VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÝ PŘEPIS ŘEČI LETECKÉ KOMUNIKACE DO TEXTU

AUTOMATIC TRANSCRIPTION OF AIR-TRAFFIC COMMUNICATION TO TEXT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **VERONIKA NEVAŘILOVÁ**
AUTHOR

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

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

150718

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Studentka:	**Nevařilová Veronika**
Program:	Informační technologie
Název:	**Automatický přepis řeči letecké komunikace do textu**
Kategorie:	Zpracování signálů
Akademický rok:	2023/24

Zadání:

1. Seznamte se s automatickým rozpoznáváním řeči a s rozpoznávačem Whisper.
2. Obstarejte si data, navrhněte anotační protokol a oanotujte dostatečné množství dat.
3. Připravte anotovaná data do vhodného formátu a adaptujte rozpoznávač Whisper. Iterativně vyhodnocujte úspěšnost zatímco budete dále anotovat data. Zaměřte se na různé formy textového výstupu (např. plný a zkrácený).
4. Zhodnoťte výsledky a navrhněte směry dalšího vývoje. Při hodnocení se zaměřte také na uživatelskou zkušenost z pohledu řídícího letového provozu.
5. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Alec Radford et al. "Robust Speech Recognition via Large-Scale Weak Supervision", https://cdn.openai.com/papers/whisper.pdf, 2022
- ATCO2 projekt, http://atco2.org
- Dále dle pokynů vedoucího

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

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

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

Abstrakt

Tato práce se zabývá adaptací Whisperu, modelu automatického rozpoznávání řeči vyvinutého společností OpenAI, na českých a anglických záznamech letecké komunikace. Poskytuje základní vhled do problematiky rozpoznávání řeči, neuronových sítí a modelů stavěných na transformer architektuře. Popsány jsou také sběr a anotace dat a nakonec průběh a porovnání učení na dvou různých formách přepisu – plném, kdy se model učí přepisovat nahrávky slovo od slova, a zkráceném, který je snadnější pro rychlé vyhledávání informací v textu a přirozenější pro řídící letového provozu.

Abstract

This thesis focuses on fine-tuning Whisper, an automatic speech recognition model developed by OpenAI, on Czech and English recordings of air-traffic communication. It provides a fundamental insight into automatic speech recognition, neural networks and transformer architecture. Further, data collection and annotation is also described and after that it details the process and outcomes of Whisper's training on two different transcription formats – full, where the model learns to transcribe recordings word by word, and abbreviated, which is more suitable for quick navigation and more natural for air traffic controllers.

Klíčová slova

Letecká komunikace, neuronové sítě, umělá inteligence, zpracování přirozeného jazyka, rozpoznávání řeči, OpenAI, Whisper.

Keywords

Air-traffic communication, neural networks, AI, natural language processing, automatic speech recognition, OpenAI, Whisper.

Citace

NEVAŘILOVÁ, Veronika. *Automatický přepis řeči letecké komunikace do textu.* Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szőke, Ph.D.

Automatický přepis řeči letecké komunikace do textu

Prohlášení

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

.....
Veronika Nevařilová
8. května 2024

Poděkování

Ráda bych poděkovala vedoucímu této bakalářské práce, panu Ing. Igoru Szőkemu, Ph.D., za velmi cenné odborné rady a návrhy, prostředí pro anotaci dat i zajištění přístupu na výpočetní cluster pro trénování modelů. Poděkování dále náleží letišti v Kunovicích za poskytnutí nahrávek letecké komunikace. V neposlední řade děkuji také mým rodičům za jejich neustálou podporu během celého mého studia.

Obsah

Kapitola 1

Úvod

Letecká komunikace pomocí řeči představuje hlavní prostředek řízení letového provozu. Řídící letového provozu je zodpovědný především za jeho plynulost a bezpečnost. Během toho se mezi řídícími a piloty přenáší velké množství velmi důležitých informací, které jsou klíčové pro zachování těchto aspektů, a každá chyba způsobená například jen přeslechnutím jednoho slova může mít fatální následky.

I když se chybám snaží letecká komunikace vyhýbat ať už ustálenými frázemi, speciální rozdílnou výslovností některých podobných slov nebo hláskováním, některé oblasti jsou přesto kritické. Například to, že řídící většinou nemá za úkol jen sledovat a řídit letadla v jemu svěřeném vzdušném prostoru, ale zároveň musí komunikovat s ostatními stanovišti poskytujícími řízení letového provozu nebo informační službu. Dále musí mít příposlech na nouzové frekvenci a pokud se jedná o stanoviště na letišti, často obsluhuje i pozemní provoz aut a jiných složek nezbytných pro správný chod letiště. Což znamená další radiové frekvence a telefonní spojení, na kterých může kdykoli jakákoli třetí strana zahájit vysílání nebo hovor přesně v moment, kdy pilot vysílá na hlavní frekvenci a řídící mu potřebuje co nejlépe a jasně rozumět. Pokud do vysílání pilota začne vysílat na jiné frekvenci někdo další, je pro člověka velmi těžké jedno z vysílání odfiltrovat a může to dopadnout i tak, že řídící nerozumí ani jedné straně a musí každou z nich žádat o opakování. Stejně tak není ojedinělé, že na hlavní frekvenci začne vysílat v jeden moment více pilotů zároveň. Taková situace je o něco složitější a nemá jednoduché řešení, ale u problému současných vysílání na různých frekvencích by mohlo pomoci mít k dispozici textový přepis alespoň frekvence hlavní, kde by si řídící mohl přečíst vysílání, která nezachytil nebo jim dostatečně nerozuměl.

Dalším možným využitím automatického přepisu letecké komunikace by mohlo být usnadnění vyhledávání konkrétních záznamů. Každé letiště má povinnost uchovávat záznamy letecké komunikace po určitou minimální dobu. Ty jsou potřeba například v případě, kdy by se stala nějaká mimořádná či nouzová událost, aby mohla být komunikace přezkoumána. Pokud se ale nahrávky uchovávají pouze v audio formě, hledání některé konkrétní může být provedeno pouze poslechem jednotlivých nahrávek. Ty mohou být sice označeny časem vzniku, avšak letištní frekvence může být při velkém provozu poměrně vytížená a během minuty se může uskutečnit mnoho vysílání. Hledat pak nahrávku mezi desítkami dalších je velmi zdlouhavé. Pokud by se dalo vyhledávat v textovém přepisu, velmi by to hledání usnadnilo.

