [3]

3.3.1. Využitie stratovej funkcie CTC v modeloch rozpoznávania reči

Strata CTC je základnou zložkou pri trénovaní modelov neurónových sietí pre úlohy sekvenčného učenia, najmä pri rozpoznávaní reči. Rieši úlohu zosúladenia vstupných sekvencií s príslušnými výstupnými sekvenciami bez potreby vopred definovanej segmentácie. CTC zavádza jedinečný prístup začlenením špeciálnej “prázdnej“ značky, ktorá uľahčuje prácu s premenlivými dĺžkami sekvencií a spájanie opakujúcich sa znakov vo výstupe. [20]

Strata CTC funguje tak, že sa vypočíta pravdepodobnosť cieľovej sekvencie vzhľadom na výstupy modelu vo všetkých možných zarovnaniach, čo účinne umožňuje modelu naučiť sa optimálne mapovanie. Základná matematická formulácia CTC je vyjadrená ako [20]:

$$L_{CTC} = -\log P(Y \mid X)$$

kde $P(Y \mid X)$ je pravdepodobnosť cieľovej sekvencie Y vzhľadom na vstupnú sekvenciu X, sčítaná cez všetky možné zarovnania.

Implementácia straty CTC umožňuje komplexné doladenie modelu wav2vec2, čím sa zvyšuje jeho schopnosť presne transkribovať zvukové nahrávky bez manuálneho označovania alebo segmentácie dát. Jej úloha v procese doladenia zabezpečuje, aby sa model naučil nielen priame mapovanie zo zvuku na text a zároveň, aby sa prispôsobil aj zložitosti a premenlivosti hovoreného jazyka, čo je nevyhnutné na dosiahnutie vysokej presnosti v reálnych aplikáciách. [20]

3.4. Prehľad modelov XLSR-53 a XLS-R

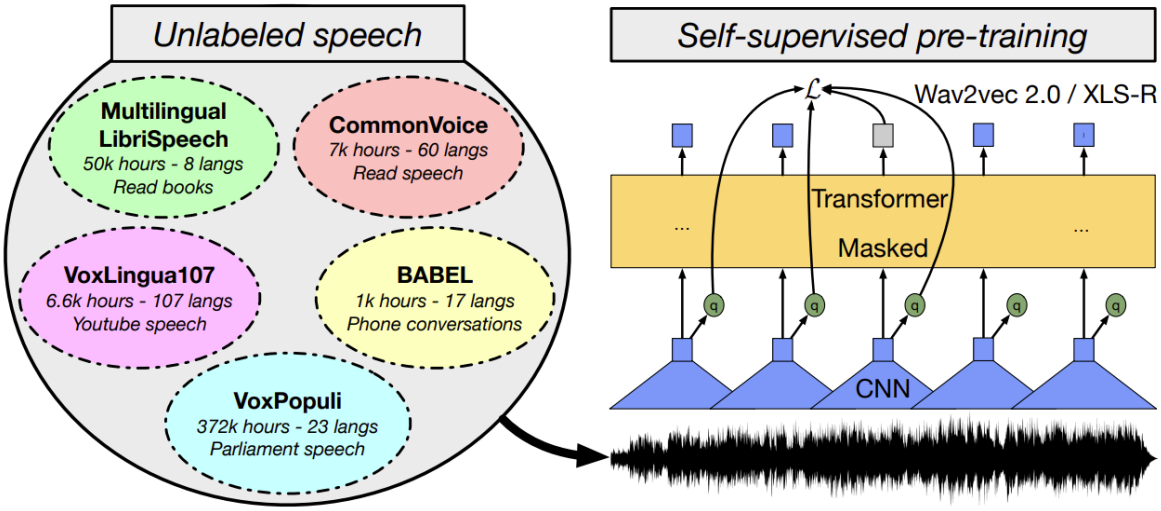
Táto kapitola popisuje dva významné modely rozpoznávania reči založené na architektúre wav2vec2 označené ako XLSR-53 a XLS-R. Označenia pochádzajú z názvu "Medzijazyková reprezentácia reči" (angl. Cross-Lingual Speech Representation). Tieto modely znamenali výrazný pokrok v oblasti zvyšovania presnosti a účinnosti rozpoznávania reči ako aj prekladu a klasifikácie v rôznych jazykoch a podmienkach. [4]

3.4.1. Trénovacie datasety

V predchádzajúcich výskumoch autori predtrénovali modely pomocou rozsiahlej kompilácie verejne dostupných rečových dát v celkovom počte 58 tisíc hodín pre model XLSR-53 a 436 tisíc hodín pre model XLS-R. Rečové dáta použité na predtrénovanie modelu XLS-R pochádzajú z rôznych veľkých datasetov:

- **VoxPopuli (VP-400K)** obsahuje celkovo 372 tisíc hodín dát v 23 európskych jazykoch parlamentných prejavov z Európskeho parlamentu. To z neho robí najväčší verejne dostupný rečový korpus. [21]
- **Multilingual Librispeech (MLS)** obsahuje približne 50 tisíc hodín dát v 8 európskych jazykoch. Väčšinu údajov tvorí angličtina (44 tisíc hodín). [22]
- **CommonVoice (CV)** je korpus čítanej reči. V rámci trénovaní bolo použité vydanie z decembra 2020(v6.1). Dataset obsahuje viac ako 7 tisíc hodín zvukového záznamu reči v 60 jazykov, v rozsahu viac ako 1,6 tisíc hodín pre angličtinu po menej ako 1 hodinu pre jazyky ako je hindčina. [23]
- **VoxLingua107 (VL)** je dataset obsahujúci 6,6 tisíc hodín dát v 107 jazykoch získaných z YouTube obsahu, s priemerom 62 hodín údajov na jazyk. [24]
- **BABEL (BBL)** je viacjazyčný korpus konverzačnej telefonickej reči obsahujúci približne tisíc hodín údajov v 17 afrických a ázijských jazykoch. [25]

Proces samo-kontrolovaného predtrénovania XLS-R modelu na neanotovaných dátach rôznych jazykov zobrazuje Obr. 7.



Obr. 7 Predtrénovanie XLS-R modelu na neanotovaných dátach [4]

3.4.2. Architektúra modelov

V rámci výskumu [4] v roku 2021 boli porovnávané modely XLSR-53 a XLS-R, ktoré využívajú architektúru wav2vec2. Napriek spoločnému základu sa ich vnútorné konfigurácie a trénovacie dáta značne líšia, čo vedie k rozdielom v ich výkonnosti. Porovnanie ich konfigurácie a trénovacích datasetov zobrazuje Tab. 1.

Tab. 1 Porovnanie architektúry modelov [4]

Model	#jaz	Trénovacie datasety	B	H_m	H_{ff}	A	#param
XLSR-53	53	MLS, CV, BBL	24	1024	4096	16	317M
XLS-R (0.3B)	128	VP-400K, MLS, CV, VL, BBL	24	1024	4096	16	317M
XLS-R (1B)	128	VP-400K, MLS, CV, VL, BBL	48	1024	4096	16	965M
XLS-R (2B)	128	VP-400K, MLS, CV, VL, BBL	48	1920	7680	16	2162M

Poznámka: V tabuľke vyššie "#jaz" predstavuje počet jazykov v rámci trénovacích datasetov. "B" označuje počet blokov transformátora, "H_m" označuje počet skrytých stavov, "H_{ff}" dimenziu feed-forward blokov, "A" označuje počet hláv pozornosti a "#param" označuje celkový počet parametrov modelu. [4]

3.4.3. Porovnanie výkonu modelov na datasete BABEL

Dataset BABEL predstavuje najnáročnejšie testovacie prostredie pre rozpoznávanie reči v rámci vybraných porovnávacích testov vykonaných v rámci výskumu [4], čo dokazuje jeho zvýšená

chybovosť slov (WER). Dataset BABEL sa vyznačuje obsahom jazykov s nízkym počtom zdrojov a prítomnosťou značného šumu podobného prirodzeným telefonickým rozhovorom. Tento dataset bol predmetom mnohých súťaží v oblasti počítačovej lingvistiky. [4]

Pri porovnaní s najlepšími uvádzanými výsledkami vo výskume [4] bola vytvorená Tab. 2, ktorá uvádza, že model XLS-R (0,3B) prekonáva výkonnosť modelu XLSR-53 vo všetkých jazykoch, pričom znižuje hodnotu WER v priemere o 1,4%. Napríklad v ásámčine sa WER znížil zo 44,1% na 42,9%; vo svahilčine z 26,5% na 24,3% a v gruzínčine z 31,1% na 28,0%. Oba modely XLSR-53 aj XLS-R boli predtrénované na dátach BABEL, ale lepšie výsledky modelu XLS-R (0.3B) poukazujú na výhody predtrénovania na ďalších externých datasetoch, ako je VoxPopuli. [4]

Tab. 2 Výsledky rozpoznávania reči na datasete BABEL z hľadiska WER [4]

Jazyk	as	tl	sw	lo	ka
Anotované dáta	55h	76h	30h	59h	46h
XLSR-53	44,1	33,2	26,5	-	31,1
XLS-R (0.3B)	42,9	33,2	24,3	31,7	28,0
XLS-R (1B)	40,4	30,6	21,2	30,1	25,1
XLS-R (2B)	39,0	29,3	21,0	29,7	24,3

Poznámka: V tabuľke vyššie "as" predstavuje ásámčinu, "tl" predstavuje tagalčinu, "sw" predstavuje svahilčinu, "lo" predstavuje laoštinu a "ka" predstavuje gruzínčinu.

Využitie väčšej veľkosti modelu XLS-R (1B) dosahuje ešte nižšiu priemernú hodnotu WER, čím prekonáva XLS-R (0,3B) o 2,5% WER. V prípade gruzínčiny to zodpovedá zlepšeniu hodnoty WER o 6% oproti modelu XLSR-53. Napokon, XLS-R (2B) pokračuje v trende zlepšovania a prekonáva XLS-R (1B) v priemere o 0,8% WER, čo slúži ako dôkaz, že zvýšenie kapacity modelu môže výrazne zvýšiť výkon. [4]

4. Dataset detskej reči v slovenčine

Táto kapitola analyzuje existujúci dataset detskej reči v slovenčine a popisuje jeho rozšírenie zberom dát a augmentačnými metódami.

4.1. Existujúci dataset detskej reči v slovenčine

Existujúci dataset detskej reči v slovenčine obsahuje transkribované nahrávky slovenskej reči detí vo veku 5 až 8 rokov z televíznej relácie Táraninky. Tento dataset bol postupne rozširovaný v štyroch rôznych prácach študentov Technickej univerzity v Košiciach. [26] [27] [28] [16] Dataset celkovo obsahuje 1667 výrokov a bol vytvorený z 88 častí relácie Táraninky. Celkovo boli zaznamenané výroky 103 unikátnych detských rečníkov, z čoho 43 je mužských a 60 ženských. Z celkového počtu výrokov bolo v rámci práce [16] vytvorené rozdelenie dát s pomerom 3:1 pre trénovaciu a testovaciu množinu. Trénovacia množina tohto datasetu po rozdelení obsahuje 1278 výrokov a testovacia množina 389. Tab. 3 zobrazuje prehľad datasetu detskej slovenskej reči vytvoreného z relácie Táraninky.

Tab. 3 Prehľad datasetu detskej slovenskej reči z relácie Táraninky

Dataset Táraninky	**Trénovacia množina**	**Testovacia množina**
Počet rečníkov	74 (33 mužských, 41 ženských)	29 (10 mužských, 19 ženských)
Počet výrokov	1278 (612 mužských, 666 ženských)	389 (186 mužských, 203 ženských)
Priem. dĺžka výroku	7,22 sekúnd	7,1 sekúnd
Trvanie (hh:mm:ss)	02:33:48	00:45:57

4.2. Tvorba datasetu

Pre rozšírenie aktuálneho datasetu detskej slovenskej reči, ktorý je spomenutý v kapitole 4.1 bolo potrebné vykonať nový zber dát. V rámci práce bola vybraná relácia rádia Regina Východ s názvom Rozhlasové leporelo, ktorá je vysielaná každú sobotu okrem prázdnin. Táto relácia je obsahom určená pre detského poslucháča a je zameraná na formovanie myslenia, tvorivosti, pohotových detských reakcií a komunikačných schopností. Dĺžka jednej epizódy Rozhlasového leporela je 30 minút. [29]

V rámci jednotlivých odvysielaných epizód je prítomný moderátor rádia Regina Východ, ktorý uvádza danú epizódu, prípadne informuje o momentálnych sviatkoch. Moderátor sa taktiež buď priamo v štúdiových priestoroch rádia alebo v teréne pýta detí otázky situovaných k téme danej epizódy. Témy epizód zahŕňali napríklad otázky ohľadom umenia, športu, vesmíru, sviatkov a iné. Pri odborných témach je v epizóde taktiež prítomný učiteľ alebo vedec, ktorý sa snaží deťom priblížiť danú odbornú tematiku lákavým alebo hravým spôsobom.

Odpovedajúce deti boli z rôznych vekových kategórií. Epizódy, na ktoré bola práca zameraná boli nahrávané s deťmi z materských škôl, škôlok a prvého stupňa základnej školy. V niektorej z epizód sa pri mladších deťoch, napríklad z umeleckej školy taktiež nachádzal aj starší žiak ale aj to do veku maximálne štrnásť rokov. Väčšina odpovedí detí je teda v rozsahu od štyroch do jedenástich rokov s veľmi malým množstvom odpovedí od detí starších ako jedenásť rokov.

4.2.1. Zber dát

Samotné nahrávky odvysielaných epizód rádiovej relácie Rozhlasové leporelo boli prevzaté priamo z online archívu [29] rádia Regina Východ na stránke RTVS. Po prevzatí boli premenované vo formáte “RLDDMMYY.mp3“. Informácie o formáte prevzatých audio súborov zobrazuje Tab. 4 vytvorená na základe informácií o nahrávke “RL301021.mp3“ získaných pomocou programu MediaInfo.

Tab. 4 Informácie o formáte nahrávky “RL301021.mp3“

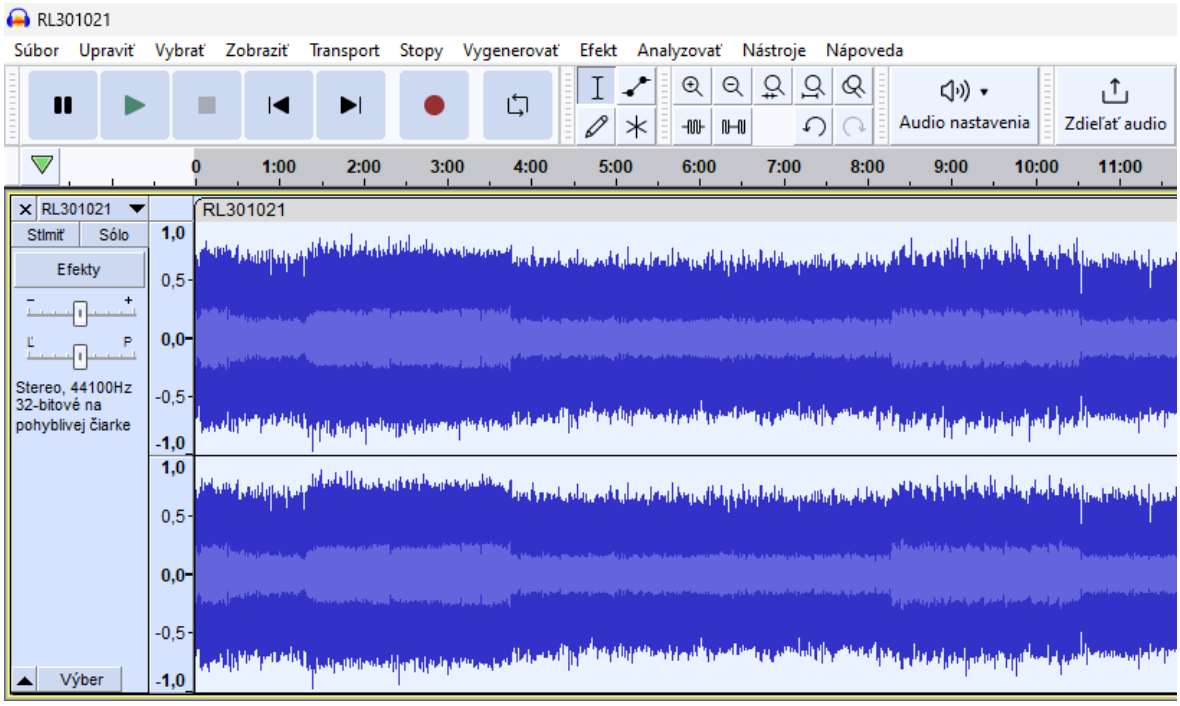
RL301021.mp3	
Formát	MPEG Audio
Verzia formátu	1
Formát profilu	Layer 3
Bitová rýchlosť	128 kb/s
Počet kanálov	2 kanály (stereo)
Vzorkovacia frekvencia	44.1 kHz
Trvanie	29 min 59 s

Takto premenované nahrávky boli následne umiestnené do priečinka “RAWrozhlasLeporelo“ a zodpovedajúcich priečinkov “YYYY-MM“ a pod priečinka “raw“. V prípade nahrávky “RL301021.mp3“ to bol priečinok “2021-10“.

V rámci práce bolo manuálne skontrolovaných 70 epizód relácie Rozhlasové Leporelo čo je 35 hodín audio záznamu, z čoho 36 epizód bolo preskočených z dôvodov nadmerného šumu/hluku v pozadí odpovede dieťaťa alebo kvôli absencii odpovedí detí v rámci hľadanej vekovej kategórie.

4.2.2. Prvotné spracovanie audio nahrávok

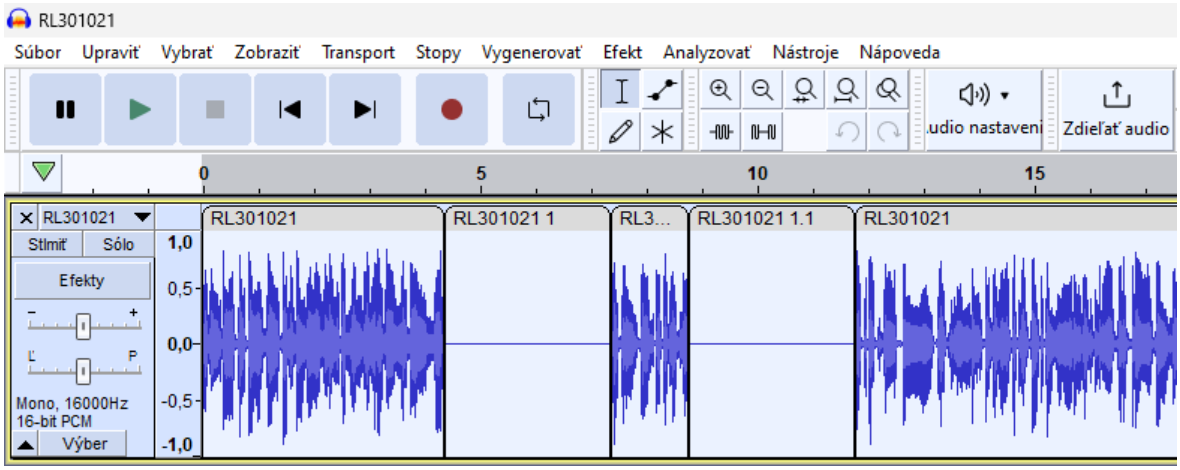
Ďalším dôležitým krokom bolo dôkladne prekontrolovanie a úprava týchto 34 epizód pomocou programu Audacity, ktorý je široko používaný bezplatný open-source softvér určený na úpravu zvuku. Každá z 34 epizód trvajúcich 30 minút bola podrobne analyzovaná. Analýza bola zameraná na identifikáciu a izoláciu segmentov, ktoré obsahovali výlučne detskú reč, čím sa eliminovali akékoľvek časti reči starších detí, dospelých ľudí alebo iný irelevantný zvukový obsah. Časť načítanej nepozmenenej nahrávky zobrazuje Obr. 8.