Právě tyto situace, které pozoruji z pohledu dispečera poskytujícího letištní letovou informační službu, mě inspirovaly pro zvolení tohoto tématu bakalářské práce. Cílem je zkoumat a analyzovat možnosti automatického přepisu řeči letecké komunikace do textu,

dosáhnout pokud možno co nejmenší chybovosti a zhodnotit, do jaké míry by byl výsledek použitelný pro účely řízení letového provozu, popřípadě vyhledávání v historii záznamů.

Práce je strukturována do několika kapitol. Kap. 2 stručně popisuje historii vývoje automatických rozpoznávačů řeči, aktuální přístupy a také základy letecké komunikace. Kapitola 3 se zaměřuje na popis neuronových sítí a princip jejich učení. V kap. 4 se poznatky z předchozích kapitol spojí při popisu architektury typu transformer a modelu automatického rozpoznávání řeči Whisper. Kap. 5 nabízí pohled do problematiky získávání a anotace dat v různých formách pro účely trénování modelů strojového učení s učitelem. Dále popisuje tvorbu datasetů, konkrétní testovací množiny, na kterých budou výsledky učení evaluovány, i trénovací skripty a prostředí. Všechny tyto výstupy budou poté využity v kap. 6, ve které je popsáno učení modelu Whisper na záznamech letecké komunikace a různé experimenty s cílem dosáhnout co nejmenší chybovosti modelu. Na závěr jsou v kap. 7 zhodnoceny dosažené výsledky a navrhnuty směry dalšího vývoje.

Kapitola 2

Automatické rozpoznávání řeči a letecká komunikace

Příchod chytrých zařízení a strojového učení postavil do popředí automatické rozpoznávání řeči. Pro člověka je dnes mluvené slovo jeden z běžných způsobů, pomocí kterých lze se zařízeními interagovat. Mezi jeho hlavní výhody totiž patří rychlost sdělení svého požadavku v porovnání například s psaním na klávesnici.

Automatické rozpoznávání řeči má ale mnohem širší oblast působnosti, než pouhá osobní elektronická zařízení, a pokusy o tvorbu systémů, které by rozuměly lidskému slovu, nejsou ani zdaleka výsadou 21. století.

2.1 Historie rozpoznávání řeči

Již v roce 1952 Bellovy laboratoře představily elektrický obvod schopný rozpoznávat řeč, který dokázal rozeznat jednotlivé číslice pro vytáčení telefonních čísel [5]. V dalších letech bylo ve světě vyvinuto spoustu dalších rozpoznávačů, avšak v této době rozpoznávače dokázaly rozeznávat pouze jednotlivá slova či hlásky.

V 80. letech začaly být studovány pro potřeby rozpoznávání řeči skryté Markovovy modely (HMM). Nejznámější takový rozpoznávač, který využíval HMM ve spojení se směsí Gaussovských rozložení (GMM), SPHINX, vyvinuli Kai-Fu Lee a Hsiao-Wuen Hon, kteří se zaměřili na rozpoznávání řeči s rozsáhlým slovníkem, nezávislém na řečníkovi [13]. Ten prokázal vysokou přesnost a HMM se začaly používat v takovém rozsahu, že byly aktuální ještě na počátku 21. století. V té době už ale výsledky tohoto přístupu začaly dosahovat svých limitů. V roce 2011 skupina výzkumníků Microsoftu představila systém HMM ve spojení s hlubokou neuronovou sítí, pracující s kontextem (angl. *context-based deep belief network hidden Markov model*, zkr. CB-DBN-HMM) [4]. Tento přístup se ukázal jako výrazně lepší než klasické GMM-HMM. Od té doby se pozornost čím dál více obracela právě k hlubokým neuronovým sítím [28].

2.2 Dnešní přístupy

Modely neuronových sítí jsou dnes pro rozpoznávání řeči a její automatický přepis nejčastější volba. Využívá se zejména tzv. *end-to-end* modelů, které přímo mapují vstupní audio či jeho spektrum na posloupnost slov. End-to-end modely obsahují místo několika různých oddělených částí, které se nachází v HMM, právě neuronovou síť, která provádí zpracování

vstupu a na výstupu se nachází již konečná sekvence slov [28]. Dnešní nejpoužívanější end-to-end architektura pro zpracování řeči je tzv. transformer architektura, která bude podrobněji vysvětlena v kapitole 4.

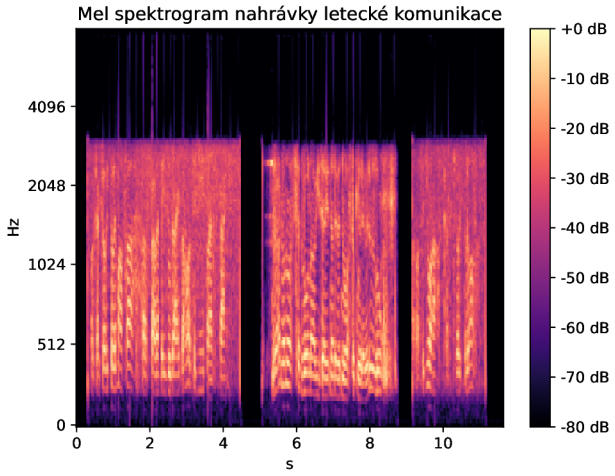
2.3 Zpracování zvuku pro potřeby rozpoznávání řeči

Zvuk je mechanické vlnění šířící se pružným prostředím. Jedná se o analogový signál, který lze snímat mikrofonem, zařízením citlivým na změny tlaku prostředí. Zvuk je mikrofonem snímán analogově a převáděn pomocí A/D převodníku na sekvenci číselných hodnot, které jsou pak ukládány v počítači.

Nahraný zvuk lze dále analyzovat – mezi nejběžnější analýzy zvuku patří Fourierova transformace [7], což je rozklad signálu do jeho frekvenčních složek. Z výsledku transformace je tedy například možné zjistit, jaké frekvence jsou ve zvukovém signálu zastoupeny nejvíce, nebo zda-li se v signálu vyskytuje šum. Tento výstup se nazývá *frekvenční spektrum signálu.*

Vzhledem k tomu, že většina signálů je neperiodická a jejich spektrum se tedy v čase mění, můžeme tyto změny pozorovat aplikací Fourierovy transformace na menší okna signálu, která se částečně překrývají. Výsledkem je *spektrogram*, který umožňuje sledovat vývoj zastoupení různých frekvencí signálu v čase.

Člověk vnímá jen malou část frekvenčního spektra zvuku. Rozmezí, které lidské ucho slyší, se uvádí zhruba od 16-20 Hz do 16-20 kHz [16]. Člověk ovšem nevnímá rozdíly frekvencí lineárně – hůře rozlišuje rozdíl mezi dvěma vysokými frekvencemi, zatímco u nižších frekvencí s tím problém nemá. K řešení tohoto jevu při vizualizaci spektrogramu se často používá tzv. Mel škála, jejíž měřítko je založeno na ekvivalentním vnímání rozdílu frekvencí lidským uchem. Spektrogram je tak transformován na Mel spektrogram, který má logaritmickou svislou osu (obr. 2.1).



Obrázek 2.1: Příklad Mel spektrogramu pro nahrávku letecké komunikace.

Právě Mel spektrogram je vstupním bodem většiny dnešních modelů pro rozpoznávání řeči. Mezi hlavní důvody, proč se jej využívá, patří zachování informací při zmenšení velikosti vstupu (signál v časové doméně je paměťově náročnější), a tím i rychlejší zpracování.

2.4 Reprezentace textu – tokenizace

V oblasti zpracování řeči a přirozeného jazyka se často používá tzv. *tokenizace*. Jde o rozdělení dat do menších částí, tzv. *tokenů*, a následné namapování těchto částí s využitím slovníků na typicky celočíselné hodnoty, které je reprezentují. Díky tomu dokážou modely zpracování řeči pracovat s textem. Při automatickém přepisu řeči do textu je tokenizace využito především v opačném smyslu, v části výsledného generování textu, kdy jsou vygenerované tokeny převáděny na řetězce znaků tvořících výsledný text.

Mezi nejpoužívanější tokenizační algoritmy patří Byte-Pair Encoding (BPE) [21]. Tvorba slovníku probíhá na základě iterativního spojování dvou nejčastějších posloupností znaků do jednoho. Díky tomu je možné reprezentovat velké slovníky pomocí menšího počtu částí slov.

BPE je nejčastějším používaným algoritmem pro modely architektury typu transformer (kap. 4). Mezi další algoritmy dále patří např. WordPiece [24] a SentencePiece [12].

2.5 Určování přesnosti rozpoznávačů

Pro určování kvality rozpoznávačů řeči a jejich porovnávání mezi sebou je potřeba mít stanovenou metriku, pomocí níž se budou vyhodnocovat. Takových metrik je více, nejběžnější je však chybovost slov, tzv. *word error rate* (WER). Pracuje na jednoduchém principu porovnání přepisu vygenerovaného rozpoznávačem s referenčním přepisem. Výpočet je definován jako

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

kde:

- S je počet nahrazených slov (*substitutions*),
- D je počet vymazaných slov (*deletions*),
- I je počet vložených slov navíc (*insertions*),
- N je počet slov v referenčním přepisu [8].

Čím je model přesnější, tím menší je WER jeho přepisů. Avšak WER je občas považována za ne úplně ideální metriku. Hlavním důvodem je, že text, který by člověk považoval za víceméně správný, může mít vysokou hodnotu WER. Metrika WER totiž nebere v úvahu jakoukoli podobnost referenčního a vygenerovaného slova. To staví téměř správně vygenerované slovo i úplně jiné slovo, než bylo vyřčeno, na stejnou úroveň. Podobně také mohou dělat problémy například slova zkrácená apostrofem v angličtině:

We didn't see him.

We did not see him.

$$WER = \frac{1 + 0 + 1}{4} = 0{,}5 = 50\%$$

I přesto, že jsou identifikována všechna slova a rozdíl je pouze ve spojení slov *did not*, na takto krátké větě má WER hodnotu 50 %, stejně jako následující příklad, kde má vygenerovaná věta naprosto jiný význam:

We didn't see him.

We didn't send it.

$$WER = \frac{2+0+0}{4} = 0{,}5 = 50\%$$

Proto je třeba brát hodnoty WER částečně s rezervou. I přesto je ale WER hlavní metrikou používanou v oblasti automatického rozpoznávání řeči. Mezi jiné metriky podobné WER patří např. *character error rate* (CER). Ta neporovnává celá slova, ale jednotlivé vygenerované znaky.

2.6 Letecká komunikace

Letecká komunikace je odvětví, ve kterém se vyskytuje značný šum způsobený přenosem přes VHF kanál. Paradoxně se ale pomocí něj přenáší důležité a někdy až kritické informace, jako jsou různé číselné informace, příkazy, volací znaky aj., u nichž je třeba, aby jim příjemce v plné míře rozuměl. Z důvodu špatné kvality přenosu a snahy o naprosto jasný význam vysílání a minimalizaci chyb proto existuje například hláskovací abeceda (tab. 2.1) či různé ustálené fráze, kterých by se piloti a řídící letového provozu měli co nejvíce držet. Špatná interpretace vysílání může mít totiž fatální následky.[1]

Písmeno	Slovo	Písmeno	Slovo
A	Alpha	N	November
B	Bravo	O	Oscar
C	Charlie	P	Papa
D	Delta	Q	Quebec
E	Echo	R	Romeo
F	Foxtrot	S	Sierra
G	Golf	T	Tango
H	Hotel	U	Uniform
I	India	V	Victor
J	Juliett	W	Whiskey
K	Kilo	X	Xray
L	Lima	Y	Yankee
M	Mike	Z	Zulu

Tabulka 2.1: Mezinárodní hláskovací abeceda.