Obr. 8 Časť nahrávky “RL301021.mp3“ načítaná v programe Audacity

Aby sa zvýšila prehľadnosť analýzy a zachoval sa systematický postup, medzi jednotlivé rečové segmenty každého dieťaťa bola vložená trojsekundová pauza. Pridanie tejto pauzy výrazne zjednodušuje identifikáciu začiatkov a koncov jednotlivých výrokov rečníkov, čo zjednodušuje krok transkripcie výrokov.

Nahrávky epizód boli následne zmiešané zo sterea na mono, prevzorkované z pôvodnej vzorkovacej frekvencie 44100 Hz na 16000 Hz a ich formát zmenený z 32-bit float na 16-bit PCM. Táto zmena bola vykonaná na základe potrebného formátu pre model wav2vec2. Hlasitosť oddelených rečových segmentov detí bola normalizovaná použitím normalizácie špičky amplitúdy na -1 dB. Proces tejto normalizácie zvyšuje hlasitosť celej zvukovej stopy, kým najhlasnejší vrchol nedosiahne úroveň -1 dB. Táto hodnota bola zvolená tak, aby sa zabezpečilo, že zvuk nahrávky bude čo najhlasnejší bez toho, aby podliehal forme skreslenia zvuku menom clipping. Vďaka takejto normalizácii sú všetky dáta taktiež konzistentné v hlasitosti a nie sú medzi jednotlivými audio súbormi veľké rozdiely. Takáto úprava výrazne pomáha pri práci s modelmi umelej inteligencie. Časť načítanej nahrávky s vykonanými zmenami zobrazuje Obr. 9.



Obr. 9 Časť upravenej nahrávky "RL301021.mp3"

Výsledná upravená nahrávka "RL301021" je exportovaná vo formáte WAV a uložená do jej prislúchajúceho priečinka "2021-10". Informácie o formáte spracovanej nahrávky zobrazuje Tab. 5. Tento proces je vykonaný na všetkých 34 spracovaných epizódach relácie Rozhlasové leporelo. Takto spracované nahrávky sú pripravené k ďalšiemu spracovaniu a transkripcii. Bližšie popísané informácie o spracovaných epizódach ako sú presné dátumy odvysielania, popis epizód, ako aj ich trvanie, dôvody preskočenia epizódy a spísané ID rečníkov prítomných v jednotlivých epizódach sa nachádzajú v priloženom Excel súbore "Details_of_database_RL.xlsx".

Tab. 5 Informácie o formáte nahrávky "RL301021.wav"

RL301021.wav	
Formát	Wave (WAV)
Nastavenie formátu	PcmWaveformat
Bitová hĺbka	16-bit
Bitová rýchlosť	256 kb/s
Počet kanálov	1 kanál (mono)
Vzorkovacia frekvencia	16 kHz
Trvanie	4 min 3 s

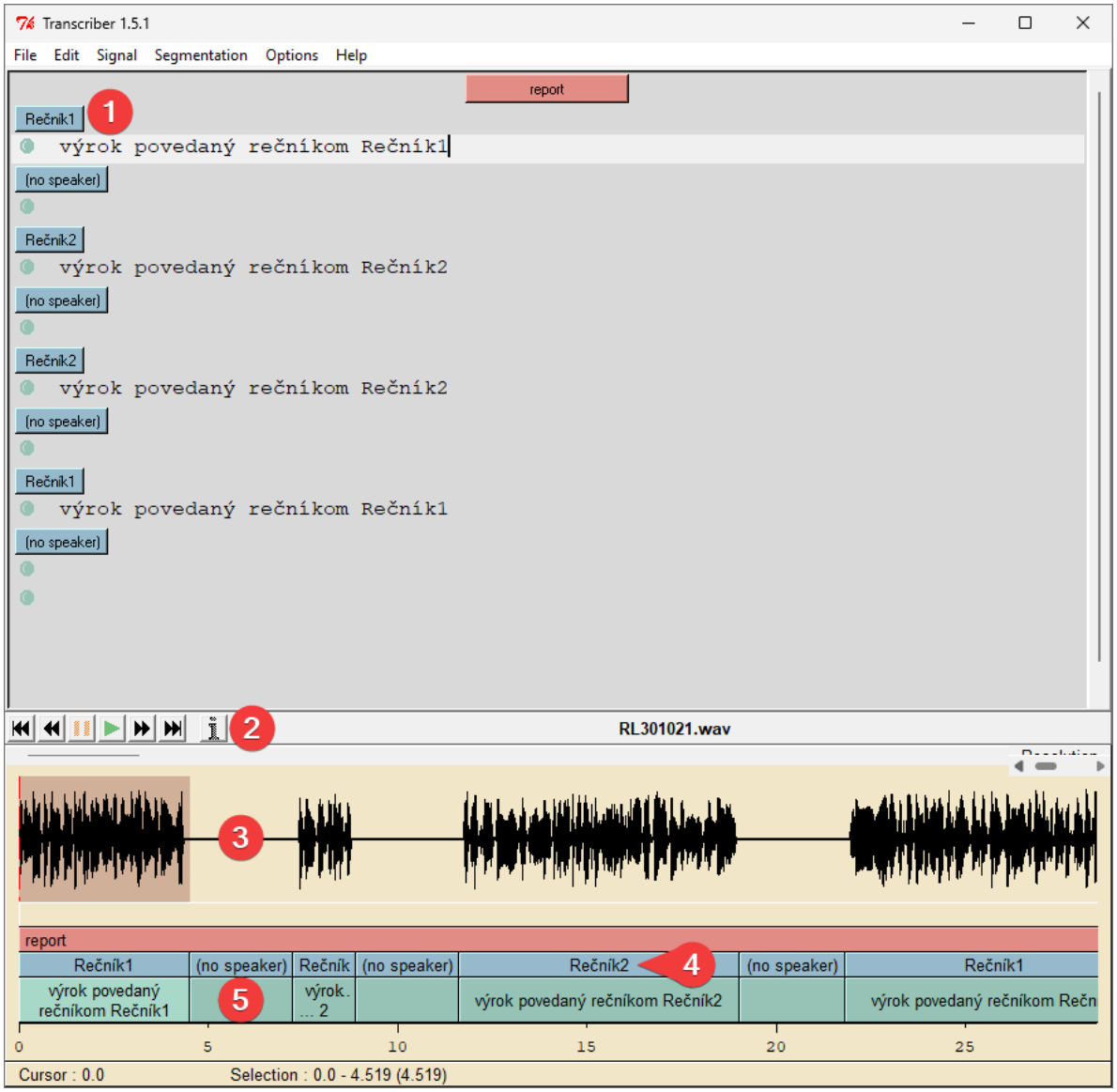
Ako bolo spomenuté, celkovo bolo spracovaných 34 epizód a z nich vytvorených 34 audio nahrávok vo formáte WAV.

4.2.3. Tvorba transkripcie pomocou programu Transcriber

Po prvotnom spracovaní audio nahrávok bolo potrebné vytvoriť ku každej spracovanej nahrávke manuálnu transkripciu. Pre vytvorenie transkripcie ku jednotlivým nahrávkam bol použitý voľne dostupný program Transcriber verzie 1.5.1.

4.2.3.1. Prostredie programu Transcriber

Pre prácu s programom Transcriber je potrebné poznať jeho prostredie a jednotlivé časti tohto prostredia. Pre jednoduchšie vysvetlenie jednotlivých častí bol vytvorený Obr. 10 prostredia programu s pridanými bodmi 1 až 5.



Obr. 10 Popis prostredia programu Transcriber

Vysvetlenie jednotlivých bodov:

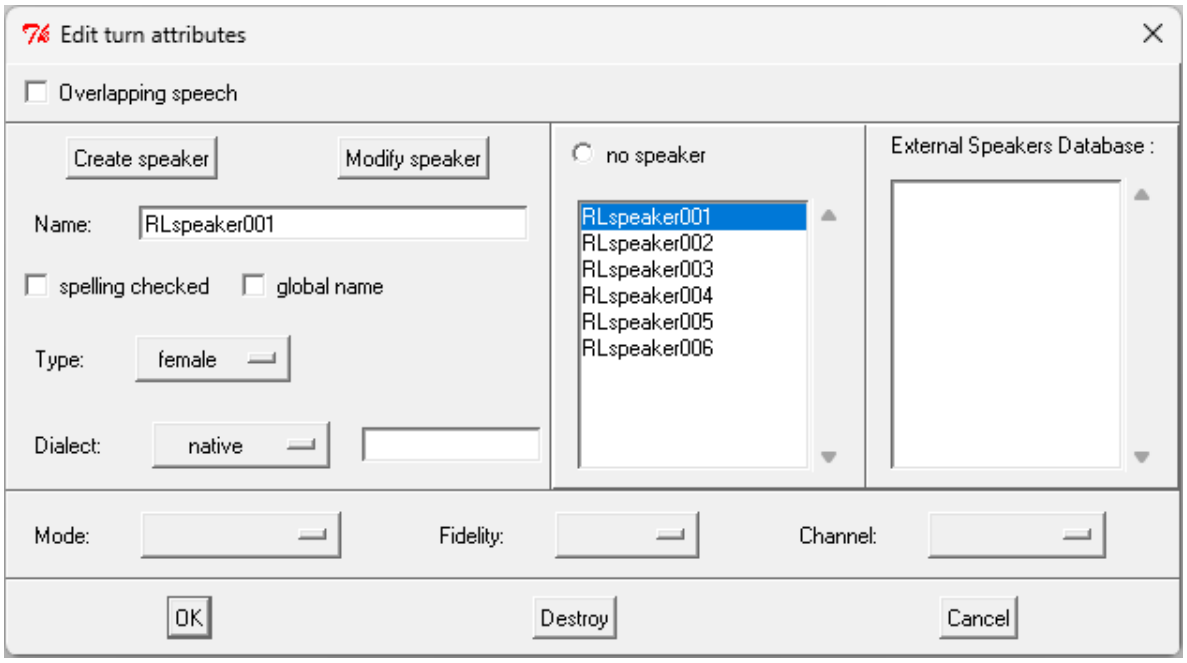
1. Oblasť pre textový prepis výroku aktuálne zvoleného zvukového segmentu s možnosťou tvorby a výberu rečníka stlačením tlačidla s momentálnym názvom "(no speaker)". Pri každom rozdelení nahrávky na menšie segmenty je vytvorený nový riadok pre transkripciu a identifikáciu rečníka. Akciu tvorby a výberu rečníka je možné vyvolať aj prostredníctvom klávesovej skratky *Ctrl + T*. Okno zobrazujúce vytvorenie a výber rečníka je neskôr popísané v podkapitole 4.2.3.2.
2. Oblasť obsahujúca ovládacie prvky prehrávania zvuku ako je prehratie, pozastavenie, posun dopredu/dozadu a skok medzi jednotlivými zvukovými segmentami.
3. Oblasť zobrazuje zvukové vlny, bežne označované aj ako zobrazenie zvukového priebehu, momentálne načítanej nahrávky. Táto funkcia vizualizuje zvukový signál a ukazuje zmeny amplitúdy v priebehu času. Tvary vĺn sú nevyhnutné pri tvorbe transkripcie, pretože poskytujú vizuálnu reprezentáciu zvuku, čo uľahčuje identifikáciu konkrétnych rečových segmentov od pridaných tichých miest pre rozdelenie jednotlivých výrokov. Pre vytvorenie rozdelenia celkovej nahrávky na menšie segmenty je používaná klávesa *Enter*, ktorá vytvorí rozdelenie v momentálne zakliknutom bode nahrávky.
4. Oblasť zobrazuje meno rečníka, ktoré je priradené k určitému segmentu a transkripcii zvukovej nahrávky v priebehu času.
5. Oblasť rozdelená na časti, segmenty, pre zobrazenie transkripcie jednotlivých výrokov v priebehu času.

4.2.3.2. Tvorba a výber rečníkov

Jedným z krokov tvorby datasetu bolo priradenie mena rečníka k určitej transkripcii segmentu, ktorý mu patrí. V rámci upravených nahrávok epizód sa mohol v jednej epizóde nachádzať jeden rečník, ktorému patrili všetky výroky alebo sa rečníci v rámci jednej epizódy menili. Najvyšší počet individuálnych rečníkov v rámci jednej epizódy bol osemnásť.

Tvorba a výber rečníka je vykonávaná v okne zobrazenom na Obr. 11. Okno obsahuje možnosť tvorby nového rečníka, úpravy už existujúceho rečníka a výberu z listu existujúcich rečníkov. Rečníkom je možné priraďovať rôzne vlastnosti ako je pohlavie, dialekt, či sa jedná o pripravenú alebo spontánnu reč a iné. V rámci práce boli rečníkom priradené iba informácie o ich identifikačnom mene a pohlaví. Ostatné informácie neboli potrebné pre prácu s modelom wav2vec2. Identifikačné meno bolo vytvorené v tvare "RLspeakerXXX" kde trojica znakov "XXX" reprezentuje číselné poradie rečníka. Rečníci sa medzi jednotlivými epizódami neopakovali. Pre

každého nového rečníka bolo priradené unikátne poradové číslo, ktoré bolo inkrementom predchádzajúceho.



Obr. 11 Okno pre tvorbu a výber rečníka

4.2.3.3. Proces transkripcie audio nahrávok

Pred začiatkom samotného procesu transkripcie bolo potrebné v programe Transcriber načítať konfiguračný súbor prostredníctvom výberu z horného menu *Options -> Load configuration file* a následným vybraním súboru "RL_transcriber_config.cfg". Tento súbor obsahuje uložené nastavenie prostredia programu Transcriber, ktoré bolo použité pri tvorbe transkripcií v tejto práci. Súbor "RL_transcriber_config.cfg" je priložený v prílohe A.

Po načítaní konfiguračného súboru bolo prostredie pripravené k začatiu tvorby transkripcie pre jednotlivé audio nahrávky. Pre začatie transkripcie je potrebné prostredníctvom výberu z horného menu *File -> New trans* a následným vybratím audio súboru vytvoriť novú transkripciu. Týmto výberom je audio nahrávka načítaná v programe Transcriber. Pre ukážku v práci bola načítaná nahrávka "RL301021.wav". Po jej načítaní bolo v prostredí programu možné vidieť v oblasti 3 popísanej na Obr. 10 zobrazenie zvukového priebehu, vďaka ktorému bola nahrávka prostredníctvom klávesy *Enter* rozdelená na začiatkoch a koncoch tichých miest pridaných v programe Audacity pre rozdelenie jednotlivých výrokov.

Pre jednotlivé rozdelené segmenty boli následne vytvorený alebo priradený zodpovedajúci rečníci. Všetkým segmentom obsahujúcim tiché miesta bol namiesto rečníka priradený "(no speaker)" (z angl. žiaden rečník) pre ďalšie spracovanie. Po priradení rečníka ku

segmentu nahrávky bolo potrebné manuálne prepísať vyslovený výrok, ktorý tento segment obsahoval. Nakoľko sa jednalo o transkripciu reči malých detí jej prepis bol častokrát zdĺhavý a zložitý proces. Proces transkripcie detskej reči predstavoval niekoľko výziev ako napríklad stále rozvíjajúce sa artikulačné schopnosti, obmedzená slovná zásoba, ktorá obsahuje neúplné alebo gramaticky nesprávne slová, rýchla a nezrozumiteľná reč a iné. Tieto charakteristiky detskej reči sťažovali proces transkripcie a často bolo potrebné niektoré audio segmenty prehrávať viackrát pre dosiahnutie optimálnej transkripcie. Pre čo najlepšie zlepšenie správnosti transkripcie boli využívané rôzne slovníky pre overenie určitých slov.

V programe Transcriber bolo taktiež možné pridávať do transkripcie poznámky prostredníctvom klávesovej skratky *Ctrl + D*. Proces transkripcie poskytnutého datasetu tvoreného z relácie Táraninky obsahoval tieto poznámky, ktoré boli použité pre označenie rôznych zvukov v rámci nahrávky ako sú dýchanie, smiech, váhanie rečníka a iné. Tieto anotácie boli potrebné pri práci s modelmi Kaldi. Pri práci s modelom wav2vec2 tieto poznámky neboli potrebné a boli z pôvodného datasetu odstraňované. V dôsledku toho pri transkripcii epizód relácie Rozhlasové leporelo bolo rozhodnuté nepriraďovať poznámky pre označenie spomínaných zvukov čo aspoň čiastočne zrýchlilo proces transkripcie.