[1] Nejasná komunikace mezi letadly a řídícím v roce 1977 na španělském ostrově Tenerife přispěla k nejtragičtější letecké nehodě v dějinách lidstva, při které zemřelo 583 lidí.

Hláskovací abeceda se používá např. u volacích znaků, výstupních a vstupních bodů vzdušného prostoru či označení letišť. Ustálené fráze jsou používány při běžných pokynech či žádostech – vysílání by měla mít určitou jednotnou formu. Taková komunikace mezi pilotem (L) a řídícím letového provozu (T) pak může vypadat následovně:

```
L: Kunovice Věž Oscar Kilo Alpha Bravo Charlie dobrý den

T: Oscar Kilo Alpha Bravo Charlie Kunovice Věž dobrý den

L: Zlín jedna dva šest z Medlánek do Kunovic, poloha Koryčany, tři tisíce
dva sta stop a žádáme vstup do vašeho CTR pro přistání

T: Oscar Kilo Alpha Bravo Charlie vstup do CTR povolen, dráha v užívání dva
nula střední, vítr dva jedna nula stupňů šest uzlů, oblačnost few tři
tisíce devět set stop, QNH jedna nula jedna tři, pokračujte pravý baseleg
dráhy dva nula střední

L: vstup do CTR povolen, dráha v užívání dva nula střední, QNH jedna nula
jedna tři, pokračujeme pravý baseleg dráhy dva nula střední Oscar Kilo
Alpha Bravo Charlie
```

Během těchto pěti vysílání bylo předáno mezi pilotem a řídícím letového provozu několik hodnot. Vzhledem k tomu, že je ale dialog výše přepsán slovo od slova, nejsou v textu tyto hodnoty na první pohled vůbec viditelné. Níže je proto přepis ve formě, se kterou pracují např. řídící letového provozu při komunikaci s letadly, když jim sdělují informace (např. čtou informace o oblačnosti či atmosférickém tlaku) nebo když si informace o daných letadlech zapisují – číselné hodnoty složené z číslic místo slov, hláskované volací znaky letadel zkráceny:

```
L: Kunovice Věž OKABC dobrý den

T: OKABC Kunovice Věž dobrý den

L: Zlín 126 z Medlánek do Kunovic, poloha Koryčany, 3200 stop a žádáme
vstup do vašeho CTR pro přistání

T: OKABC vstup do CTR povolen, dráha v užívání 20C, vítr 210 stupňů 6 uzlů,
oblačnost few 3900 stop, QNH 1013, pokračujte pravý baseleg dráhy 20C

L: vstup do CTR povolen, dráha v užívání 20C, QNH 1013, pokračujeme pravý
baseleg dráhy 20C OKABC
```

Nejdůležitější hodnoty předané řídícím letového provozu musí vždy pilot v odpovědi zopakovat (tzv. *readback*), což je další způsob, jak minimalizovat šanci přeslechů. Pokud pilot hodnotu zopakuje špatně, řídící pilota opraví a ten musí hodnotu znovu zopakovat.

Špatné porozumění vysílání není vůbec ojedinělé. Běžně některým vysíláním ve špatné kvalitě nerozumí ani piloti a řídící, kteří poslouchají leteckou komunikaci denně. Použití obecných rozpoznávačů pro rozpoznávání řeči letecké komunikace je proto prakticky nemožné. Nejen, že obecné rozpoznávače nejsou zvyklé na tak velkou míru šumu, při které občas i člověk selhává, ale také neznají specifické zkratky a slova, které se při komunikaci

běžně používají. Lepších výsledků je proto možné dosáhnout dedikovanými rozpoznávači natrénovanými na záznamech letecké komunikace (například projekt ATCO2[2]). Takových rozpoznávačů je ovšem málo a primárně se zaměřují na komunikaci v anglickém jazyce, který je v civilním letectví celosvětově hlavním používaným jazykem. Na menších regionálních letištích a obecně ve sportovním létání se používá především mateřský jazyk, tedy u nás čeština. Na tuto část letectví se ale rozpoznávače už tolik nesoustředí.

Důvodem, proč je celkově rozpoznávačů letecké komunikace velmi málo, mohou být data. Trénování rozpoznávačů řeči letecké komunikace vyžaduje záznamy letecké komunikace. Ty jsou ale většinou velmi těžce sehnatelné a rozumný počet záznamů, ideálně s přidanými přepisy komunikace, je veřejně dostupný pouze zřídka v anglickém jazyce, natož v českém.

[2] https://www.atco2.org/

Kapitola 3

Umělá inteligence a neuronové sítě

Umělá inteligence (AI) je oblast informatiky, která se v posledních letech dostala do popředí v mnoha oborech – od medicíny po průmyslovou výrobu. Jejím cílem je vytváření inteligentních strojů, které dokáží simulovat lidské chování a vykonávat úlohy, které jsou považovány za typicky lidské činnosti.

Jednou z hlavních aplikací umělé inteligence je strojové učení (ML). Jedná se o oblast, která usiluje o vytváření systémů, které dokáží samostatně jednat, rozhodovat se a zlepšovat se na základě svých zkušeností. V dnešní době je strojové učení široce využíváno v oborech jako je počítačové vidění, detekce anomálií nebo zpracování řeči a přirozeného jazyka. Pro tyto poměrně různorodé obory a konkrétní úlohy je využíváno rozsáhlého spektra různých algoritmů [10].

3.1 Princip neuronových sítí

Nejčastějším modelem strojového učení v dnešní době jsou umělé neuronové sítě (angl. *artificial neural networks*, zkr. ANN nebo NN). Ty jsou inspirovány funkcí lidského mozku. Jejich základní stavební prvky jsou nazývány, stejně jako jejich biologický vzor, *neurony*. Ty disponují tzv. aktivační (přenosovou) funkcí. Synapse jsou v umělých neuronech reprezentovány jako spojení s váhami. Každý neuron má kromě takových propojení s jinými neurony ještě jeden vstup, tzv. *bias*, který má konstantní hodnotu 1 a mění se pouze jeho váha. Výstup neuronu je pak výstup aktivační funkce neuronu, jejíž vstup je vážený součet hodnot vstupních spojení. Učení neuronů je prováděno pomocí nastavování vah jednotlivých propojení.