Posledným krokom po dokončení transkripcie bolo uloženie projektu programu Transcriber prostredníctvom výberu z horného menu *File -> Save as* a uložením v prípade ukážky ako “RL301021.trs“. Takéto uloženie povoľovalo opätovné vrátenie sa k transkripcii a umožňovalo opravovať možné chyby. Finálnu transkripciu v prostredí programu Transcriber je možné vidieť na Obr. 13. Po uložení projektu bola vytvorená transkripcia exportovaná prostredníctvom výberu z horného menu *File -> Export -> Export to STM format* v tomto prípade “RL301021.STM“. Ukážku časti STM súboru je možné vidieť na Obr. 12. Formát STM bol zvolený pre ďalšie spracovanie pomocou skriptov. Tento proces bol zopakovaný pre všetkých 34 spracovaných epizód Rozhlasového leporela.

```
RL301021 1 RL301021_RLspeaker001 0.000 4.380 <o,f0,female> to vymysleli preto lebo aby zahnali zlých
duchov zimy aby im nebolo zle
RL301021 1 inter_segment_gap 4.380 7.368 <o,f0,>
RL301021 1 RL301021_RLspeaker002 7.368 8.796 <o,f0,female> taký trošku čierny
RL301021 1 inter_segment_gap 8.796 11.762 <o,f0,>
RL301021 1 RL301021_RLspeaker002 11.762 19.004 <o,f0,female> tekvice a keď je dušičky tak tam sú sviečky a
chodíme na cintorín
RL301021 1 inter_segment_gap 19.004 21.958 <o,f0,>
RL301021 1 RL301021_RLspeaker001 21.958 29.574 <o,f0,female> no hovorí sa im tak preto lebo môžme pre
svojich blízkych, ktorí už zomreli vyprosiť úplný odpustok a môžu ísť do neba
RL301021 1 inter_segment_gap 29.574 32.501 <o,f0,>
RL301021 1 RL301021_RLspeaker001 32.501 36.740 <o,f0,female> no, že vtedy sú jak keby také prázdniny,
vtedy chodíme na pohreby
```

Obr. 12 Ukážka časti transkripcie v STM súbore

Obr. 13 Finálna transkripcia v programe Transcriber

4.2.4. Prehľad spracovaných epizód s transkripciou

V rámci práce bolo spracovaných a transkribovaných 34 epizód rádiovej relácie Rozhlasové leporelo. Pomocou Python skriptu a použitia knižnice pydub boli zistené trvania jednotlivých spracovaných epizód ako aj celkové trvanie epizód dokopy. Na základe týchto informácií bola vytvorená Tab. 6. Celkové trvanie získaných nahrávok pred ďalším spracovaním a bez odstránenia jednotlivých 3-sekundových medzier medzi jednotlivými rečovými segmentami je 1 hodina 33 minút a 41 sekúnd.

Tab. 6 Sumárny prehľad spracovaných epizód a ich trvania

Názov epizódy	Dátum odvysielania	Trvanie (hh:mm:ss)
RL301021	30.10.2021	00:04:04
RL131121	13.11.2021	00:03:29
RL271121	27.11.2021	00:02:04
RL041221	4.12.2021	00:02:03
RL080122	8.1.2022	00:02:59
RL220122	22.1.2022	00:10:07
RL290122	29.1.2022	00:00:29
RL050222	5.2.2022	00:00:42
RL190222	19.2.2022	00:02:47
RL210522	21.5.2022	00:00:51
RL040622	4.6.2022	00:01:09
RL180622	18.6.2022	00:02:18
RL170922	17.9.2022	00:02:10
RL240922	24.9.2022	00:02:17
RL221022	22.10.2022	00:02:16
RL191122	19.11.2022	00:08:55
RL031222	3.12.2022	00:07:11
RL171222	17.12.2022	00:00:55
RL070123	7.1.2023	00:00:55
RL140123	14.1.2023	00:03:08
RL110223	11.2.2023	00:07:48
RL180223	18.2.2023	00:02:27
RL040323	4.3.2023	00:01:19
RL110323	11.3.2023	00:02:53
RL180323	18.3.2023	00:02:24
RL150423	15.4.2023	00:01:53
RL220423	22.4.2023	00:01:14
RL060523	6.5.2023	00:00:48
RL130523	13.5.2023	00:02:54
RL200523	20.5.2023	00:03:27
RL270523	27.5.2023	00:02:58
RL030623	3.6.2023	00:00:22
RL100623	10.6.2023	00:01:54
RL170623	17.6.2023	00:00:30
Celkovo		**01:33:41**

4.2.5. Rozdelenie nahrávok na jednotlivé segmenty

Po predošlom spracovaní, nahrávky stále obsahovali 3-sekundové pauzy a jednotlivé segmenty neboli rozdelené. Pre správne trénovanie modelu wav2vec2 bolo potrebné tieto nahrávky ďalej spracovať. Pre spracovanie boli spracované nahrávky vo formáte WAV spolu s ich transkripciou vo formáte STM umiestnené do jedného spoločného priečinka. Prostredníctvom programu WinSCP a použitia školskej VPN bol tento priečinok nahraný na školský server Tesla, ktorý poskytoval funkcionality nástroja Kaldi.

Na školskom serveri Tesla bola vykonávaná úprava dát v pracovnom priečinku /Sevc/. V nasledujúcich odrážkach sú popísané podpriečinky nachádzajúce sa v pracovnom priečinku:

- **corpus/rozhlas_leporelo/** Do podpriečinka rozhlas_leporelo boli umiestnené všetky spracované WAV a STM súbory
- **corpus/data/** Do podpriečinka data boli vytvárané výstupné súbory pomocou skriptov
- **local/** Do podpriečinka local bol umiestnený skript “create_segments.sh“ a “lines_map.sh“

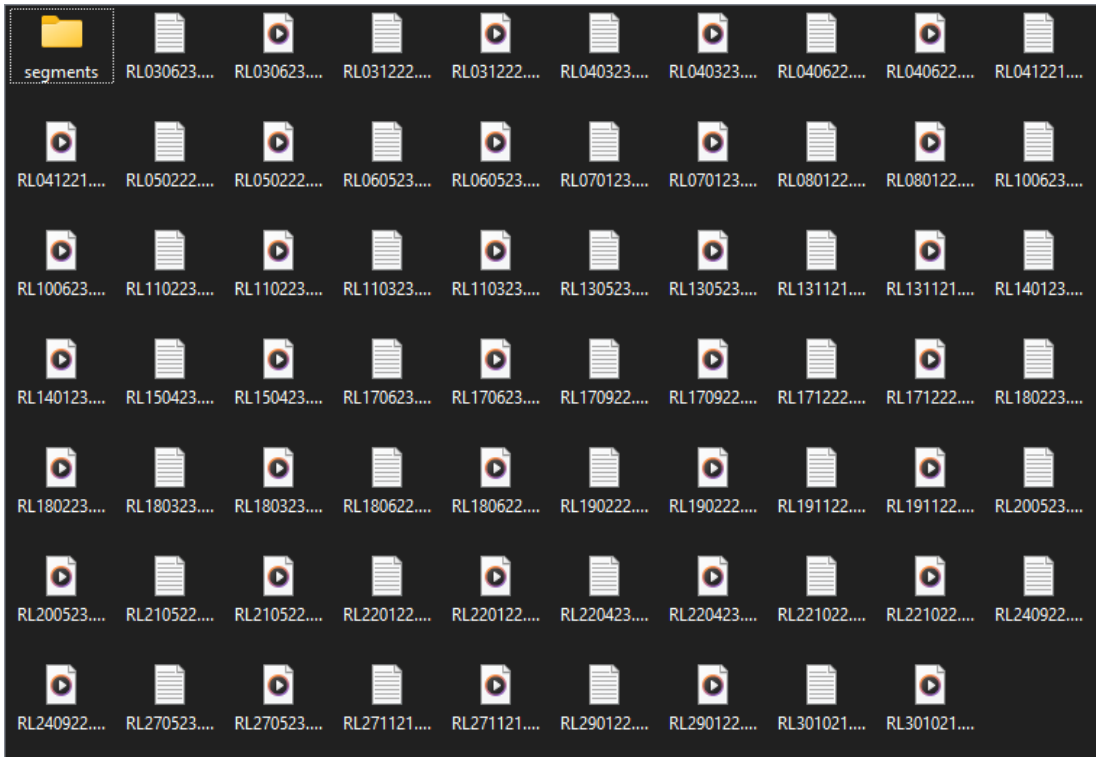
Skripty “create_segments.sh“ a “lines_map.sh“ sú upravenou verziou poskytnutých skriptov [16], ktoré boli použité aj pri tvorbe datasetu z relácie Táraninky. Spustením skriptu “./local/create_segments.sh“ bol v priečinku “rozhlas_leporelo“ vytvorený podpriečinok “segments“. Následne skript prechádzal všetkými “.STM“ súbormi v priečinku “/corpus/rozhlas_leporelo” a získaval informácie o transkripcii pre zvukové súbory, ako sú časové údaje o začiatku a konci segmentov jednotlivých rečníkov a samotnú transkripciu. Riadky neobsahujúce potrebné informácie (napr. riadky obsahujúce “;;”, “excluded_region”, “inter_segment_gap”) boli ignorované.

Pre ostatné riadky boli vykonávané následné operácie:

- **Vytvorenie súboru “spk2gender”** – Tento súbor priradzuje každého rečníka “spk” k jeho pohlaviu “gender”. Skript kontroluje, či už rečník bol pridaný do zoznamu, ak nie, tak ho pridal s príslušným pohlavím.
- **Vytvorenie súboru “utt2spk”** – Tento súbor priradzuje každý segment reči “utt” k príslušnému rečníkovi. Segmenty reči boli identifikované unikátnymi názvami, ktoré zahŕňajú identifikátor rečníka, súboru a poradové číslo segmentu.
- **Vytvorenie súboru “text”** – Tento súbor obsahuje unikátne názvy segmentu jedného výroku a medzerou oddelenú transkripciu daného výroku.

- **Segmentácia zvukových súborov** – Pomocou nástroja "sox" boli zvukové nahrávky rozdelené na menšie segmenty podľa časových údajov zo ".STM" súborov. Výsledkom sú zvukové segmenty, ktoré obsahujú len výroky rečníkov a 3-sekundové pauzy boli odstránené.
- **Vytvorenie súboru "spk2utt"** – Pomocou nástroja Kaldi je vytvorená inverzia súboru "utt2spk", ktorá priradzuje jednotlivým rečníkom im patriace zvukové segmenty.

Ukážka priečinka "rozhlas_leporelo" v programe WinSCP, kde je možné vidieť umiestnené spracované WAV a STM súbory spolu s vytvoreným priečinkom "segments" pomocou skriptov je zobrazená na Obr. 14. Spustenie skriptu "create_segments.sh" taktiež zavolalo skript "lines_map.sh", ktorý odstránil možné interpunkčné znamienka obsiahnuté v transkripcii v súbore "text".



Obr. 14 Obsah priečinka rozhlas_leporelo

Výstupný súbor "text" bol upravený do formátu CSV, k názvom segmentov bola pridaná koncovka ".wav" a medzera medzi názvom súboru a transkripciou bola nahradená čiarkou. Niektoré slová, ako napríklad mená, boli v rámci transkripcie písané s veľkým začiatočným písmenom. Vo finálnej forme boli všetky písmená pomocou skriptu zmenené na malé. Výsledný súbor bol uložený ako "metadata.csv".

4.2.6. Prehľad vytvoreného datasetu relácie Rozhlasové leporelo

Spracovaním 34 nahrávok pomocou spomenutých skriptov bolo vytvorených 617 jednotlivých zvukových segmentov, ktoré boli uložené vo formáte WAV. Súbory “spk2utt“ a “spk2gender“ boli ďalej použité pre vytvorenie prehľadu vytvoreného datasetu z relácie Rozhlasové leporelo. Pomocou Python skriptov boli získané informácie pre vytvorenie Tab. 7, ktorá poskytuje informácie o počte rečníkov, celkový počet výrokov, priemernú dĺžku segmentu a celkové trvanie vytvoreného datasetu relácie Rozhlasové leporelo.

Tab. 7 Prehľad finálneho datasetu relácie Rozhlasové leporelo

Dataset Rozhlasové leporelo	
Počet rečníkov	188 (78 mužských, 110 ženských)
Počet výrokov	617 (229 mužských, 388 ženských)
Priemerná dĺžka výroku	6,42 sekúnd
Trvanie (hh:mm:ss)	01:06:00

4.3. Rozšírenie datasetu spojením

Jedným zo zameraní práce bolo zlepšenie presnosti a robustnosti modelu wav2vec2 v kontexte rozpoznávania detskej reči. Aby toho bolo dosiahnuté, existujúci dataset [16] detskej reči vytvorený z relácie Táraninky bol skombinovaný s novým datasetom detskej reči vytvorenom z rádiovej relácie Rozhlasové leporelo podobného charakteru, ktorý bol starostlivo vytvorený a anotovaný ako súčasť výskumu tejto práce. Cieľom tohto rozšírenia bolo obohatenie rozmanitosti a zvýšenie objemu trénovacích dát pre trénovanie modelu wav2vec2.

Existujúci dataset slúžil ako rozsiahly korpus, ktorý ponúkal pestrú škálu vzoriek detskej reči. Aby boli vyriešené potenciálne nedostatky a zavedená širšia škála jazykových vlastností a akustickej variability pôvodného datasetu, tak bol dataset rozšírený o dáta vytvorené v tejto práci.

Nakoľko je zber dát detskej reči, ich spracovanie a transkripcia veľmi časovo náročný proces, získané dáta sú veľmi cenné pre samotný tréning a zlepšenie presnosti predikcie ako pre rozširovanie testovacej množiny pre vyhodnotenie presnosti modelu. V rámci práce preto bolo rozhodnuté, že rozšírenie datasetu bolo vykonané len pre trénovaciu množinu dát. Testovacia množina dát nebola rozširovaná a bola zachovaná v pôvodnom rozsahu, segmentoch a rečníkoch. Takýto stav testovacej množiny bol použitý pre vyhodnotenie presnosti wav2vec2 modelu v práci P. Michalanského [16]. Takéto rozhodnutie taktiež napomáha pri vyhodnocovaní modelov a ich porovnaní s doposiaľ naučeným modelom a jeho presnosti na rovnakej testovacej množine.

4.3.1. Postup prvotného rozšírenia datasetu spojením

Prvotné rozšírenie trénovacej množiny bolo vykonané s úmyslom zachovať metodológiu trénovania použitého pri trénovaní modelu “wav2vec2-XLS-R-SK-CHILD-v8.2“ [16], ktorý dosiahol najlepšiu presnosť predikcie na testovacej množine datasetu Táraninky. Toto rozšírenie spočívalo v spojení trénovacej časti datasetu relácie Táraninky a datasetu relácie Rozhlasové leporelo.

Prvá fáza rozšírenia zahŕňala dôkladné preskúmanie existujúceho datasetu relácie Táraninky a nového datasetu, aby sa zabezpečila kompatibilita a konzistentnosť formátovania údajov a obsahu.

Po overení bol vykonaný proces spojenia kombináciou súborov “metadata.csv” z oboch datasetov. Tento súbor slúži ako hlavný zoznam, ktorý obsahuje názvy jednotlivých zvukových segmentov a im súvisiace transkripcie. Zlúčenie týchto súborov si vyžadovalo pozornosť, aby bola zachovaná integrita údajov a zabezpečilo sa, že názov súboru každého zvukového segmentu zodpovedá jeho transkripcii.

Po úspešnej kombinácii súborov “metadata.csv” bol ďalší krok zameraný na samotné zvukové súbory. Zvukové súbory existujúceho aj nového datasetu boli prekopírované do jedného adresára. Po spojení bolo odkontrolované, že každý súbor zvukového segmentu je obsiahnutý v aktualizovanom súbore “metadata.csv”.

Výsledkom je rozšírený dataset pripravený k ďalšiemu spracovaniu a následnému použitiu pri trénovaní modelu wav2vec2. Pomocou Python skriptov bol vytvorený prehľad rozšíreného datasetu, ktorý je popísaný v Tab. 8.

Tab. 8 Prehľad rozšíreného trénovacieho datasetu

Rozšírený trénovací dataset	
Počet rečníkov	262 (111 mužských, 151 ženských)
Počet výrokov	1895 (841 mužských, 1054 ženských)
Priemerná dĺžka výroku	6,96 sekúnd
Trvanie (hh:mm:ss)	03:39:49

4.4. Prvotná dátová augmentácia rozšíreného datasetu

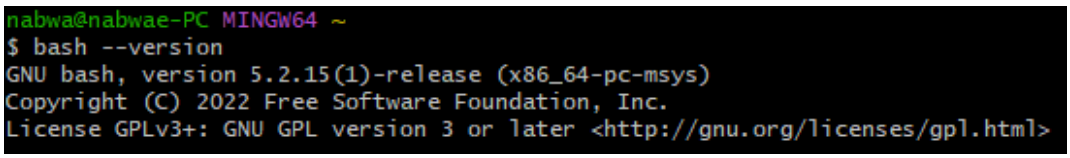
Na základe výsledkov [16] predchádzajúcej práce P. Michalanského v oblasti rozpoznávania slovenskej reči a spracovania zvuku bolo zistené možné zlepšenie výkonnosti modelu po rozšírení datasetu pomocou augmentácie. V rámci predchádzajúcej práce bol dataset vytvorený z relácie Táraninky rozširovaný pomocou šiestich metód augmentácie zvukových dát. Analýzou existujúcej

práce bolo zistené, že najúčinnejšie výsledky pre model wav2vec2 dosahovali kombinácie augmentácie spektrogramu (metóda SpecAugment) a augmentácie pomocou perturbácie rýchlosti.

V nadväznosti na tieto výsledky bolo rozhodnuté zamerať sa a aplikovať spomenuté augmentácie a ich kombinácie, ktoré dosiahli najlepšie výsledky aj na rozšírený trénovací dataset vytvorený v tejto práci.

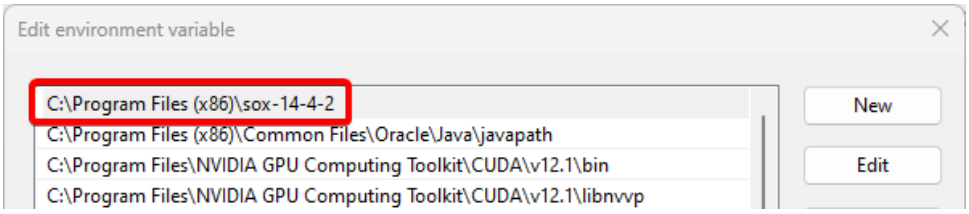
4.4.1. Príprava systému na augmentáciu datasetu

Na lokálny počítač s operačným systémom Windows 11 bol nainštalovaný program Git Bash, ktorý poskytuje v systéme Windows terminálové prostredie podobné Unixu, ktoré umožňuje spúšťanie skriptov a príkazov podobných systému Linux. Verzia nástroja Git Bash použitá v práci bola "2.42.0.windows.2". Po jeho inštalácii bolo možné overiť verziu GNU bash-u pomocou príkazu "bash --version" po spustení aplikácie Git Bash. Výstup tohto príkazu je zobrazený na Obr. 15.



Obr. 15 Výstup príkazu pre kontrolu bash verzie

Ďalším potrebným krokom bola inštalácia nástroja Sound eXchange (ďalej len SoX). Verzia nástroja SoX použitá pri práci bola "v14.4.2". Po inštalácii bolo potrebné pridať cestu k nainštalovanému nástroju SoX do premennej prostredia PATH lokálneho systému. Tento krok bol kľúčový pre sprístupnenie príkazu "sox" v Git Bash alebo inom prostredí príkazového riadka na tomto systéme. Správne pridanú cestu v premennej prostredia PATH zobrazuje Obr. 16.



Obr. 16 Nastavenie cesty PATH pre nástroj SoX

Správnosť inštalácie nástroja SoX bolo možné vykonať pomocou spustenie nástroja "sox.exe" s argumentom "--version". Výstup spustenia nástroja SoX s týmto argumentom je zobrazený na Obr. 17.

```
C:\Program Files (x86)\sox-14-4-2>sox.exe --version
sox.exe:      SoX v14.4.2
```

Obr. 17 Výstup kontroly verzie nástroja SoX

4.4.2. Postup prvotnej augmentácie rozšíreného datasetu

Pre spoľahlivé odsledovanie výsledkov a možného zlepšenia alebo zhoršenia predikcie modelu wav2vec2 bola pre prvotné aplikovanie augmentácie zachovaná metodológia rozšírenia datasetu pomocou augmentácie z práce P. Michalanského. Primárnym cieľom bolo zistiť, či techniky augmentácie, ktoré priniesli sľubné výsledky s pôvodným datasetom relácie Táraninky, si zachovajú svoju účinnosť alebo vykážu ešte väčšie zlepšenie, keď sa aplikujú na rozšírený dataset vytvorený v tejto práci. Ako bolo spomenuté, najlepšie výsledky pri trénovaní modelu wav2vec2 dosahovali augmentácie spektrogramu (SpecAugment) a augmentácia pomocou perturbácie rýchlosti.

Pre augmentáciu celej rozšírenej trénovacej množiny dát pomocou perturbácie rýchlosti bol v rámci práce vytvorený a použitý bash skript “speed_perturbation_windows.sh” na automatizáciu tohto procesu. Skript vykonáva niekoľko funkcií, vrátane organizácie zvukových súborov, manipulácie s ich rýchlosťou pomocou nástroja SoX na účely augmentácie a rozšírenie súboru “metadata.csv”.

Proces augmentácie použitím skriptu “speed_perturbation_windows.sh” a nástroja SoX bol vykonávaný na lokálnom počítači s operačným systémom Windows 11. Augmentácia dát pomocou perturbácie rýchlosti bola vykonávaná v pracovnom priečinku /Speed_perturbation/. V nasledujúcich odrážkach sú popísané podpriečinky nachádzajúce sa v pracovnom priečinku:

- **train/segments/** Do podpriečinka segments boli umiestnené všetky spracované zvukové segmenty
- **train/speed/** Do podpriečinka speed boli vytvárané výstupné súbory pomocou skriptu (zvukové segmenty originálnej, spomalenej a zrýchlenej rýchlosti a upravený súbor “metadata.csv”)
- **train/** Do podpriečinka train bol umiestnený súbor “metadata.csv” pre spracované zvukové segmenty
- **local/** Do podpriečinka local bol umiestnený skript “speed_perturbation_windows.sh“

Spustením skriptu “./local/speed_perturbation_windows.sh“ bol v priečinku “train” vytvorený podpriečinok “speed”, ktorý slúžil ako cieľový priečinok pre upravené zvukové súbory.

Skript následne prechádzal všetkými záznamami v súbore “metadata.csv”, ktorý obsahoval názvy zvukových súborov a príslušné transkripcie.

V rámci procesu skript vykonával nasledujúce operácie:

- **Vytvorenie a úprava súboru “metadata.csv”** – Pre každý zvukový súbor bol vytvorený záznam v novom “metadata.csv” súbore, ktorý obsahoval pôvodný názov súboru, názov súboru so zmenenou rýchlosťou spolu s príslušnou transkripciou.
- **Triedenie záznamov** – Súbor “metadata.csv” bol usporiadaný tak, že prvý riadok (hlavička) bol zachovaný na vrchu a zvyšné riadky boli zotriedené, čím bola zabezpečená konzistentnosť a prehľadnosť údajov.
- **Kopírovanie a úprava zvukových súborov** – Skript skopíroval všetky pôvodné “.wav” súbory do cieľového priečinka “speed” a pomocou nástroja SoX vytvoril pre každý zvukový súbor dve verzie s upravenou rýchlosťou. V rámci práce sú použité parametre spomaľujúci faktor (angl. slowdown_factor), ktorý bol nastavený na hodnotu 0,9, čo predstavuje spomalenie nahrávok o 10% pôvodnej rýchlosti (teda rýchlosť bola upravená na 90% pôvodnej), a zrýchľovací faktor (angl. speedup_factor) s nastavenou hodnotou 1,1, čo znamená zrýchlenie nahrávok o 10% pôvodnej rýchlosti (teda rýchlosť bola upravená na 110% pôvodnej). Pre každú verziu bol vytvorený samostatný súbor s unikátnym názvom, ktorý identifikoval upravenú rýchlosť.

Výstupom spustenia skriptu boli zvukové súbory s pôvodnou rýchlosťou, vytvorené zvukové súbory s modifikovanou rýchlosťou a príslušný upravený súbor “metadata.csv” s transkripciou pre všetky zvukové súbory. Tento proces umožnil efektívnu a automatickú augmentáciu zvukových dát pomocou perturbácie rýchlosti v rámci práce. Takáto automatizácia je zvlášť užitočná pri príprave veľkého množstva dát pre trénovanie modelov strojového učenia ako je model wav2vec2.

Výsledkom je, že pôvodný rozšírený dataset, ktorého tvorba a rozsah je podrobnejšie popísaná v kapitole 4.3.1, bol pomocou augmentácie perturbácie rýchlosti rozšírený na trojnásobok pôvodného počtu výrokov. Jeho nový prehľad s rozšíreným počtom výrokov ako aj celkovým trvaním zobrazuje Tab. 9.

Tab. 9 Prehľad augmentovaného rozšíreného trénovacieho datasetu

Augmentovaný rozšírený trénovací dataset	
Počet rečníkov	262 (111 mužských, 151 ženských)
Počet výrokov	5685 (2523 mužských, 3162 ženských)
Priemerná dĺžka výroku	7 sekúnd
Trvanie (hh:mm:ss)	11:03:52

Augmentácia spektrogramu je podporovaná knižnicou Transformers a bola vykonávaná priamo pri trénovaní modelu wav2vec2 a je možné ju nakonfigurovať pri samotnom načítaní predtrénovaného modelu.

5. Trénovanie modelov

V tejto kapitole je popísaný priebeh trénovania modelov automatického rozpoznávania reči pre detských používateľov. Kapitola podrobne popisuje postup trénovania a vyhodnocovania modelov.

5.1. Využitie platformy Kaggle na trénovanie

Proces trénovania wav2vec2 modelu predstavuje dve hlavné výzvy a tými je dlhé trvanie trénovania a značné požiadavky na pamäť. Tieto výzvy sú vlastné povahe najmodernejších modelov hlbokého učenia, ktoré sú výpočtovo a pamäťovo náročné.

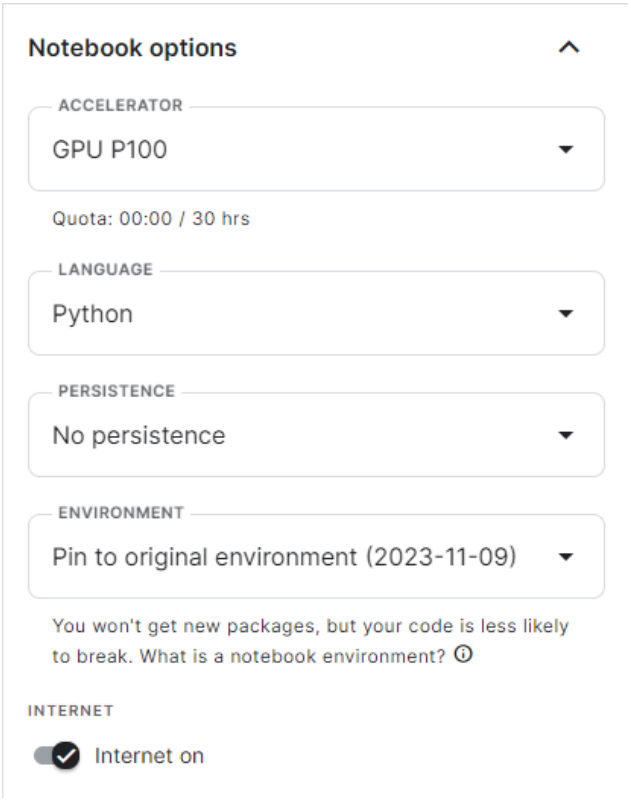
Dlhé časy trénovania: Model wav2vec2 vyžaduje vďaka svojej hlbokej a komplexnej architektúre značné množstvo výpočtového výkonu. Trénovanie takýchto modelov zahŕňa spracovanie a učenie sa z veľkých datasetov, čo si vyžaduje rozsiahle časové nároky na výpočty. Tieto časové nároky sa ešte viac zvýšia, keď dostupnému hardvéru, ako napríklad lokálnej GPU, chýbajú špecializované jadrá a optimalizovaná architektúra prítomná v pokročilejších GPU vyhradených práve pre výpočty sofistikovaných modelov hlbokého učenia. Výsledkom je, že trénovací proces na lokálnej GPU môže trvať neprakticky dlho, alebo byť pamäťou úplne obmedzený, čo bráni efektívnosti a pokroku.

Pamäťové obmedzenie: Okrem výpočtových požiadaviek, model wav2vec2 pri trénovaní tiež kladie značné požiadavky na pamäť. To zahŕňa pamäť potrebnú na ukladanie parametrov modelu, dát na trénovanie a medzivýpočty, ktoré sú potrebné počas trénovacieho procesu. Lokálny hardvér nebol vybavený dostatočnou kapacitou pamäte, takže táto limitácia bola prekážkou, ktorá obmedzovala schopnosť zvládnuť zložitosť modelu a objem údajov v uspokojujúcom čase.

Na základe týchto výziev boli v rámci práce využité výpočtové zdroje poskytnuté spoločnosťou Kaggle. Platforma Kaggle ponúka vysokovýkonné GPUs s rozsiahlymi pamäťami, ktoré sú špeciálne navrhnuté tak, aby zvládli požiadavky tréningu sofistikovaných modelov hlbokého učenia, ako je model wav2vec2. Využitie týchto zdrojov umožnilo efektívnejší tréningový proces a obišli sa obmedzenia, ktoré predstavuje lokálny hardvér z hľadiska výpočtového výkonu a kapacity pamäte.

5.1.1. Príprava prostredia Kaggle notebooku

Pre prácu v prostredí Kaggle notebooku bolo potrebné vytvorenie nového notebooku na stránke Kaggle. Notebook použitý pre prvotné trénovanie a zoznámenie sa s prostredím ako aj s modelom wav2vec2 bol nazvaný “training.ipynb”. Taktiež bolo potrebné zrýchliť výpočtové úlohy pri trénovaní modelu wav2vec2 a toho bolo dosiahnuté pomocou otvorenia konkrétneho notebooku a prechodom na bočný panel na pravej strane. Tento panel je zobrazený na Obr. 18.



Obr. 18 Panel nastavení Kaggle notebooku

Pomocou tohto panelu bolo potrebné nastaviť možnosť “Accelerator” a z rozbaľovacej ponuky vybrať grafický akcelerátor “GPU P100”. Priamo pod týmto výberom je následne ukázaná kvóta, ktorú poskytuje spoločnosť Kaggle. Zápis “00:00 / 30 hrs” predstavuje aktuálny stav využitia zdrojov GPU akcelerátora voči pridelenej kvóte spoločnosťou Kaggle. Momentálny stav zobrazený na Obr. 18 predstavuje, že doteraz neboli využité žiadne výpočtové zdroje GPU z celkovej kvóty 30 hodín, ktorá je resetovaná raz týždenne a to v sobotu v noci. Na základe stanovenej kvóty bolo potrebné dôkladné overenie parametrov a kódu pred začatím používania výpočtových zdrojov. Efektívne využitie výpočtových zdrojov GPU bolo kľúčové pre zabezpečenie trénovania čo najväčšieho možného počtu modelov s rôznymi parametrami v rámci výskumu tejto práce. Navyše od tejto kvóty je Kaggle notebook taktiež limitovaný na 12 hodín nepretržitého spustenia, po ktorých sa automaticky vypne.

Pre Kaggle notebook bolo taktiež potrebné povoliť prístup k internetu, čo zabezpečilo možnosť stiahnutia datasetu uloženého na Google disku, ako aj stiahnutie predtrénovaného modelu wav2vec2 a inštaláciu ďalších potrebných knižníc, ktoré nie sú štandardne v rámci Kaggle notebooku dostupné.

5.1.2. Inštalácia potrebných knižníc pre trénovanie

Vytvorený Kaggle notebook s vykonanými nataveniami bolo potrebné ďalej pripraviť pre prácu s modelom wav2vec2. Táto fáza zahŕňala inštaláciu potrebných knižníc jazyka Python, z ktorých každá bola vybraná pre uľahčenie procesu spracovania dát a trénovania. Inštaláciu týchto knižníc pomocou príkazu "pip install" je možné vidieť na Obr. 19.

```
!pip install datasets==2.16.1
!pip install transformers==4.33.1
!pip install jiwer==3.0.3
```

Obr. 19 Inštalácia potrebných knižníc pre trénovanie

Ako prvá knižnica bola nainštalovaná knižnica "Datasets" verzie 2.16.1. Táto knižnica je vyvíjaná spoločnosťou Hugging Face a poskytuje rozsiahlu sadu nástrojov pre načítanie, spracovanie a vyhodnotenie rôznych datasetov. Využitie tejto knižnice v rámci práce napomohlo k jednoduchému načítaniu datasetov, ich predspracovaniu a následnému využitiu pre trénovanie a vyhodnocovanie modelu wav2vec2.

Následne bola do prostredia Kaggle notebooku nainštalovaná knižnica "Transformers" verzie 4.33.1, ktorá je taktiež vyvíjaná spoločnosťou Hugging Face. Táto knižnica je známa svojou komplexnou ponukou predtrénovaných modelov špecializovaných na vykonávanie rôznych úloh NLP. Vďaka tejto knižnici bolo možné pracovať s modelom wav2vec2 a trénovať ho.

Ako posledná bola nainštalovaná knižnica "Jiwer" verzie 3.0.3, ktorá poskytuje prostriedky na vyhodnocovanie výkonnosti modelov, najmä v úlohách zahŕňajúcich prácu s rečou a textom. Táto knižnica je nápomocná pri výpočte miery chybovosti slov (WER). Miera chybovosti slov je bežná metrika, ktorá sa často používa pri rozpoznávaní reči pre meranie rozdielu medzi predikovaným textovým výstupom modelu a skutočným referenčným textom rozpoznávanej zvukovej nahrávky.

Vďaka takejto inštalácii knižníc v notebooku Kaggle bola zaručená kompatibilita kódu medzi jednotlivými trénovaniami v rôznych etapách práce pri trénovaní modelov wav2vec2 na rôzne upravených datasetoch.

5.1.3. Prvotné trénovania v prostredí Kaggle notebooku

Prvotné trénovania boli vykonávane v Kaggle notebooku s názvom "training.ipynb". Prvotné trénovania slúžili pre zoznámenie sa s trénovaním modelu wav2vec2 ako aj s prostredím Kaggle notebooku. Pre prvotné trénovania bola adoptovaná pôvodná metodológia rozdelenia datasetu a jeho použitia pri trénovaní podľa predchádzajúcej práce [16].

5.1.3.1. Príprava dát

V rámci tohto notebooku bola vykonaná príprava prostredia popísaná v kapitole 5.1.1. Po vykonaní tejto prípravy bolo potrebné prevziať augmentovaný trénovací dataset z Google disku, ktorého tvorba je popísaná v kapitole 4.4.2 a uložiť ho do priečinku "/audio_dataset/". Podobne ako bol prevzatý trénovací dataset bolo potrebné prevziať aj testovací dataset. Následne bolo potrebné tieto datasety extrahovať z archívov s názvami "train.rar" a "test.rar" do im príslušných priečinkov "/audio_dataset/train" a "/audio_dataset/test". Tento proces bol vykonaný v dvoch rozdielnych bunkách Kaggle notebooku, ktoré zobrazuje Obr. 20.

```
!gdown https://drive.google.com/u/0/uc?id=1n5APjYUAAkryhoaKhJU0RE2aLn9wo6Hs -O /audio_dataset/
!mkdir -p /audio_dataset/train/
!unrar e /audio_dataset/train.rar -C /audio_dataset/train/
```

```
!gdown https://drive.google.com/u/0/uc?id=1ehfn84lTWi4Z6_voa4FBOMOlH6qHMmzI -O /audio_dataset/
!mkdir -p /audio_dataset/test/
!unrar e /audio_dataset/test.rar -C /audio_dataset/test/
```

Obr. 20 Proces prevzatia trénovacieho a testovacieho datasetu

Na načítanie údajov potrebných pre trénovanie a testovanie modelu bola použitá knižnica Datasets. Funkcia "load_dataset" bola použitá pre načítanie datasetov priamo z ich priečinkov. Táto funkcia načítala všetky zvukové súbory ako aj príslušný "metadata.csv" súbor. Takéto načítanie vytvára dva objekty "train" a "test", ktoré je následne oveľa ľahšie analyzovať a spracovávať pred samotným trénovaním. Proces načítania datasetov zobrazuje Obr. 21.

```
from datasets import load_dataset

train = load_dataset("audiofolder", data_files={"train": "/audio_dataset/train/**"})
test = load_dataset("audiofolder", data_files={"test": "/audio_dataset/test/**"})
```

Obr. 21 Proces načítania datasetov

Po načítaní dát do objektov bolo potrebné skontrolovať správnosť ich načítania a to bolo možné vypísaním obsahu dataset objektu. Obsah dataset objektu bolo po predchádzajúcom načítaní možné vypísať príkazmi "train['train']" a "test['test']". Výstup týchto príkazov zobrazuje Obr. 22. Z výpisu je možné vidieť, že do trénovacej množiny bolo načítaných 5685 zvukových súborov spolu s ich transkripciou, čo zodpovedá počtu výrokov augmentovaného rozšíreného trénovacieho datasetu, ktorého prehľad zobrazuje Tab. 9. Výpis zobrazuje taktiež načítanú testovaciu množinu dát, do ktorej bolo načítaných 389 zvukových súborov a ich transkripcie, čo

taktiež zodpovedá počtu výrokov testovacej časti datasetu. Prehľad testovacieho datasetu, ktorý nebol v rámci práce rozširovaný zobrazuje Tab. 3.

```
[6]: train['train']

[6]: Dataset({
         features: ['audio', 'transcription'],
         num_rows: 5685
     })

[7]: test['test']

[7]: Dataset({
         features: ['audio', 'transcription'],
         num_rows: 389
     })
```

Obr. 22 Výpis obsahu dataset objektov train a test

Štruktúru načítaných dát v objektoch zobrazuje Obr. 23 na prvom zázname v množinách "train" a "test". V rámci objektu je uložená cesta k zvukovému súboru, pole reprezentujúce audio signál (vo formáte NumPy poľa so zvukovými vzorkami), vzorkovaciu frekvenciu a transkripciu pre daný súbor. V rámci práce boli načítané množiny dát podrobnejšie analyzované a bolo potrebné ich predspracovať. Ako bolo spomenuté, dataset vytvorený z relácie Táraninky v predchádzajúcich prácach obsahoval anotačné poznámky reprezentujúce rôzne zvuky v nahrávke, pri ktorých sa nejednalo o reč.

```
[8]: train['train'][0]

[8]: {'audio': {'path': '/audio_dataset/train/RLspeaker001_RL301021_001.wav',
       'array': array([0.00192261, 0.00085449, 0.00164795, ..., 0.00076294, 0.00057983,
              0.00085449]),
       'sampling_rate': 16000},
      'transcription': 'to vymysleli preto lebo aby zahnali zlých duchov zimy aby im nebolo zle'}

[9]: test['test'][0]

[9]: {'audio': {'path': '/audio_dataset/test/TRspeaker001_TR01122019_004.wav',
       'array': array([ 0.00000000e+00, -1.22070312e-04,  9.15527344e-05, ...,
               1.52587891e-04, -9.15527344e-05,  6.10351562e-05]),
       'sampling_rate': 16000},
      'transcription': 'ja sa pripravujem [fil] vyčistíme čiž [int] čižmu [int] [fil] dáme na okno
     a potom nám donesie mikuláš'}
```

Obr. 23 Štruktúra načítaných dát v dataset objektoch

Na Obr. 23 je možné vidieť výskyt poznámok ([fil], [int], [spk]) pri výpise prvého záznamu množiny “test”. Tieto poznámky ako aj všetky možné interpunkčné znamienka, ktoré by sa v dátach mohli nachádzať boli v rámci predspracovania dát odstránené pomocou vytvorenej funkcie “clean_transcription_text”. Pre ukážku transkripcie pred a po jej spracovaní bola vytvorená Tab. 10, ktorá obsahuje prvých 5 transkripcií z trénovacieho datasetu.