Schéma obecného neuronu je zobrazeno na obrázku 3.1. Každý vstup neuronu x_i disponuje svou váhou w_i a výstup neuronu O je definován níže:

$$O = f(net) = f\left(\sum_{j=0}^{N} w_j x_j\right) \tag{3.1}$$

Funkce f je aktivační funkce neuronu. Proměnná net je definována jako skalární součet vektorů $\vec{w}$ a $\vec{x}$:

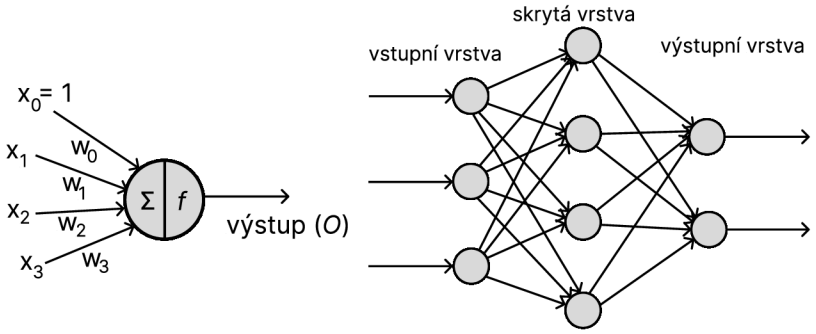
$$net = w^{\mathrm{T}} x = w_1 x_1 + \cdots + w_n x_n, \tag{3.2}$$

kde T je sloupcový vektor a nejjednodušší aktivační funkci si pak můžeme představit následovně:

$$f(net) = \begin{cases} 1 & \text{pro } w^{\mathrm{T}}x \geq 0 \\ 0 & \text{pro } w^{\mathrm{T}}x < 0 \end{cases} \tag{3.3}$$

Jak je z předpisu funkce patrné, pro jakékoli vstupní hodnoty větší nebo rovné 0 bude na výstupu hodnota 1, tzn. neuron bude aktivován. Při záporných hodnotách neuron aktivován nebude.

Spojováním více vrstev neuronů vznikají neuronové sítě, které jsou schopny řešit komplexnější úlohy. Skládají se ze tří hlavních druhů neuronových vrstev – vstupní, vnitřní (skryté), kterých je typicky několik, a výstupní. Vstupní vrstva přijímá data od vnějšího okolí a výsledek po průchodu dat neuronovou sítí je poté k dispozici na vrstvě výstupní. Neuronové sítě s velkým množstvím skrytých vrstev bývají také nazývány jako *hluboké neuronové sítě*.



Obrázek 3.1: Schéma obecného modelu umělého neuronu a vícevrstvé neuronové sítě.

Mezi základní typy neuronových sítí patří *feed-forward*, *rekurentní* a *konvoluční* neuronové sítě.

- **Feed-forward**, jak název napovídá, netvoří svými propojeními žádné cykly. Data tedy sítí prochází pouze jedním směrem – od vstupní vrstvy směrem k výstupní [22].
- **Rekurentní neuronová síť** obsahuje alespoň jeden cyklus, kdy se výstup neuronu propaguje směrem zpět. Často je tento výstup neuronu použit jako vstup téhož neuronu, čímž si neuron uchová informaci z předchozího kroku a chová se tedy jako malá paměťová buňka [22].
- **Konvoluční neuronové sítě** obsahují konvoluční jádra s váhami. Pomocí těchto jader se provádí konvoluce vstupními daty, díky čemuž je možné nalézat vzorce v datech. Tento typ neuronových sítí se často používá pro zpracování obrazu [18].

3.1.1 Aktivační funkce

Aby bylo možné naučit neuronové sítě komplexnějším závislostem, používají se složitější aktivační funkce. Jednou ze základních aktivačních funkcí je tzv. *sigmoida* [23], která má následující předpis:

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.4}$$

Další funkcí je tzv. ReLU [23], neboli *rectified linear unit*. Její předpis je

$$ReLU(x) = max(0, x), \tag{3.5}$$

tedy pokud jejím vstupem bude kladné číslo, bude toto číslo i výstupem. Pokud bude vstup ovšem záporný, výstupem bude hodnota 0. Neuronová síť s touto aktivační funkcí může občas trpět problémem zvaným *dying ReLU problem* [15]. To znamená, že některé neurony sítě začnou reagovat na jakýkoli vstup hodnotou 0. Jde o situaci, kterou nelze zvrátit přeučením neuronové sítě a může se týkat i několika desítek procent neuronů sítě. Řešením může být aktivační funkce leaky ReLU [23], která pracuje i se zápornými hodnotami.

V modelech neuronových sítí zaměřených pro zpracování řeči se také používá aktivační funkce GELU (*Gaussian Linear Unit*) [14]:

$$GELU(x) = 0{,}5x\left(1 + tanh\left(\sqrt{\frac{2}{\pi}}(x + 0{,}044715x^3)\right)\right), \tag{3.6}$$

která se ukázala jako účinná pro tuto oblast.

3.2 Učení neuronových sítí

Učení neuronových sítí probíhá nastavováním vah jednotlivých propojení neuronů na základě průchodu dat sítí. Tím v podstatě neuronům říkáme, jakým aspektům v datech mají věnovat větší či menší pozornost za účelem zlepšení přesnosti neuronové sítě. Nastavování vah se provádí pomocí algoritmu zpětné propagace (angl. *backpropagation*), který nastavuje váhy propojení neuronů v jednotlivých vrstvách v opačném směru, než v jakém procházela data sítí, díky čemuž se pak neuronová síť při učení zlepšuje.