Tab. 10 Ukážka transkripcií pred a po spracovaní

Transkripcia pred spracovaním
ja sa pripravujem [fil] vyčistíme čiž [int] čižmu [int] [fil] dáme na okno a potom nám donesie mikuláš veľké vrecko a v ňom sladkosti
to je jeho nepriateľ
odnesie do pekla a potom ho uvaria alebo [fil] alebo bude upratovať v pekle [int]
klavír taký je [fil] že tam sa hrá palcami a potom ide taký ťahavý zvuk
Transkripcia po spracovaní
ja sa pripravujem vyčistíme čiž čižmu dáme na okno a potom nám donesie mikuláš veľké vrecko a v ňom sladkosti
to je jeho nepriateľ
odnesie do pekla a potom ho uvaria alebo alebo bude upratovať v pekle
klavír taký je že tam sa hrá palcami a potom ide taký ťahavý zvuk

5.1.3.2. Tvorba slovníka

Po príprave dát bolo potrebné vytvoriť slovník pre trénovanie modelu wav2vec2. Pre vytvorenie slovníka bola vytvorená funkcia “get_characters”. Pomocou tejto funkcie bol spracovaný trénovací a testovací dataset a boli spojené všetky reťazce transkripcií do jedného dlhého textového reťazca. Následne boli z tohto zreťazeného textu vytvorené dva zoznamy jedinečných znakov, ktoré sa nachádzali v rámci všetkých transkripcií trénovacieho a testovacieho datasetu. Na základe týchto zoznamov bol vytvorený list znakov, ktorý bol vytvorený spojením zoznamov jedinečných znakov z trénovacieho a testovacieho datasetu. Tým sa zabezpečilo, že list znakov zahrňoval všetky jedinečné znaky, ktoré boli obsiahnuté v oboch použitých datasetoch.

Tento list znakov bol následne zoradený a bol z neho vytvorený slovník, kde bolo každému znaku priradené jedinečné celočíselné ID. Išlo o štandardný proces prípravy slovníka na tokenizáciu, pri ktorom je každému jedinečnému tokenu (v tomto prípade znaku) priradené špecifické celé číslo.

Znak medzery " " bol nahradený špeciálnym oddeľovacím znakom "|". Toto nahradenie sa veľmi často používa pri práci s modelmi založených na CTC, ako je wav2vec2, na znázornenie

medzier medzi jednotlivými slovami. Do tohto slovníka boli pridané dva ďalšie špeciálne tokeny, [UNK] pre neznáme znaky a [PAD] pre výplň, pričom každému bolo priradené jedinečné celočíselné ID, ktoré pokračovalo od posledného ID už existujúcich znakov v tomto slovníku.

Dokončený slovník bol serializovaný do súboru JSON pod názvom “vocab.json”, podľa požiadaviek modelu wav2vec2. Tento spôsob uloženia taktiež zabezpečil možnosť konzistentného použitia tohto slovníka v rôznych výpočtových prostrediach pre vyhodnocovanie modelu alebo pri jeho implementácii.

5.1.3.3. Tvorba tokenizéra

Ďalším potrebným krokom bola inicializácia objektu “Wav2Vec2CTCTokenizer”. Tá bola vykonaná pomocou funkcie “Wav2Vec2CTCTokenizer.from_pretrained”, ktorú ponúka knižnica Transformers. Táto funkcia slúži primárne na načítanie už vytvoreného tokenizéra pomocou načítania jeho uložených konfiguračných súborov. V prípade, že niektoré konfiguračné súbory pri načítaní chýbajú, tak sú pre nich načítané predvolené nastavenia. Tento prístup umožnil jednoduchšiu tvorbu tokenizéra.

V rámci tvorby tokenizéra bol pomocou funkcie “Wav2Vec2CTCTokenizer.from_pretrained” načítaný vytvorený slovník “vocab.json” spolu s nasledujúcimi parametrami:

- **unk_token="[UNK]"** – Tento parameter nastavil neznámy token na "[UNK]". Je použitý na reprezentáciu znakov v texte, ktoré sa nenachádzajú v slovníku tokenizéra. To zaisťuje, že model dokáže spracovávať neznáme znaky.
- **pad_token="[PAD]"** – Parameter nastavil token výplne na "[PAD]". Tento token sa používa na štandardizáciu dĺžky všetkých sekvencií spracovaných modelom. Takáto jednotnosť je nevyhnutná na účely spracovania po dávkach (angl. batching) počas tréningu a hodnotenia modelu.
- **word_delimiter_token="|"** – Token pre oddelenie slov bol nastavený na "|". Slúži na zreteľné oddelenie slov v spracovaných sekvenciách. Takéto rozdelenie je dôležité pri rozpoznávaní reči, aby sa zachovala integrita začiatkov a koncov slov po tokenizácii.

5.1.3.4. Tvorba extraktora príznakov

Následne bolo potrebné vytvoriť extraktor príznakov (angl. feature extractor) pomocou inicializácie objektu “Wav2Vec2FeatureExtractor”. Tvorba extraktora príznakov bola potrebným krokom pre proces predspracovania audio dát pre model wav2vec2.

Inicializácia bola vykonaná použitím funkcie “Wav2Vec2FeatureExtractor” z knižnice Transformers s nasledujúcimi parametrami:

- **feature_size=1** – Tento parameter znamená, že každý príznak v audio signáli je jednorozmerný, priamo zodpovedajúci hodnotám amplitúdy v každom bode vzorkovania.
- **sampling_rate=16000** – Tento parameter nastavuje vzorkovaciu frekvenciu dát na 16000 Hz. Toto je bežná vzorkovacia frekvencia pre úlohy rozpoznávania reči, pretože vyvažuje kvalitu zvuku a výpočtovú efektívnosť. Pri nastavení tohto parametru bolo dôležité, aby nastavená vzorkovacia frekvencia pre extraktor príznakov bola zhodná s vzorkovacou frekvenciou očakávaných vstupných dát modelu.
- **padding_value=0.0** – Parameter nastavuje hodnotu výplne, ktorá sa využíva na prispôsobenie variability dĺžok zvukových segmentov v rámci dávky (angl. batch). Kratšie segmenty sú predĺžené tak, aby zodpovedali dĺžke najdlhšieho segmentu. Na tento účel bola použitá výplňová hodnota 0.0, ktorá predstavuje ticho, čím sa zaistila jednotná dĺžka vstupných segmentov bez zavedenia skreslenia.
- **do_normalize=True** – Týmto parametrom bola povolená normalizácia zvukových dát, čo znamená, že amplitúda zvukových signálov bola škálovaná medzi jednotlivými súbormi pre konzistentnosť a zníženie variability.
- **return_attention_mask=True** – Tento parameter povolil generáciu masiek pozornosti (angl. attention masks) popri spracovaných audio príznakoch. Masky pozornosti indikujú, ktoré časti vstupu sú skutočnými zvukovými dátami oproti výplni. To umožňuje modelu zamerať sa na relevantné časti zvukového signálu a ignorovať oblasti obsahujúce výplň, čo je dôležité pri trénovaní s použitím audio segmentov rôznej dĺžky.

5.1.3.5. Tvorba procesora

Po vytvorení tokenizéra a extraktora príznakov bolo prostredie pripravené na vytvorenie procesora. Procesor bol vytvorený pomocou funkcie “Wav2Vec2Processor”, ktorú taktiež ponúka knižnica Transformers. Tento procesor kombinuje extraktor príznakov a tokenizér do jedného objektu, ktorý zjednodušuje prípravu zvukových dát na spracovanie modelom wav2vec2.

5.1.3.6. Predspracovanie dát

Ďalším krokom po vytvorení procesora bolo využitie vytvoreného procesora pre predspracovanie dát. Predspracovanie zahŕňalo transformáciu vstupných audio súborov a transkripcií do formátu vhodného pre použitie modelom. Pre túto transformáciu bola vytvorená funkcia “prepare_dataset”, ktorou boli spracované všetky zvukové a transkripčné dáta trénovacej a testovacej množiny. Metodika predspracovania dát bola nasledovná:

- **Predspracovanie zvukových dát** – Funkcia extrahovala zvukové informácie zo súborov s využitím vopred vytvoreného procesora. Transformácia pomocou procesora zahŕňala

konverziu zvukových záznamov na sekvenciu číselných hodnôt, ktoré predstavovali príznaky zvuku a boli vstupnými hodnotami pre model. Procesor taktiež zabezpečil, že zvukové dáta prechádzali normalizáciou.

- **Tokenizácia transkripcií** – Funkcia taktiež spracovávala transkribované texty príslušných zvukových súborov. Pomocou tokenizačnej časti procesora boli transkripcie tokenizované. Tento proces pozostával z konverzie nespracovaných transkripcií na sekvenciu tokenových identifikátorov. Tieto identifikátory alebo štítky (angl. labels) sú pre model potrebné pri učení sa korelácií medzi hovorenými slovami a ich textovými reprezentáciami.

Následne boli odstránené mená stĺpcov dát po spracovaní, čo zaručilo, že dáta obsahovali výlučne spracované vstupné hodnoty a ich štítky, ktoré boli pripravené na ďalšie spracovanie.

5.1.3.7. Proces pridania výplne pomocou kolektora dát

Keďže vstupné hodnoty v rámci jednej dávky mohli byť rôznych dĺžok, bolo potrebné ich spracovanie pomocou pridania výplne. Pre toto spracovanie bol použitý kolektor dát (angl. data collator) “DataCollatorCTCWithPadding”, ktorý bol navrhnutý na použitie s modelmi, ktoré pri úlohách rozpoznávania reči využívajú CTC (Connectionist Temporal Classification). Podstata tohto kolektora dát spočíva v schopnosti dynamicky upravovať dĺžku vstupných sekvencií zahŕňajúcich audio dáta a ich zodpovedajúce spracované textové prepisy na jednotnú dĺžku v celej dávke dát. Takáto úprava je kľúčová pre efektívne spracovanie dát v dávkach modelom wav2vec2, keďže vstupné sekvencie sa líšia v dĺžke, pretože trvanie hovoreného obsahu v jednotlivých segmentoch je rôzne.

Kolektor bol inicializovaný spolu s vytvoreným procesorom, ktorý mal za úlohu extrakciu zvukových príznakov a tokenizáciu transkripcií. Stratégia výplne bola nakonfigurovaná tak, aby boli všetky sekvencie v rámci jednej dávky vyplnené podľa najdlhšej sekvencie v tejto dávke. Vytvorený kolektor dát bol neskôr využívaný v rámci trénovacieho procesu, kde bol volaný v inštancii Trainer.

5.1.3.8. Metrika hodnotenia

Pre určenie presnosti modelu wav2vec2 bolo dôležité vypočítať jeho WER (Word Error Rate), čo je bežná metrika používaná v doméne rozpoznávania reči. Hodnota WER predstavuje chybovosť transkripcie, pričom indikuje percento slov, ktoré boli modelom nesprávne predpovedané.

Na vytvorenie procesu vyhodnocovania bola využitá funkcia “load_metric” z knižnice Datasets, ktorá načítala relevantnú metriku pre výpočet WER. Následne bola definovaná funkcia

“compute_metrics”, určená na výpočet WER na základe predikcií modelu. Táto funkcia vykonávala niekoľko krokov pri spracovaní a vyhodnocovaní predikcií:

- **Extrakcia a dekódovanie predikcií** – Prvotne boli extrahované logity (angl. logits) predstavujúce predikcie modelu. Tieto logity boli následne spracované prostredníctvom funkcie argmax, aby boli získané najpravdepodobnejšie ID tokenov pre každý časový úsek v sekvencii. Paralelne funkcia upravovala štítky výplne, ktorým kolektor dát priradil hodnotu -100. Tieto hodnoty boli nahradené tokenom "|", ktorý tokenizér používa pre označenie výplne. Táto úprava bola potrebná pre presný výpočet metriky WER pri porovnávaní predikcií s originálnymi štítkami.
- **Dekódovanie predikcií a štítkov** – Predikované a originálne ID tokenov boli dekódované späť do textových reťazcov pomocou metódy “processor.batch_decode”. Pri dekódovaní originálnych štítkov bol použitý parameter “group_tokens=False”, ktorý zaisťoval, že nenastalo žiadne zoskupenie tokenov, čím sa zachovala integrita pôvodného textu transkripcií pre spravodlivé porovnanie.
- **Výpočet WER** – Miera chybovosti slov bola vypočítaná porovnaním dekódovaných predikovaných reťazcov s dekódovanými reťazcami originálnych štítkov. Toto porovnanie poskytlo kvantitatívnu mieru presnosti modelu, ktorá odrážala podiel chýb v predikovaných transkripciách v porovnaní so skutočnými transkripciami.

5.1.3.9. Inicializácia a konfigurácia modelu

Ďalším dôležitým krokom bola inicializácia a konfigurácia Wav2Vec2ForCTC modelu pomocou knižnice Transformers. Model bol načítaný s predtrénovanými váhami z kontrolného bodu (angl. checkpoint) “facebook/wav2vec2-xls-r-300m” spolu so špecifickými parametrami. Tento kontrolný bod bol vybraný na základe kompromisu medzi výkonnosťou a výpočtových požiadaviek. Využitie väčších verzií XLS-R modelu by nebolo možné v rámci trénovania na platforme Kaggle. Taktiež využitie menšieho modelu šetrí pamäť potrebnú pre jeho nasadenie a jeho dekódovanie môže trvať kratšie ako pri využití väčších verzií. Táto verzia ako aj iné sú popísané v kapitole 3.4. Inicializáciu a konfiguráciu načítavaného modelu zobrazuje Obr. 24.

```python
from transformers import Wav2Vec2ForCTC

model = Wav2Vec2ForCTC.from_pretrained(
    "facebook/wav2vec2-xls-r-300m",
    attention_dropout=0.0,
    hidden_dropout=0.0,
    feat_proj_dropout=0.0,
    layerdrop=0.0,
    # SpecAugment configuration
    apply_spec_augment=True,
    mask_time_prob=0.5,
    mask_time_length=20,
    mask_time_min_masks=2,
    mask_feature_prob=0.5,
    mask_feature_length=80,
    mask_feature_min_masks=0,
    ctc_loss_reduction="mean",
    pad_token_id=processor.tokenizer.pad_token_id,
    vocab_size=len(processor.tokenizer),
)
```

Obr. 24 Inicializácia a konfigurácia wav2vec2 modelu

Konfigurácia modelu Wav2Vec2ForCTC zahŕňala viacero parametrov a možných techník:

- **Techniky regularizácie** – Aby bolo znížene riziko nadmerného preučenia sa, model poskytuje možnosť využitia rôznych vyraďovacích (angl. dropout) techník. V rámci experimentov boli hodnoty regularizačných techník počas trénovania rôznych modelov menené v rozsahu 0.0 – 0.1, čo predstavuje menej agresívne nastavenie.
- **Aplikácia SpecAugment** – Model podporoval augmentačnú metódu SpecAugment, ktorá zavádza náhodné maskovanie vstupov spektrogramu, čím zvyšovala schopnosť modelu zovšeobecňovať. V rámci najväčšieho počtu experimentov konfigurácia zahŕňala 50% pravdepodobnosť časového maskovania, pričom každá z týchto masiek mala dĺžku 20 časových krokov a boli použité minimálne 2 takéto masky. Okrem toho bolo nakonfigurované aj maskovanie frekvencií s pravdepodobnosťou 50% a dĺžkou masky až 80 prvkov, pričom nebola stanovená minimálna požiadavka pre počet týchto masiek. Tieto hodnoty boli počas experimentov mierne menené a boli taktiež vykonané dotrénovania modelu, kde bola aplikácia augmentačnej metóda SpecAugment úplne vypnutá.
- **Konfigurácia CTC straty** – Metóda agregácie CTC straty bola nastavená na “mean”, čím bol štandardizovaný výpočet straty v rôznych veľkostiach dávok a zaručená konzistentná aktualizácia parametrov počas trénovania.

- **Kompatibilita slovníka a tokenizácie** – ID tokenu pre výplň bolo zosúladené s tokenizérom procesora, čím bolo zabezpečené jednotné spracovanie vstupných sekvencií s rôznou dĺžkou. Okrem toho bola veľkosť slovníka prispôsobená slovníku tokenizéra pre zaručenie správnej interpretácie zakódovaných dát.

Po konfigurácii Wav2Vec2 modelu s využitím kontrolného bodu “facebook/wav2vec2-xls-r-300m” bola použitá metóda “model.freeze_feature_encoder()” podľa odporúčania odborného výskumu [3]. Táto metóda zmrazila parametre kódovača príznakov, čím zabránila ich aktualizácii počas ďalších fáz trénovania. Vďaka tomuto zmrazeniu sú zachované robustné časti modelu, ktoré boli vopred natrénované na spracovanie zvukových dát. Zmrazenie taktiež sústredí ďalšie dotrénovanie na vyššie vrstvy modelu, ktoré sú špecifické pre danú úlohu ASR. Táto stratégia urýchľuje proces trénovania znížením počtu trénovateľných parametrov a zároveň zachováva naučené robustné vlastnosti modelu z fáze predtrénovania.

5.1.3.10. Definovanie trénovacích parametrov

Ďalším potrebným krokom bola definícia trénovacích parametrov vykonaná pomocou knižnice Transformers využitím triedy “TrainingArguments” pre vytvorenie objektu “training_args”. Táto konfigurácia bola potrebná pre vyladenie trénovania modelu pre dosiahnutie optimálneho výkonu. Jej parametre boli nastavené nasledovne:

- **Výstupný priečinok** – Výstupný priečinok bol definovaný ako "./results/wav2vec2-XLS-R-SK-CHILD-ADV-v0.5", kde boli uchovávané výsledky tréningu a kontrolné body modelu.
- **Konfigurácia dávkovania** – Dávky podobnej dĺžky boli zoskupené pre vyššiu efektívnosť výpočtov znížením množstva potrebnej výplne. Veľkosť trénovacej dávky na jedno zariadenie bola nastavená na 4, pričom akumulácia gradientu bola nastavená na 4 kroky. Takéto nastavenie akumulácie gradientu na 4 kroky kombinuje gradienty zo 4 minidávok pred aktualizáciou váh modelu, čím sa simuluje efektívna veľkosť dávky 16 na jednom zariadení. Tým sa optimalizuje trénovanie s obmedzenými pamäťovými zdrojmi bez priameho zvýšenia skutočnej veľkosti dávky.
- **Stratégia trénovania a hodnotenia** – Trénovací proces bol nastavený tak, aby bol model vyhodnotený a bol mu vytvorený kontrolný bod každých 200 krokov. Počet trénovacích epoch bol medzi jednotlivými trénovaniami menený v rozsahu 5 až 80.
- **Optimalizačné techniky** – Boli povolené kontrolné body gradientu a trénovanie so zmiešanou presnosťou (fp16), ktoré výrazne znížili spotrebu pamäte a urýchlili trénovanie.

- **Rýchlosť učenia a zahrievanie** – Počiatočná rýchlosť učenia bola menená v rozsahu 0.000025 až 0.002 s nastavenou fázou zahrievania 500 krokov pre stabilizáciu dynamiky učenia.
- **Kritéria ukladania modelu –** Trénovací proces bol nakonfigurovaný tak, aby udržiaval až 4 uložené kontrolné body a aby na konci trénovania načítal najlepší model na základe metriky WER, pričom uprednostňuje nižšie hodnoty vďaka parametru "greater_is_better=False".

Definovanie trénovacích parametrov je zobrazené na Obr. 25.

```
from transformers import TrainingArguments

training_args = TrainingArguments(
  output_dir="./results/wav2vec2-XLS-R-SK-CHILD-ADV-v0.5",
  group_by_length=True,
  per_device_train_batch_size=4,
  gradient_accumulation_steps=4,
  evaluation_strategy="steps",
  num_train_epochs=15,
  gradient_checkpointing=True,
  fp16=True,
  save_steps=200,
  eval_steps=200,
  logging_steps=200,
  learning_rate=0.0001,
  warmup_steps=500,
  save_total_limit=4,
  load_best_model_at_end=True,
  metric_for_best_model="wer",
  greater_is_better=False,
)
```

Obr. 25 Definovanie trénovacích parametrov

5.1.3.11. Konfigurácia trénovacieho procesu

Posledným potrebným krokom pred spustením trénovania bola konfigurácia trénovania pomocou triedy "Trainer" z knižnice Transformers. Vytvorenie objektu "trainer" spolu s jeho parametrami zobrazuje Obr. 26.

```
from transformers import Trainer

trainer = Trainer(
    model=model,
    data_collator=data_collator,
    args=training_args,
    compute_metrics=compute_metrics,
    train_dataset=train,
    eval_dataset=test,
    tokenizer=processor.feature_extractor,
)
```

Obr. 26 Parametre konfigurácie objektu “trainer”

Parametre tohto objektu integrovali model s predtrénovanými váhami prostredníctvom parametra “model” a kolektor dát prostredníctvom parametra “data_collator”. Parametru “args” boli priradené nastavenia hyperparametrov vykonané vo vytvorenom objekte “training_args”. Na vyhodnotenie presnosti modelu bola využitá vytvorená funkcia “compute_metrics”.

Parametre “train_dataset” a “eval_dataset” definujú časti datasetu pre trénovanie a validáciu. Ako už bolo spomenuté, prvotné trénovania pre zoznámenie sa s trénovacím procesom využívali rovnakú metodológiu rozdelenia datasetov ako predošlá práca [16]. Tým pádom boli prvotné modely pomenované “wav2vec2-XLS-R-SK-CHILD-ADV-vX.X”, kde X.X predstavuje číselné označenie verzie trénované na celom trénovacom datasete a validované na testovacom datasete. Vďaka takémuto rozdeleniu bolo možné odsledovať dôsledky rozšírenia trénovacieho datasetu pomocou dát získaných z rádiovej relácie Rozhlasové leporelo.

5.1.3.12. Spustenie trénovacieho procesu a uloženie natrénovaného modelu

Trénovanie modelu bolo spustené príkazom “trainer.train()”, ktorý využil definované konfigurácie a rozdelenie súborov. Úspešne ukončený proces trénovania jedného z modelov a jeho metriky zobrazuje Obr. 27.

[3200/3200 4:16:37, Epoch 35/35]

Step	Training Loss	Validation Loss	Wer
200	8.284100	4.058145	1.000000
400	3.420800	3.211784	1.000000
600	3.180700	3.153908	1.000000
800	3.106000	2.798738	0.999819
1000	2.165000	0.972262	0.702024
1200	1.561700	0.622439	0.491326
1400	1.328500	0.487412	0.410192
1600	1.179800	0.426133	0.346223
1800	1.106000	0.394869	0.335020
2000	1.034200	0.380088	0.311529
2200	0.988200	0.346251	0.300325
2400	0.950500	0.338688	0.287676
2600	0.905700	0.340323	0.285327
2800	0.886000	0.339100	0.286050
3000	0.867000	0.332027	0.274846
3200	0.855000	0.329079	0.274304

Obr. 27 Ukončený proces trénovania

Po ukončení fázy trénovania boli natrénované modely ukladané pomocou funkcie “trainer.save_model()”, ktorá ukladá naučené váhy a konfiguráciu modelu do určeného priečinka. Tým sa zabezpečí, že model možno bez problémov implementovať s jeho dosiahnutými výsledkami. Taktiež bola použitá funkcia “processor.save_pretrained()” na uloženie konfigurácie procesora, ktorá zahŕňa komponenty extrakcie príznakov aj tokenizácie. Tento krok je kľúčový pre zachovanie konzistentnosti dát, čo umožňuje presnú reprodukciu krokov predspracovania dát počas vyhodnocovania modelu, čím sa zabezpečí integrita a použiteľnosť výsledkov výskumu.

5.2. Integrácia jazykového modelu

V snahe zvýšenia presnosti predikcie trénovaného modelu wav2vec2 bol integrovaný 3-gramový jazykový model (z angl. skratka LM), pričom bol využitý rovnaký jazykový model, aký bol použitý v predchádzajúcej práci [16]. Toto rozhodnutie vyplynulo z preukázanej účinnosti tohto LM pri zlepšovaní výsledkov rozpoznávania reči v podobnej integrácii. 3-gramový model, známy svojou schopnosťou predpovedať pravdepodobnosť sekvencie troch slov, poskytuje cenné

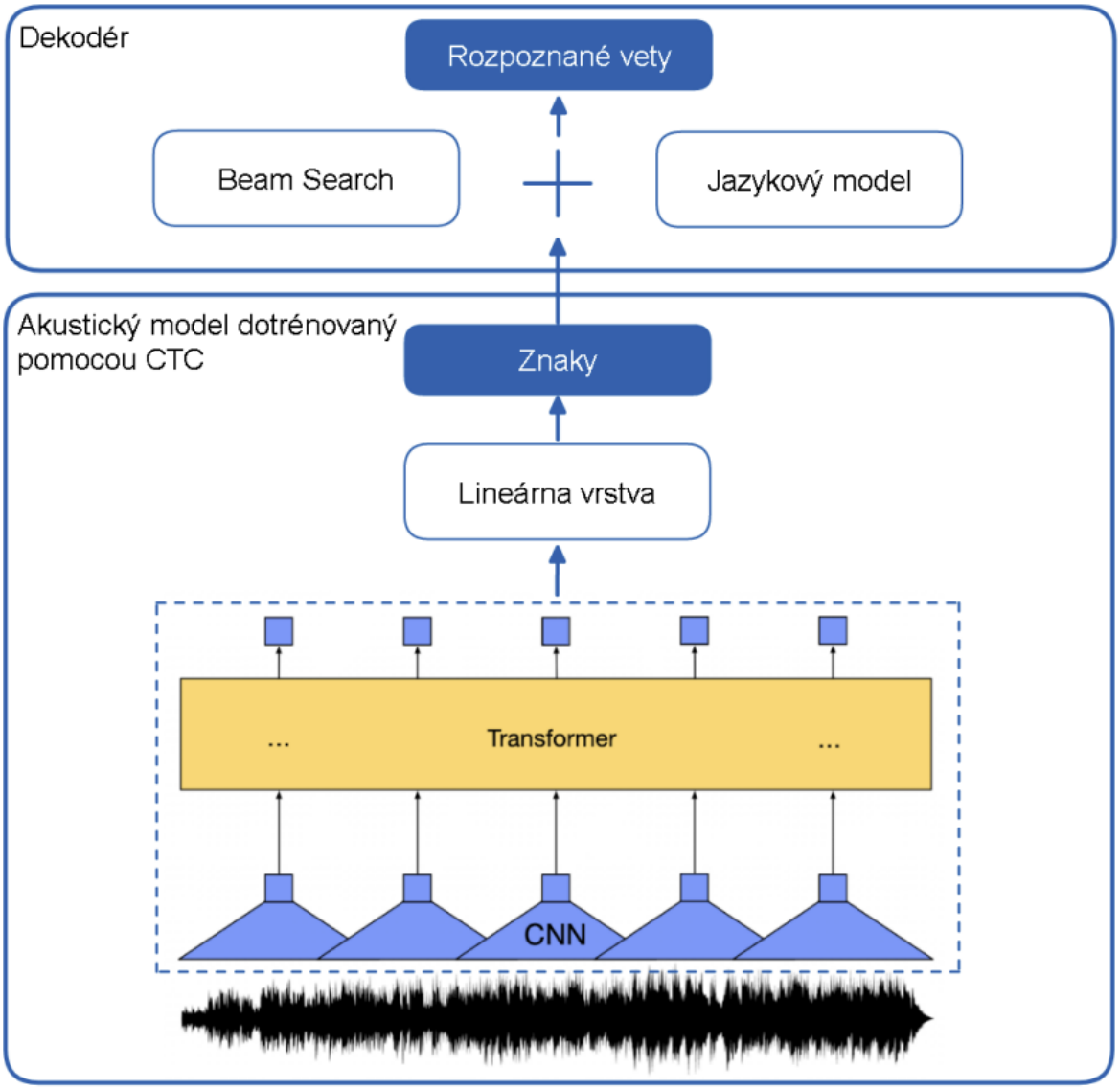
kontextové informácie, ktoré významne pomáhajú pri rozlíšení rečových signálov, čo vedie k presnejším a koherentnejším predikciám.

Jazykový model použitý v tejto integrácii bol pôvodne vyvinutý a doladený v samostatnej štúdii [30], kde preukázal podstatné zlepšenie výkonu ASR. Na prispôsobenie tohto LM na použitie s modelom wav2vec2 bol prevedený z pôvodného formátu ARPA do binárneho formátu pomocou toolkitu KenLM. Tento binárny formát znižuje požiadavky na úložný priestor a umožňuje rýchlejšie načítanie a vyhľadávanie dát jazykového modelu počas procesu rozpoznávania reči, čím bola zvýšená celková efektívnosť pri použití s modelom wav2vec2.

Integrácia spočívala v priložení súborov LM do priečinka, ktorý obsahoval súbory dotrénovaného wav2vec2 modelu. Pri vyhodnotení výsledkov modelov s využitím LM bolo potrebné použiť metódu "Wav2Vec2ProcessorWithLM.from_pretrained" na načítanie predtrénovaného procesora spoločne s integrovaným jazykovým modelom.

5.2.1. Dekódovanie pomocou jazykového modelu

Pri implementácii jazykového modelu s modelom wav2vec2 z knižnice Hugging Face Transformers na rozpoznávanie reči bola použitá knižnica PyTCDecode na podporu lúčového vyhľadávania (angl. beam search). Toto nastavenie umožňuje vylepšiť predikcie pomocou lúčového vyhľadávania a jazykového modelu. Lúčové vyhľadávanie pomáha udržiavať viaceré hypotézy pre rečový vstup, zatiaľ čo jazykový model poskytuje kontext. Každá hypotéza alebo "lúč" sa skladá z potenciálnej postupnosti slov. Tieto sekvencie sú hodnotené na základe ich pravdepodobnosti podľa výstupu modelu a kontextového porozumenia jazykového modelu. Hodnotiaca funkcia zvyčajne kombinuje tieto pravdepodobnosti na zoradenie sekvencií, čo umožňuje systému postupne vybrať najpravdepodobnejšiu sekvenciu. Táto metodika zabezpečuje, že konečná transkripcia je nielen priamym odrazom hovoreného obsahu, ale je aj kontextovo vhodná a jazykovo koherentná. Blokovú schému dotrénovaného wav2vec2 modelu zobrazujúcu dekódovanie s integráciou lúčového hľadania a jazykového modelu je možné vidieť na Obr. 28.



Obr. 28 Dekódovanie modelom wav2vec2 s integráciou lúčového hľadania a jazykového modelu [16]

5.3. Proces vyhodnotenia modelov

Všetky výsledky v rámci práce boli získane pomocou vytvorených Jupyter Notebookov pre vyhodnotenie modelov na validačnej a testovacej množine bez a s využitím LM. Tieto notebooky boli spustené v prostredí Google Colab, vďaka čomu bola jednoduchšia integrácia samotných modelov pomocou využitia Google disku a zároveň takéto vyhodnotenie nečerpalo z kvóty na trénovanie v prostredí spoločnosti Kaggle.

Celkovo boli vytvorené štyri Jupyter Notebooky pre vyhodnotenie modelov: “Evaluate_on_valid”, “Evaluate_on_valid_with_LM”, “Evaluate_on_test” a “Evaluate_on_test_with_LM”. Tieto súbory boli uložené do priloženého priečinka s názvom “Evaluation_scripts”.

5.3.1. Príprava prostredia a spracovanie dát

Proces hodnotenia zahŕňal integráciu Google disku do prostredia Google Colab. Táto integrácia bola potrebná pre prístup k validačnej a testovacej množine dát, modelu a jeho súborom uloženým na Google disku. Ďalším krokom bola inštalácia potrebných knižníc na manipuláciu s datasetom, prácu s modelom, dekódovanie pomocou jazykového modelu a hodnotenie presnosti. Inštalované knižnice spoločne s ich verziami zobrazuje Obr. 29.

```
!pip install datasets==2.16.1
!pip install transformers==4.35.2
!pip install -U --no-cache-dir gdown==5.1.0
!pip install https://github.com/kpu/kenlm/archive/master.zip pyctcdecode==0.5.0
!pip install jiwer==3.0.3
```

Obr. 29 Inštalácia potrebných knižníc pre vyhodnotenie

Ďalším krokom bolo prevzatie validačnej alebo testovacej množiny podľa momentálneho vyhodnocovania. Tieto časti datasetu boli komprimované a uložené na Google disku odkiaľ boli skopírované do lokálneho prostredia notebooku Google Colab a extrahované. Následne boli transkripcie množiny použitej na vyhodnotenie spracované aby neobsahovali žiadnu možnú interpunkciu a anotačné poznámky, ktoré by mohli skresliť hodnotenie presnosti modelu.

5.3.2. Načítanie modelu a príprava na generovanie predikcií

Po načítaní a spracovaní dát bola načítaná verzia momentálne hodnoteného dotrénovaného wav2vec2 modelu spolu s príslušným procesorom. Pri hodnotení bol model každej verzie načítaný v dvoch rôznych konfiguráciách a to bez využitia jazykového modelu, pre posúdenie jeho základnej presnosti a následne s integrovaným jazykovým modelom, pre odsledovanie a zlepšenie jeho celkovej presnosti.

Na spracovanie zvukových súborov prostredníctvom modelu bola definovaná predikčná funkcia. Táto funkcia vykonávala úlohu generovania transkripcie zo zvukového vstupu, čím sa efektívne simulovalo reálne použitie modelu pri prepisovaní detskej reči.

5.3.3. Generovanie predikcií a výpočet WER

Spracovaná množina dát následne prešla procesom hodnotenia pre výpočet presnosti. Každý zvukový súbor tejto množiny bol spracovaný prostredníctvom modelu na generovanie sady transkripčných predikcií, ktoré boli následne porovnané s ručne vytvorenými transkripciami. Ukážku štyroch náhodných predikcií vygenerovaných pomocou modelu spolu s ich reálnymi transkripciami zobrazuje Obr. 30. Metrikou pre toto porovnanie bola miera chybovosti slov (WER).

Výsledky tohto hodnotenia boli pre každý model zaznamenané a použité pre konečné vyhodnotenie presnosti najlepších modelov.

```
Predikcia:      vytvarné umenie je napríklad nejaké umelecké dielo
Transkripcia:   výtvarné umenie je napríklad nejaké umelecké dielo

Predikcia:      listnatý ihličnatý a to je asi všetko
Transkripcia:   listnatý ihličnatý a to je asi všetko

Predikcia:      čo bol váš najťahší výrobok vyrobiť
Transkripcia:   čo bol váš najťažší výrobok vyrobiť

Predikcia:      ja si myslím že mimozemšťania môžu existovať ale ešte s toro nauti to nezistili
Transkripcia:   ja si myslím že mimozemšťania môžu existovať ale ešte astronauti to nezistili
```

Obr. 30 Ukážka predikcií a ich reálnych transkripcií

5.3.4. Výsledky prvotných trénovaní

Pri hodnotení prvotných trénovaní modelu wav2vec2 bola využitá rovnaká metodológia, ako v predchádzajúcej práci z dôvodu odsledovania vplyvu rozšírenia datasetu novými dátami z relácie Rozhlasové leporelo. V rámci predchádzajúcej práce boli modely trénované na celej trénovacej množine, zatiaľ čo na validáciu, výber a vyhodnotenie modelu bola použitá testovacia množina. Najlepší výsledok prvotných trénovaní s porovnaním najlepšieho modelu predošlej práce zobrazuje Tab. 11. Takéto trénovanie a hodnotenie nie je optimálne a jeho nedostatky popisuje kapitola 5.5.

Tab. 11 Porovnanie výsledkov prvotných trénovaní s predošlou štúdiou

Názov modelu	**Dátová augmentácia**	**Množstvo trénovacích dát (hh:mm)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**
wav2vec2-XLS-R-SK-CHILD-v8.2 (predošlá štúdia) [16]	O+SP+SA	7:43	**30,11**	**18,13**
wav2vec2-XLS-R-SK-CHILD-ADV-v5.0	O+SP+SA	11:04	**25,34**	**16,6**

Poznámka: V tabuľke vyššie "O" predstavuje originálne dáta. "SP" označuje, že dáta boli rozšírené pomocou perturbácie rýchlosti. "SA" značí využitie techniky SpecAugment, ktorá bola aplikovaná počas trénovania.