Mezi základní přístupy učení neuronových sítí patří tzv. učení s učitelem (*supervised learning*) a učení bez učitele (*unsupervised learning*) [17].

3.2.1 Učení s učitelem

Pro učení s učitelem je typická existence vzorových označených dat (anotací) s cílem naučit síť rozpoznávat data, která nikdy dříve neviděla, na základě svých zkušeností s těmito označenými daty.

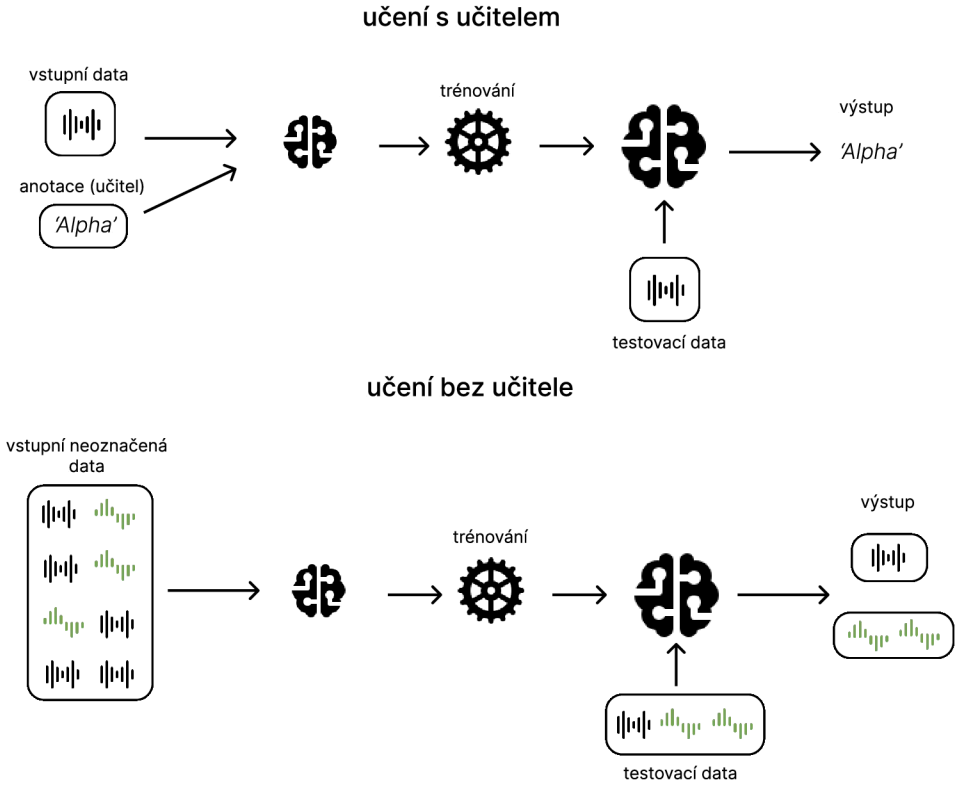
Při učení data prochází sítí a její výstup je pak porovnán se vzorovým výstupem pro tato data pomocí tzv. *loss* funkce, která představuje chybovost sítě na daných datech (podrobněji popsána v sekci 3.2.4). Pokud je výstup sítě stejný jako vzorový, síť pro tato data pracuje správně a není třeba ji měnit. Pokud se ale výstupy liší, je potřeba změnit nastavení vah jednotlivých propojení, aby síť začala na data reagovat správně.

Učení s učitelem se využívá pro řešení úloh pomocí neuronových sítí v oblastech jako rozpoznávání obrazu a počítačové vidění, nebo právě zpracování přirozeného jazyka, které je předmětem této práce.

3.2.2 Učení bez učitele

Učení bez učitele se na rozdíl od učení s učitelem liší v tom, že se síti při učení žádná vzorově označená data neposkytují. Učení pak probíhá tak, že síť sama vzory v datech vyhledává bez jakékoli pomoci vnějšího okolí [17]. Tato metoda učení se nejčastěji využívá pro klastrování, tj. roztřídění dat do několika skupin s podobnými vlastnostmi.

Obrázek 3.2 představuje příklad učení s učitelem a bez učitele. U učení s učitelem se model neuronové sítě naučí rozpoznávat data dle anotací (zde přepis řečeného slova). U učení bez učitele model dokáže roztřídit jednotlivé nahrávky do skupin, do kterých patří.



Obrázek 3.2: Příklad učení s učitelem a bez učitele.

3.2.3 Problémy spojené s učením

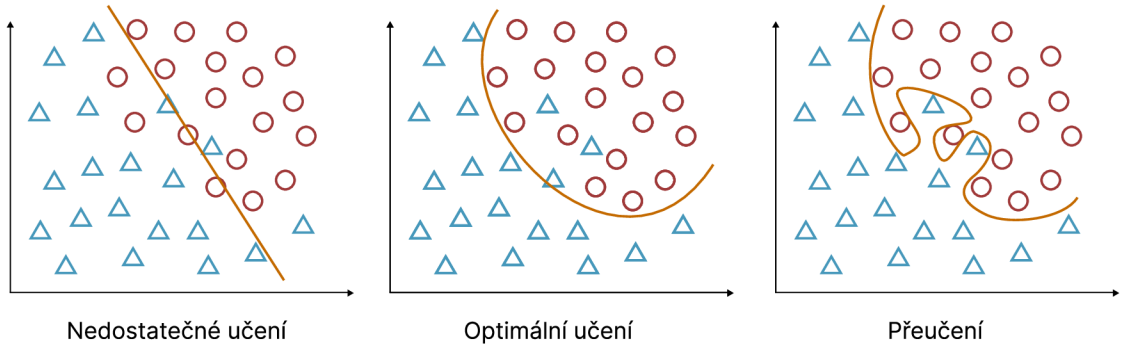
Z informací uvedených výše je zřejmé, že stejně jako lidé nepracují bezchybně, nelze bezchybnost očekávat ani od neuronové sítě. Ta pracuje, v případě učení s učitelem, především na základě svých zkušeností s daty. Zejména v oblastech počítačového vidění a zpracování řeči je nespočet různých vstupů, které může neuronová síť obdržet. Naučit síť na všech možných případech je nemožné, ale obecně platí, že jedním z nejdůležitějších aspektů, které značně zvyšují úspěšnost učení neuronových sítí, je právě velký objem a variabilita trénovacích dat, se kterými se síť při učení setká.