5.4. Využitie modelu na opravu transkripcie testovacej množiny

V tejto fáze projektu bol použitý natrénovaný model “wav2vec2-XLS-R-SK-CHILD-ADV-v5.0” ako nástroj na úpravu presnosti manuálnej transkripcie dát testovacej množiny. Tento prístup bol

založený na porovnaní automatických predikcií modelu s existujúcimi manuálnymi transkripciami zvukových nahrávok.

Metodika zahŕňala spustenie natrénovaného modelu na zvukových vzorkách z testovacej množiny s cieľom vytvoriť automatické predikované transkripcie. Tieto predikcie boli následne systematicky porovnávané s ich manuálnymi transkripciami. Cieľom tohto porovnania bolo identifikovať dve hlavné kategórie nezrovnalostí: prvé zahŕňali situácie, kde manuálna transkripcia obsahovala rozdielny počet slov v porovnaní s predikciami modelu, a druhé prípady, kde sa jednotlivé manuálne transkribované slová gramaticky alebo významovo líšili od výstupov modelu. Prítomnosť ďalších slov v manuálnej transkripcii, ktoré sa v predpovediach modelu nenachádzali, naznačila potenciálne chyby v procese manuálnej transkripcie alebo spracovaní audio nahrávok. Naopak, rozdiely jednotlivých slov medzi manuálnou transkripciou a predikciami modelu upozorňovali na oblasti, v ktorých mohol manuálny proces zachytiť zvuk nepresne.

Po zistení týchto rozdielov sa vykonala kritická analýza s cieľom určiť pravdivosť manuálnych transkripcií v porovnaní s predikciami modelu. V rámci tejto analýzy boli porovnané manuálne transkripcie spolu s ich predikciou a následne pri nájdených nezrovnalostiach boli vypočuté zvukové nahrávky príslušné daným transkripciám. Na základe manuálnej kontroly týchto zvukových nahrávok bolo určené, či sa naozaj jednalo o chybu manuálnej transkripcie alebo chybu predikcie modelu. V prípade zistenia chyby v manuálnej transkripcii bola transkripcia upravená tak, aby presne odrážala slová dieťaťa na nahrávke. V rámci trénovacej množiny boli nájdené a opravené chyby v manuálnych transkripciách 35 rôznych audio nahrávok z celkového počtu 389. Všetky nájdené chyby a vykonané zmeny boli zdokumentované v priloženom textovom súbore “testmetadata_log.txt”.

Tento proces spresňovania manuálnej transkripcie pomocou predikcií modelu wav2vec2 zdôrazňuje hodnotu modelu nielen ako nástroja pre predikciu novej transkripcie, ale aj ako prostriedku na overenie a zlepšenie už existujúcej manuálnej transkripcie. Využitím modelu bolo možné zvýšiť presnosť ľudskej transkripcie testovacej množiny, čo ukazuje jedno z mnohých možných praktických využití tohto modelu. Vplyv opravených chýb testovacej množiny na výsledky modelov popísaných v kapitole 5.3.4 zobrazuje Tab. 12.

Tab. 12 Porovnanie WER pomocou pôvodnej a opravenej testovacej množiny

	Pôvodná testovacia množina		Opravená testovacia množina	
Názov modelu	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**
wav2vec2-XLS-R-SK-CHILD-v8.2 (predošlá štúdia)	**30,11**	**18,13**	**28,18**	**16,91**
wav2vec2-XLS-R-SK-CHILD-ADV-v5.0	**25,34**	**16,6**	**24,35**	**15,15**

5.5. Analýza metodológie prvotných trénovaní

V počiatočných fázach práce bola použitá metodológia trénovania inšpirovaná predchádzajúcou prácou v tejto oblasti, konkrétne na trénovania wav2vec2 modelov v tejto práci. Tento prístup zahŕňal využitie celého dostupného datasetu na účely trénovania a následné použitie testovacej množiny na validáciu počas fázy vývoja modelu a zároveň pre vyhodnotenie konečnej presnosti modelu. Takýmto spôsobom bol výber modelu ovplyvnený výsledkami priamo na testovacej množine, čo zaviedlo skreslenie (angl. bias) výsledkov.

Po dôkladnom preskúmaní a analýze tejto metodológie a jej zosúladení s osvedčenými postupmi pri vývoji modelov strojového učenia bola identifikovaná kritická oblasť na zlepšenie. Tradične rozdelenie údajov do samostatných trénovacích, validačných a testovacích množín slúži na zabezpečenie toho, aby sa model nielen efektívne učil z trénovacích dát, ale aj dobre zovšeobecňoval na nové, doposiaľ nevidené dáta. Validačná množina zohráva kľúčovú úlohu pri ladení a výbere modelu bez toho, aby bola ohrozená integrita testovacej množiny, ktorá je určená na objektívne vyhodnotenie presnosti modelu.

Po zistení dôležitosti tohto oddelenia pre zovšeobecniteľnosť trénovaného modelu, bola použitá metodológia upravená. Vykonanou úpravou bolo rozdelenie trénovacej časti datasetu na trénovaciu a validačnú časť. Toto rozdelenie bližšie popisuje kapitola 5.6. Táto úprava umožnila presnejšie vyladenie hyperparametrov, prijímanie informovaných rozhodnutí o úpravách modelu a lepšie posúdenie presnosti modelu počas fázy vývoja, a to všetko pri zachovaní testovacej množiny pre jej zamýšľaný účel, ktorým je poskytnúť nezaujaté hodnotenie konečného modelu.

Toto zdokonalenie použitej metodológie je v súlade so zavedenými osvedčenými postupmi a zlepšuje proces trénovania modelu, čím sa zabezpečuje, že zistenia v tejto práci a uvádzané ukazovatele presnosti sú spoľahlivé a odrážajú skutočnú schopnosť modelu zovšeobecniť sa na

doposiaľ nevidené dáta. Existuje predpoklad, že táto úprava významne prispieva k dôveryhodnosti tejto práce a sú dodržané štandardy metodológie výskumu v rámci strojového učenia.

5.6. Rozdelenie dát pre vytvorenie datasetu s validačnou množinou

Dataset určený na trénovanie modelu wav2vec2 sa skladal zo zvukových záznamov detskej reči a ich transkripcií, ktoré pochádzali z dvoch rôznych relácií: relácie Táraninky a rádiovej relácie Rozhlasové leporelo. Prehľady jednotlivých častí tohto datasetu zobrazujú Tab. 3 a Tab. 7. Vzhľadom na absenciu vopred existujúcej validačnej množiny a jej nevyhnutnosť v procese trénovania modelu bolo pomocou Python skriptov vykonané rozdelenie trénovacích dát pre jej vytvorenie.

Proces začal samostatným spracovaním každého zdroja, aby sa zachovali vlastnosti a pomer datasetov. Z trénovacej množiny datasetu relácie Táraninky bolo 80% dát pridelených na účely trénovania a zvyšných 20% bolo určených na vytvorenie validačných dát. Toto rozdelenie bolo rovnako použité aj pre rádiovú reláciu Rozhlasové leporelo, pričom sa použil rovnaký pomer rozdelenia 80/20, aby sa zabezpečila konzistentnosť spracovania dát z oboch zdrojov.

Po individuálnom rozdelení každej relácie nasledoval krok, ktorý zahŕňal zlúčenie týchto oddelených častí do jednotnej trénovacej a validačnej množiny rozšíreného datasetu. Toto zlúčenie bolo vykonané tak, aby sa zachovalo zastúpenie jednotlivých zdrojov v rámci kombinovaného datasetu. Kombinovaný dataset odrážal pomerný príspevok jednotlivých relácií, čím bola zachovaná dominancia dát relácie Táraninky. Prehľad verzie datasetu označenej ako “NN” s validačnou množinou bez augmentácie trénovacej množiny zobrazuje Tab. 13.

Tab. 13 Prehľad verzie datasetu NN

Dataset NN	**Trénovacia množina**	**Validačná množina**
Počet rečníkov	242 (105 mužských, 137 ženských)	139 (51 mužských, 88 ženských)
Počet výrokov	1515 (617 mužských, 844 ženských)	380 (170 mužských, 210 ženských)
Priem. dĺžka výroku	6,9 sekúnd	7,2 sekúnd
Trvanie (hh:mm:ss)	02:54:14	00:45:35

Následne bola vytvorená druhá verzia datasetu s validačnou množinou kde bola trénovacia množina rozšírená pomocou perturbácie rýchlosti. Verzia datasetu s augmentovanou trénovacou množinou bola označená ako “AN”. Postup augmentácie bol identický s metódou použitou pri prvotnej augmentácii pomocou perturbácie rýchlosti, ako je podrobne opísane v kapitole 4.4.2. Prehľad verzie datasetu označenej ako “AN” s validačnou množinou spolu s augmentáciou trénovacej množiny zobrazuje Tab. 14.

Tab. 14 Prehľad verzie datasetu AN

Dataset AN	Trénovacia množina	Validačná množina
Počet rečníkov	242 (105 mužských, 137 ženských)	139 (51 mužských, 88 ženských)
Počet výrokov	4545 (2013 mužských, 2532 ženských)	380 (170 mužských, 210 ženských)
Priem. dĺžka výroku	6,95 sekúnd	7,2 sekúnd
Trvanie (hh:mm:ss)	08:46:13	00:45:35

5.7. Trénovanie modelov s využitím validačnej množiny

Táto kapitola popisuje zdokonalený proces trénovania modelu wav2vec2 v dvoch rôznych sériách trénovania, ktoré boli označené ako NN (trénované pomocou datasetu verzie NN) a AN (trénované na verzii datasetu AN). Na základe poznatkov a zistených nedostatkov z počiatočných trénovacích experimentov popísaných v predchádzajúcich kapitolách bolo potrebné zamerať sa na implementáciu zlepšenej trénovacej stratégie. Tá zahŕňala rozdelenie datasetu na vytvorenie validačnej množiny a dôkladné doladenie trénovacích parametrov s cieľom zvýšiť presnosť modelu. Táto fáza predstavuje zásadný pokrok v úsilí o optimalizáciu modelov pre využitie v reálnych scenároch.

Významným prvkom použitej metodológie v tejto fáze je používanie validačnej množiny. Táto množina slúži nielen ako nástroj na pravidelné hodnotenie presnosti počas procesu trénovania, ale zohráva aj kľúčovú úlohu pri výbere konečného modelu. Dôkladnou analýzou presnosti modelov na tejto validačnej množine bola identifikovaná a vybraná verzia modelu, ktorá vykazovala najvyššiu presnosť. Tento prístup zabezpečuje, že konečný výber modelu je podložený spoľahlivými výsledkami, ktoré zdôrazňujú schopnosť modelu efektívne generalizovať na nevidených dátach.

Cieľom tohto trénovacieho a hodnotiaceho prístupu bolo zistiť vplyv rozšírenia datasetu (novými dátami ako aj perturbáciou rýchlosti) a vytvorenia validačnej množiny na presnosť trénovaných modelov. Porovnaním modelu NN, ktorý využíva pôvodný, nedotknutý dataset, s modelom AN, natrénovaným na rozšírenom datasete, bolo v snahe identifikovať model s najlepším výsledkom na validačnej množine.

5.7.1. Úprava trénovacieho postupu

Na základe implementácie validačnej množiny do trénovacieho procesu bolo potrebné mierne upravenie samotného kódu Kaggle notebooku, ktorý je podrobne popísaný v kapitole 5.1.3. Jednou z vykonaných zmien bol presun datasetov z Google disku na priamy hosting platformy Kaggle. Tento krok zjednodušil proces načítania dát, čím sa stal časovo efektívnejším a

menej náchylným na problémy súvisiace s prístupom ku Google disku, ktoré boli počas práce viac krát zaznamenané. Toto vylepšenie pomohlo ušetriť kvótu využitia GPU určenú spoločnosťou Kaggle. Nahrané verzie datasetov s validačnou množinou NN a AN na hostingu Kaggle zobrazuje Obr. 31.

Obr. 31 Datasety nahrané na hosting Kaggle

Na základe zmeny uloženia datasetov bolo taktiež potrebné upraviť samotné načítanie v rámci notebooku Kaggle. Keďže sa pracovalo s dvoma verziami datasetov, boli vytvorené dva odlišné Kaggle notebooky pre trénovanie modelov pomenované “training_80_20_NN.ipynb” a “training_80_20_AN.ipynb”. Načítanie datasetu pre trénovanie modelu pomocou datasetu NN zobrazuje Obr. 32. Rovnako bolo potrebné upraviť predspracovanie dát a tvorbu slovníka nakoľko bola na validáciu využívaná validačná množina namiesto testovacej množiny.

```
# TRAIN 80 without AUG
!mkdir -p /audio_dataset/train/
!cp -r /kaggle/input/dp-80-20-train-valid-nn/train /audio_dataset/
```

```
# VALIDATION 20 without AUG
!mkdir -p /audio_dataset/valid/
!cp -r /kaggle/input/dp-80-20-train-valid-nn/valid /audio_dataset/
```

Obr. 32 Načítanie trénovacej a validačnej časti datasetu NN

Významným metodologickým vylepšením bolo začlenenie validačnej množiny do konfigurácie objektu “Trainer”. Táto zmena znamenala nahradenie testovacej množiny použitej pre argument “eval_dataset” novo vytvorenou validačnou množinou. Použitie validačnej množiny odlišnej od testovacej množiny umožňovalo presnejšie posúdenie presnosti modelu bez biasu a jeho zovšeobecňujúcich schopností. Validačná množina slúžila ako kontrolný bod, ktorý umožňoval identifikovať a zmierniť nadmerné preučenie a poskytovala použiteľné poznatky, ktoré slúžili na ďalšie ladenie trénovacích parametrov modelu.

5.8. Dosiahnuté výsledky

V tejto kapitole sú uvedené výsledky presnosti trénovaných modelov pomocou datasetov verzie NN a AN a rôznych konfigurácií trénovania. Výber najlepších modelov jednotlivých kategórií bez zaujatia spočíval v analýze ich výsledkov na validačnej množine. Ako bolo spomenuté v kapitole 0. Nakoľko bolo zistené značné zlepšenie presnosti modelov pri použití spomínaného 3-gram jazykového modelu, výber najlepších modelov zohľadňoval najlepšie výsledky na validačnej množine s využitím jazykového modelu. Finálna presnosť vybraných modelov bola vyhodnotená pomocou využitia opravenej testovacej množiny, ktorú počas trénovania žiaden z modelov nevidel a blízko predstavuje výsledky na reálnych dátach pri nasadení modelov.

5.8.1. Výsledky modelov série NN

Modely označené "NN" (celým názvom "wav2vec2-XLS-R-SK-CHILD-NN-vX.X") boli trénované na datasete "NN", ktorého tvorba je popísaná v kapitole 5.6. Výsledky najlepších presnosti modelov série NN s využitím rôznych dátových augmentácií zobrazuje Tab. 15.

Tab. 15 Výsledky modelov série NN

		Validačná množina		**Testovacia množina**	
Dátová augmentácia	**Množstvo trénovacích dát (hh:mm)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**
O	2:54	**39,32**	**34,62**	**41,14**	**36,38**
O+SA	2:54	**24,29**	**14,94**	**26,48**	**16,54**

5.8.2. Výsledky modelov série AN

Modely označené "AN" (celým názvom "wav2vec2-XLS-R-SK-CHILD-AN-vX.X") boli trénované na datasete "AN", ktorého tvorba je taktiež popísaná v kapitole 5.6. Výsledky najlepších presnosti modelov série AN s využitím rôznych dátových augmentácií zobrazuje Tab. 16.

Tab. 16 Výsledky modelov série AN

		Validačná množina		**Testovacia množina**	
Dátová augmentácia	**Množstvo trénovacích dát (hh:mm)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**
O+SP	8:46	**27,78**	**23,83**	**30,64**	**26,32**
O+SP+SA	8:46	**24,14**	**15,58**	**25,55**	**16,38**

Všetky použité hodnoty trénovacích parametrov a výsledky modelov boli zaznamenané do priloženého Excel súboru “Training_info.xlsx”.

5.8.3. Porovnanie výsledkov modelov sérií NN a AN

Porovnanie výsledkov modelov série NN a AN z kapitol 5.8.1 a 5.8.2 poskytuje pohľad na ich presnosť na validačnej množine a testovacej množine, ktorá predstavuje modelmi doposiaľ nevidené dáta. Ako bolo spomenuté v kapitole 5.8, výber najlepších modelov pre každú trénovaciu konfiguráciu bol závislý od ich presnosti na validačnej množine, pričom konečné hodnotenie ich presnosti prebiehalo na testovacej množine. Tento postup zabezpečuje, že výber a hodnotenie modelov je objektívne a nezavádza žiadnu zaujatosť do procesu výberu.

V procese zvyšovania presnosti wav2vec2 modelu pri úlohách rozpoznávania detskej reči v slovenčine boli použité dve rôzne stratégie augmentácie dát: perturbácia rýchlosti a metóda SpecAugment. Výsledkom použitia týchto techník bol vývoj dvoch sérií modelov NN a AN. Modely NN boli trénované s použitím originálneho trénovacieho datasetu s kombináciou metódy SpecAugment, zatiaľ čo modely AN využívali trénovací dataset rozšírený prostredníctvom perturbácie rýchlosti v spojení s rovnakými nastaveniami metódy SpecAugment.

Počiatočné trénovania NN modelu na originálnom, neaugmentovanom datasete prinieslo chybovosť slov (WER) na testovacej množine o hodnote 41,14% bez LM a 36,38% s LM. Avšak zavedenie samotnej perturbácie rýchlosti výrazne znížilo hodnotu WER na 30,64% bez LM a 26,32% s LM, čo zdôrazňuje účinnosť tohto rozšírenia pri zlepšovaní presnosti modelu voči zmenám tempa a výšky reči. Napriek tomuto zlepšeniu viedli následne tréningy s použitím oboch techník augmentácie (perturbácia rýchlosti a SpecAugment) pre modely AN k pozoruhodne podobným výsledkom WER ako pri modeloch NN, či už s integráciou jazykového modelu alebo bez neho. Zavedenie metódy SpecAugment pre NN modely znížilo WER na 26,48% bez LM a 16,54% s LM, zatiaľ čo rovnaká metóda aplikovaná na AN modely dosiahla veľmi mierne lepšiu chybovosť slov, konkrétne 25,55% bez LM a 16,38% s LM. Tieto výsledky boli výrazne lepšie ako výsledky na originálnych dátach bez augmentácie, ale nepreukázali významný rozdiel medzi datasetom augmentovaným výlučne metódou SpecAugment a datasetom ďalej rozšíreným pomocou perturbácie rýchlosti.