Samozřejmě lze (a někdy to jinak nejde, pokud jsou trénovací data dostupná pouze omezeně) učit neuronovou síť na menších objemech dat. V tom případě je ale potřeba počítat s pravděpodobnou menší úspěšností učení a také je nutné dávat pozor na možné přeučení sítě, tzv. *overfitting*. Málo trénovacích dat totiž pro síť může znamenat, že při učení nalezne v datech náhodné falešné vzory, které by jinak při více datech zanikly. Síť se adaptuje na tyto vzory a poté začne chybně reagovat na data, která při učení neviděla.

Žádný univerzální postup pro dosažení nejlepších výsledků při učení neuronových sítí však neexistuje. Vždy se z velké části jedná o kompromis. Pokud se za každou cenu budeme

snažit vyhnout přeučení, můžeme se setkat s opačným jevem, tzv. *underfitting* (nedostatečné učení). Ten se může vyskytnout, pokud model během krátkého času učení nestihne nalézt složitější vzory a vztahy v datech, nebo také proto, že je model zkrátka příliš jednoduchý pro odhalení komplexnějších závislostí [1].

Na obrázku 3.3 lze vidět problém nedostatečného učení a naopak přeučení neuronových sítí, demonstrovaný na jednoduché klasifikaci dat. Při nedostatečném učení síť špatně klasifikuje velké množství dat, jelikož nedokázala detailněji odhalit vzor výskytu dat. Při optimálním učení síť správně klasifikuje největší množství dat. Při přeučení se z hranice plynulé stává hranice nepřirozená a velmi přizpůsobená trénovacím datům, jejímž důsledkem bude chybná klasifikace dat v některých oblastech klasifikační roviny.

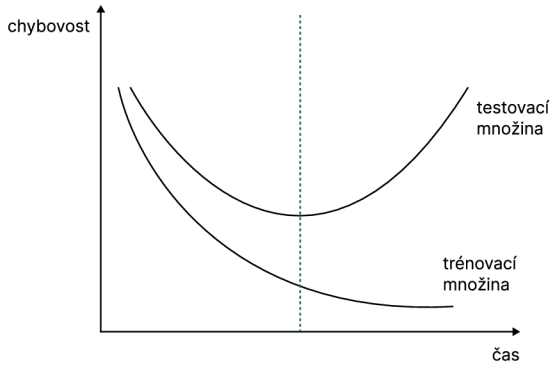


Obrázek 3.3: Nedostatečné učení a přeučení demonstrované na jednoduché klasifikaci dat.

3.2.4 Loss funkce

Zda dochází k přeučení neuronové sítě je možné zjistit už při samotném učení sledováním základních metrik, především tzv. *train loss* a *validation loss* (někdy také nazývána *test loss*) [1]. Jedná se o funkce, jejichž hodnoty reprezentují, jak přesně si model vede při zpracování dat. Train loss reprezentuje, do jaké míry síť správně reaguje na trénovací data. Obecně, čím je tato hodnota menší, tím více se síť přibližuje správným výstupům na těchto datech. Podobně funguje validation loss, která představuje chybovost na testovací množině. Při učení usilujeme o to, aby se neustále snižovaly obě tyto metriky – síť se učí na trénovacích datech, se kterými si vede čím dál tím lépe, a také se zároveň zlepšuje na testovací množině dat, která je pro síť neznámá, jelikož ji při učení neviděla. První známky přeučení ovšem přichází v moment, kdy se nadále snižuje train loss, ale začíná narůstat validation loss [1]. To znamená, že se síť příliš přizpůsobuje trénovacím datům na úkor těch neviděných a tím pádem může dojít ke zhoršení její generalizace.

Obrázek 3.4 znázorňuje typický vývoj přesnosti na trénovací a testovací množině během učení modelů. Nejprve se rapidně snižují obě metriky, ovšem v bodě označeném na grafu přerušovanou čarou se pokles hodnot loss funkce na testovací množině zastaví a od té chvíle se začne opět zvyšovat, což může indikovat přeučení.



Obrázek 3.4: Vývoj hodnot loss funkce na trénovací a testovací množině během učení.

Ve strojovém učení je výběr správné loss funkce důležitý pro efektivní učení modelu, jelikož každá z nich je vhodná pro specifické typy úloh. V oblasti klasifikačních modelů se nejčastěji používá tzv. *cross entropy loss*, nebo také *log loss* [9].

Pro klasifikaci do N tříd je předpis této funkce následující:

$$CrossEntropy = -\sum_{c=1}^{N} y_c \, log(softmax(z)_c) \tag{3.7}$$

Proměnná y_c se zde rovná hodnotě 1, pouze pokud vstup neuronové sítě patří do této třídy c. U všech ostatních tříd má hodnotu 0. Proměnná z představuje výstup poslední vrstvy neuronové sítě a $softmax(z)_c$ odhad pravděpodobnosti z tohoto výstupu, že vstup patří do třídy c [9], přičemž tato aktivační funkce má následující předpis [2]:

$$softmax(z)_c = \frac{e^{z_c}}{\sum_{j=1}^{N} e^{z_j}} \tag{3.8}$$

Tato funkce je aplikována na každý prvek z_i výstupního vektoru z neuronové sítě a výsledné hodnoty jsou normalizovány dělením součtem všech hodnot. Tímto způsobem je zajištěno, že součet složek výstupního vektoru bude 1.

Předpokládejme, že provádíme klasifikaci vysloveného písmena hláskovací abecedy. Pro jednoduchost budeme uvažovat pouze abecedu o čtyřech písmenech – *Alpha*, *Bravo*, *Charlie*, *Delta*. Tato písmena budou naše třídy, do kterých vyslovená slova budeme klasifikovat.