Toto pozorovanie naznačuje, že zatiaľ čo perturbácia rýchlosti ako samostatná metóda augmentácie výrazne zlepšuje výkon modelu, jej dodatočný prínos sa znižuje v kombinácií s metódou SpecAugment. Metóda SpecAugment, ktorá je známa svojím komplexným prístupom k simulácii akustických zmien – vrátane frekvenčného maskovania a časového maskovania – pravdepodobne v značnej miere zahŕňa účinky perturbácie rýchlosti. V dôsledku toho je

prírastková hodnota integrácie perturbácie rýchlosti spoločne s metódou SpecAugment menej výrazná alebo redundantná. Tab. 17 zobrazuje výsledky najlepších modelov sérií NN a AN.

Tab. 17 Porovnanie výsledkov najlepších modelov sérií NN a AN

	Validačná množina		**Testovacia množina**	
Názov modelu	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**	**WER bez jazykového modelu (%)**	**WER s jazykovým modelom (%)**
wav2vec2-XLS-R-SK-CHILD-NN-v2.5	**24,29**	**14,94**	**26,48**	**16,54**
wav2vec2-XLS-R-SK-CHILD-AN-v2.5	**24,14**	**15,58**	**25,55**	**16,38**

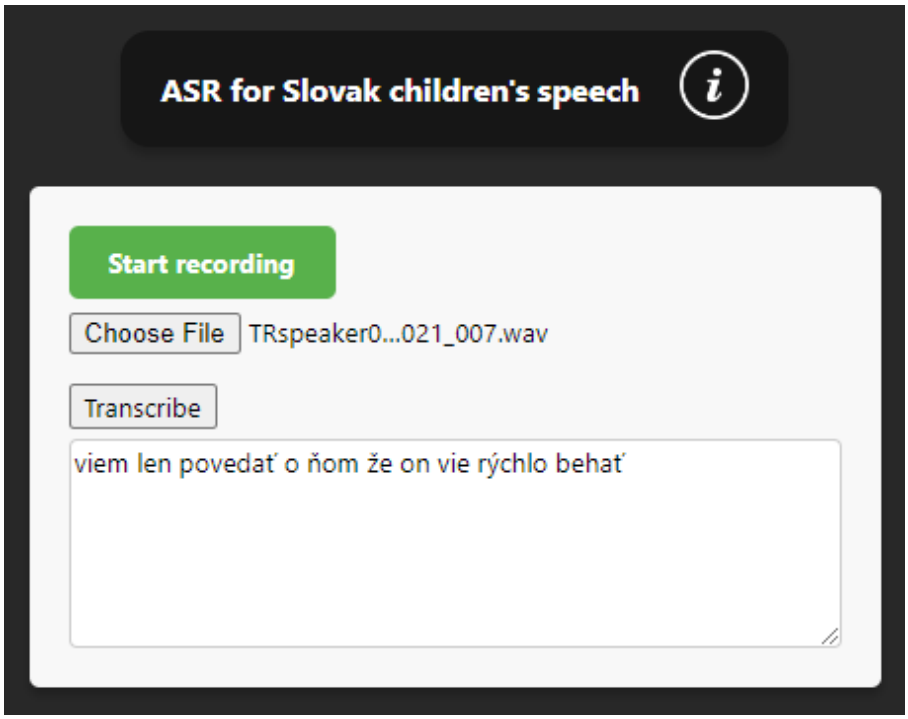
Model “wav2vec2-XLS-R-SK-CHILD-NN-v2.5” vykázal najlepšie výsledky na validačnej množine pri použití jazykového modelu na základe čoho bol vybraný na implementáciu do webovej aplikácie určenej na transkripciu slovenskej detskej reči.

6. Vytvorenie webovej aplikácie na transkripciu slovenskej detskej reči

Táto kapitola opisuje integráciu dotrénovaného wav2vec2 modelu v rámci webovej aplikácie založenej na Flasku, ktorá bola následne kontajnerizovaná pomocou Dockeru na nasadenie v rôznych systémoch. Zameraním bolo vytvorenie používateľského rozhrania na transkripciu zvukových nahrávok pomocou dotrénovanej verzie wav2vec2 modelu prispôsobenej pre slovenskú detskú reč.

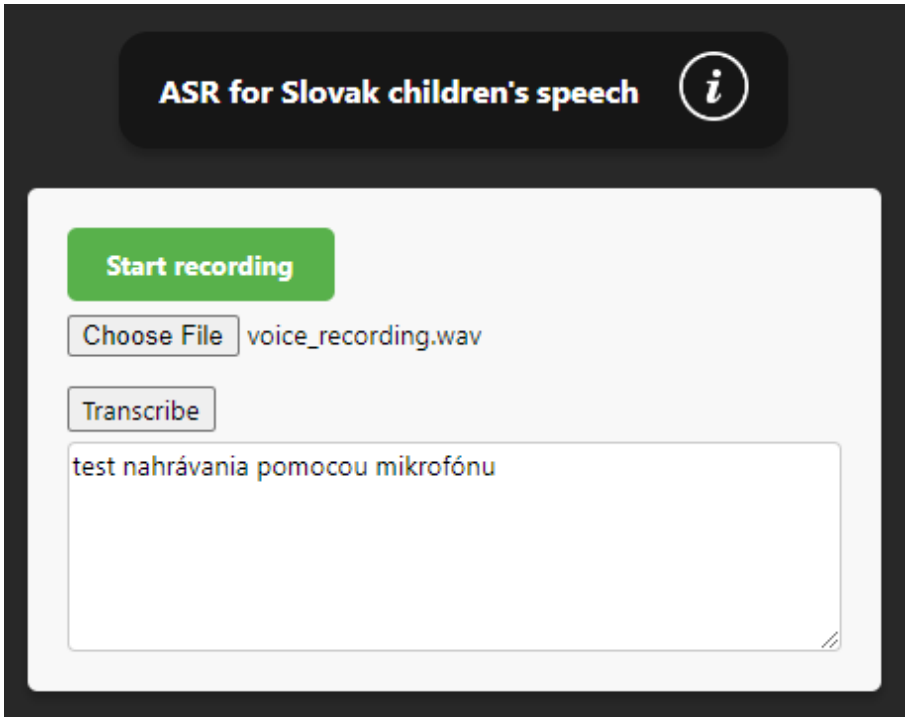
6.1. Dizajn používateľského rozhrania

Frontend webovej aplikácie bol navrhnutý so zameraním na interakciu s používateľom a využíva HTML, CSS a JavaScript na vytvorenie intuitívneho používateľského rozhrania. Používateľské rozhranie umožňuje nahrávanie reči priamo cez mikrofón pomocou tlačidla “Start recording” (Spustiť nahrávanie) alebo nahranie existujúceho “.wav” zvukového súboru prostredníctvom rozhrania s možnosťou následného spustenia procesu transkripcie pomocou tlačidla “Transcribe” (Transkribovať). Dizajn obsahuje aj informačnú ikonu, ktorá po kliknutí zobrazuje používateľskú príručku a ďalšie podrobnosti o systéme. Výstup transkripcie je dynamicky zobrazený v textovom poli na tej istej stránke hneď po spracovaní backendom, čo poskytuje rýchly prístup k transkripcii nahrávky. Na Obr. 33 je znázornená ukážka výstupnej transkripcie v textovom poli webovej stránky. Tento obrázok zachytáva transkripciu zvukovej nahrávky “TRspeaker003_TR16052021_007.wav”, ktorá pochádza z testovacej množiny z datasetu slovenskej detskej reči.



Obr. 33 Ukážka transkripcie nahrávky pomocou webovej aplikácie

Podobne Obr. 34 znázorňuje príkladnú ukážku transkripcie nahrávky nahratej pomocou mikrofónu, kde bola vyslovená veta „Test nahrávania pomocou mikrofónu“.



Obr. 34 Ukážka transkripcie nahrávky vytvorenej pomocou mikrofónu

6.2. Backendové spracovanie

Backend postavený na frameworku Flask a jazyku Python spracúva prichádzajúce HTTP požiadavky a obsluhuje webový obsah. Ako jadro backendu pre spracovanie zvukových nahrávok a vygenerovanie transkripcie bol vybraný a implementovaný dotrénovaný model “wav2vec2-XLS-R-SK-CHILD-NN-v2.5” spoločne s jazykovým modelom, ktorý v rámci vyhodnotenia vykazoval najlepšie výsledky na validačnej množine s využitím jazykového modelu. Po prijatí zvukového súboru vo formáte “.wav” backend vykoná predspracovanie a zabezpečí, aby nahrávka spĺňala požiadavky modelu konverziou vzorkovacej frekvencie a úpravou nahrávky na mono kanál. Následne je spracovaná nahrávka odovzdaná modelu na vygenerovanie transkripcie. Výsledná transkripcia je potom odovzdaná frontendu, aby bola prezentovaná používateľovi.

6.3. Dockerizácia a nasadenie

Kľúčovú rolu v stratégií nasadenia aplikácie zohrávala jej dokerizácia, ktorá zjednodušila a zjednotila celý proces nasadenia. Vytvorením súboru Dockerfile boli špecifikované inštrukcie pre vytvorenie Docker obrazu (angl. Docker Image), ktorý obsahuje aplikáciu a všetky potrebné knižnice a súčasti. Táto izolácia prostredia chráni pred rozdielmi na úrovni systému, čo umožňuje aplikácii fungovať jednotne, či je nasadená na lokálnych strojoch alebo v cloudových službách. Využitie Dockeru zaručuje, že správanie sa aplikácie a dostupnosť zostávajú konzistentné vo všetkých nasadeniach. V rámci práce bola aplikácia nasadená a testovaná na lokálnom virtuálnom stroji s operačným systémom Ubuntu a Google Cloud hostingu, kde v oboch prípadoch aplikácia vykazovala identické správanie.

Záver

Táto diplomová práca bola zameraná na vytvorenie systému automatického rozpoznávania detskej reči v slovenčine a jeho následnú implementáciu v rámci webovej aplikácie. V teoretickej časti boli opísané základy systémov automatického rozpoznávania reči spoločne s výzvami rozpoznávania detskej reči. Taktiež boli priblížené augmentačné metódy, ktoré dosahovali uspokojujúce výsledky pri riešení podobnej problematiky. Práca taktiež priblížila funkcionalitu moderného end-to-end modelu wav2vec 2.0 a jeho predtrénovanú verziu XLS-R. Posledná časť teoretickej časti stručne popisuje vybranú hodnotiacu metriku WER a existujúci dataset detskej reči v slovenčine.

V rámci praktickej časti práce bol vykonaný zber dát detskej reči v slovenčine spracovaním 34 rôznych epizód rádiovej relácie Rozhlasové leporelo. Získané nahrávky obsahujúce segmenty detskej reči v slovenčine boli v rámci práce spracované a manuálne transkribované. Zo získaných dát bol vytvorený dataset rádiovej relácie Rozhlasové leporelo o trvaní 1 hodina a 6 minút čistej reči detí. Následne boli dáta datasetu relácie Rozhlasové leporelo spojené s existujúcim datasetom relácie Táraninky pre vytvorenie rozšíreného trénovacieho datasetu s trvaním 3 hodiny 39 minút a 49 sekúnd.

Na základe prvotných experimentov dotrénovania modelov XLS-R pomocou rozšíreného datasetu detskej reči v slovenčine s využitím metodiky predchádzajúcej práce bola zistená oblasť pre zlepšenie, vďaka ktorej bola do procesu dotrénovania zavedená validačná množina. Dotrénovaný model z prvotných experimentov bol následne v rámci práce použitý pre kontrolu a opravu chybných manuálnych transkripcií testovacej množiny.

Pomocou upravenej metodiky boli následne dotrénované nové verzie modelov XLS-R, ktoré boli ladené využitím validačnej množiny a vyhodnotené testovacou množinou. Najlepší dotrénovaný model v rámci vykonaných experimentov dosiahol hodnotu WER na testovacej množine 26,48% bez využitia jazykového modelu a 16,54% s využitím jazykového modelu.

Tento model bol následne implementovaný v rámci webovej aplikácie s intuitívnym používateľským rozhraním, ktorá umožňuje jednoduchú transkripciu detskej slovenskej reči pomocou vstupu z mikrofónu alebo nahraním zvukovej nahrávky.

Pre budúcu prácu je nevyhnutné zvážiť ďalšie rozšírenie datasetu detskej reči v slovenčine pre trénovanie a odsledovať jeho vplyv na presnosť modelu spoločne s ďalšími experimentami aplikácie augmentačných metód. Na základe experimentov tejto práce existuje predpoklad, že ďalšie rozšírenie trénovacích dát môže mať veľmi pozitívne účinky na presnosť dotrénovaného

XLS-R modelu. Ďalším možným zlepšením je využitie dotrénovaného modelu na korekciu manuálnej transkripcie trénovacej a validačnej množiny pre opätovné dotrénovanie modelu využitím opravených transkripcií s cieľom zlepšenia presnosti modelu bez potreby získavania nových dát.

Zoznam použitej literatúry

[1] D. Al-Fraihat, Y. Sharrab, F. Alzyoud, A. Qahmash a A. A. Maaita, „Speech Recognition Utilizing Deep Learning: A Systematic Review of the Latest Developments," *Human-centric Computing and Information Sciences,* zv. 14, 2024.

[2] S. Shraddha, G. Jyothish Lal a S. Sachin Kumar, „Child Speech Recognition on End-to-End Neural ASR Models," rev. *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Karnataka, 2022.

[3] A. Baevski, H. Zhou, A. Mohamed a M. Auli, „wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," 2020. [Online]. Available: https://arxiv.org/abs/2006.11477.

[4] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau a M. Auli, „XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," [Online]. Available: https://arxiv.org/abs/2111.09296.

[5] S. Shahnawazuddin, N. Adiga, H. K. Kathania a B. T. Sai, „Creating speaker independent ASR system through prosody modification based data augmentation," *Pattern Recognit. Lett.,* zv. 131, pp. 213-218, 2020.

[6] V. Kadyan, H. Kathania, P. Govil a M. Kurimo, „Synthesis speech based data augmentation for low resource children ASR," *Speech and Computer (Lecture Notes in Computer Science),* zv. 12997, pp. 317-326, 2021.

[7] W. Wang, Z. Zhou, Y. Lu, H. Wang, C. Du a Y. Qian, „Towards Data Selection on TTS Data for Children's Speech Recognition," rev. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, 2021.

[8] R. Serizel a D. Giuliani, „Vocal tract length normalisation approaches to DNN-based children's and adults speech recognition," rev. *2014 IEEE Spoken Language Technology Workshop (SLT 2014)*, South Lake Tahoe, 2014.

[9] G. Yeung, R. Fan a A. Alwan, „Fundamental Frequency Feature Normalization and Data Augmentation for Child Speech Recognition," rev. *ICASSP 2021 - 2021 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, 2021.

[10] J. Fainberg, P. Bell, M. Lincoln a S. Renals, „Improving Children's Speech Recognition Through Out-of-Domain Data Augmentation," rev. *Proc. Interspeech 2016*, 2016, pp. 1598-1602.

[11] G. Chen, X. Na, Y. Wang, Z. Yan, J. Zhang, S. Ma a Y. Wang, „Data Augmentation For Children's Speech Recognition -- The "Ethiopian" System For The SLT 2021 Children Speech Recognition Challenge," 2020. [Online]. Available: https://arxiv.org/abs/2011.04547.

[12] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk a Q. V. Le, „SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," 2019. [Online]. Available: https://arxiv.org/abs/1904.08779.

[13] V. P. Singh, H. Sailor, S. Bhattacharya a A. Pandey, „Spectral Modification Based Data Augmentation For Improving End-to-End ASR For Children's Speech," 2022. [Online]. Available: https://arxiv.org/abs/2203.06600.

[14] R. Jain, A. Barcovschi, M. Y. Yiwere, D. Bigioi, P. Corcoran a H. Cucu, „A Wav2Vec2-Based Experimental Study On Self-Supervised Learning Methods To Improve Child Speech Recognition," 2023.

[15] M. Gevirtz, „What is Word Error Rate (WER)?," Deepgram, 20 November 2023. [Online]. Available: https://deepgram.com/learn/what-is-word-error-rate. [Cit. 6 April 2024].

[16] P. Michalanský, „Trénovanie modelov detskej reči pre systémy automatického rozpoznávania reči v slovenčine (Diplomová práca)," Technická univerzita v Košiciach, 2023.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser a I. Polosukhin, „Attention Is All You Need," [Online]. Available: https://arxiv.org/abs/1706.03762.

[18] J. Devlin, M.-W. Chang, K. Lee a K. Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805.

[19] A. van den Oord, Y. Li a O. Vinyals, „Representation Learning with Contrastive Predictive Coding," 2018. [Online]. Available: https://arxiv.org/abs/1807.03748.

[20] X. Cai, J. Yuan, Y. Bian, G. Xun, J. Huang a K. Church, „W-CTC: A CONNECTIONIST TEMPORAL

CLASSIFICATION LOSS WITH WILD CARDS,“ [Online]. Available: https://openreview.net/pdf?id=0RqDp8FCW5Z.

[21] C. Wang, M. Rivière, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino a E. Dupoux, „VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation,“ [Online]. Available: https://arxiv.org/abs/2101.00390.

[22] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve a R. Collobert, „MLS: A Large-Scale Multilingual Dataset for Speech Research,“ [Online]. Available: https://arxiv.org/abs/2012.03411.

[23] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers a G. Weber, „Common Voice: A Massively-Multilingual Speech Corpus,“ [Online]. Available: https://arxiv.org/abs/1912.06670.

[24] J. Valk a T. Alumäe, „VoxLingua107: a Dataset for Spoken Language Recognition,“ [Online]. Available: https://arxiv.org/abs/2011.12998.

[25] M. J. F. Gales, K. M. Knill, A. Ragni a S. P. Rath, „Speech recognition and keyword spotting for low-resource languages : Babel project research at CUED,“ [Online].

[26] M. Gerboc, „Detská reč, jej charakteristiky a spracovanie (Bakalárska práca),“ Technická univerzita v Košiciach, 2020.

[27] M. Gerboc, „Akustické modelovanie detskej reči (Diplomová práca),“ Technická univerzita v Košiciach, 2022.

[28] P. Michalanský, „Tvorba rečových databáz (Bakalárska práca),“ Technická univerzita v Košiciach, 2021.

[29] RTVS, „Rozhlasové leporelo - Relácie rádií,“ RTVS, [Online]. Available: https://www.rtvs.sk/radio/program/1567.

[30] M. Lojka, P. Viszlay, J. Staš, D. Hládek a J. Juhár, „Slovak Broadcast News Speech Recognition and Transcription System,“ *Advances in Network-Based Information Systems,* pp. 385-394, 2018.

Prílohy

Príloha A: CD médium – Zdrojové kódy, Informácie o datasetoch a trénovaní, Konfiguračný súbor pre program Transcriber