Bylo vysloveno slovo *Bravo*. Jeho vektor y proto bude mít následující podobu:

$$y = [0; 1; 0; 0],$$

což znamená, že správná klasifikace tohoto vyřčeného slova je třída *Bravo* (umístěná na druhé pozici). Tento vektor y právě reprezentuje ono označení dat člověkem, podle kterého se model učí.

Nechť výstupní vektor neuronové sítě pro toto vyřčené slovo je

$$z = [2{,}0; 1{,}0; -1{,}0; 0{,}5].$$

Aplikací softmax funkce dostaneme pravděpodobnosti pro každou třídu:

$$softmax(z) = [\frac{e^{2,0}}{S}; \frac{e^{1,0}}{S}; \frac{e^{-1,0}}{S}; \frac{e^{0,5}}{S}],$$

kde S je součet všech hodnot, tedy

$$e^{2,0} + e^{1,0} + e^{-1,0} + e^{0,5}.$$

Výpočet hodnoty cross entropy funkce bude pak následující:

$$CrossEntropy = -\sum_{c=1}^{4} y_c \, log(softmax(z)_c)$$

$$CrossEntropy = -(0 \cdot log(0{,}61) + 1 \cdot log(0{,}22) + 0 \cdot log(0{,}03) + 0 \cdot log(0{,}14))$$

$$CrossEntropy = -log(0{,}22)$$

$$CrossEntropy \approx 0{,}658$$

Pro tento výstup modelu bude hodnota loss funkce zhruba 0,658. Pokud by si model byl jist správnou odpovědí a pravděpodobnost $softmax(z)_{Bravo}$ by tedy byla 1, z předpisu cross entropy funkce je zjevné, že by její hodnota v takovém případě byla 0, čehož se při učení snažíme docílit.

Cross entropy je sice nejčastěji používaná funkce, ovšem studie provedená na Krakovské univerzitě z roku 2017 nastiňuje, že se nemusí vždy jednat o nejefektivnější loss funkci pro učení v oblasti klasifikačních úloh. Studie mimo jiné popisuje i několik jiných loss funkcí, které mají lepší výsledky než cross entropy [9].

Cross entropy funkce je použita také v modelu Whisper, na který se tato bakalářská práce zaměřuje. Whisper model je detailněji popsán v následující kapitole, avšak dle studie zaměřené na efektivní trénování modelů zpracování řeči není použití cross entropy funkcí pro modely zpracování řeči příliš vhodné. Její hodnoty totiž nemusí korelovat se sledovanými metrikami v této oblasti, např. chybovost slov (word error rate) nebo sémantická chybovost (semantic error rate) [20]. V takovém případě záleží, co je hlavním cílem učení. Pokud chceme dosáhnout co nejmenší chybovosti nehledě na to, jak moc si je model svými výstupy jistý, lze model trénovat i za cenu zvýšených hodnot loss funkce. Pokud je pro nás hlavní, aby si model svými predikcemi byl co nejvíce jist, měli bychom se soustředit na to, aby loss funkce na testovacím datasetu nenarůstala.

3.2.5 Zvýšení robustnosti sítě a zabránění přeučení

K tomu, aby bylo možné z trénovacích dat vytěžit maximum, ale zároveň nedošlo k přeučení sítě a dosáhlo se největší efektivity učení, si lze pomoci využitím různých parametrů učení neuronových sítí, tzv. *hyperparametrů*. Níže jsou představeny základní a pro tuto práci důležité hyperparametry.

Epochy

Za jeden ze základních hyperparametrů při učení modelů je považován počet epoch. Jedná se o nejjednodušší způsob, kterým lze měnit způsob učení modelu. Počet epoch znamená, kolikrát trénovací data během učení projdou přes model. Po jedné epoše tedy každý vzorek dat prošel modelem právě jednou a přispěl k nastavení vah jednotlivých neuronů [3].

Epochy se používají často u grafů znázorňujících výkonnost modelu během času. Jelikož čas není příliš vypovídající hodnota a s velikostí trénovacích dat a modelu se čas učení velmi mění, graf s využitím epoch na horizontální ose dokáže alespoň do jisté míry naznačit, jakým způsobem byl model učen.

Počet epoch je nutné zvolit adekvátně. Málo epoch může znamenat, že modelu nebude poskytnut dostatečný čas na nalezení vzorů v datech a model bude mít zbytečně velkou chybovost. Při velkém počtu epoch zase model uvidí trénovací data tolikrát, že se na ně příliš adaptuje a přeučí se.

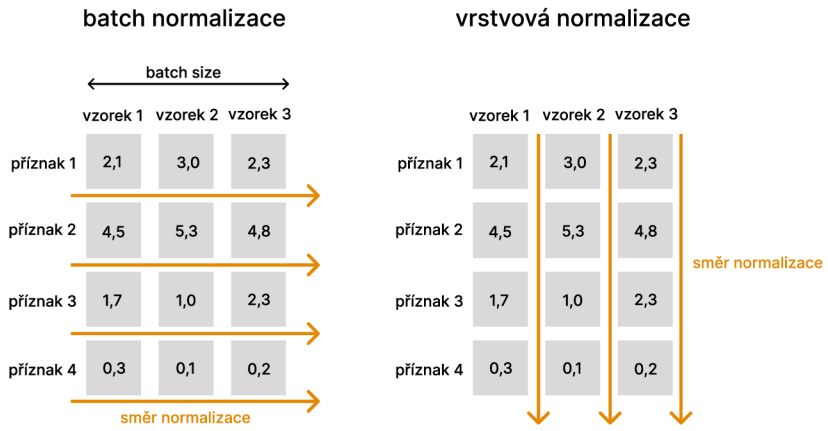
Batch size

Batch size představuje počet vzorků trénovacích dat, po kterém model pravidelně aktualizuje váhy. Batch size je zpravidla celé číslo větší nebo rovno 1 a menší nebo rovno počtu vzorků trénovacího datasetu [3]. Výhodou větších hodnot batch size je možnost paralelizace učení mezi více strojů, jejímž výsledkem je kratší čas potřebný k natrénování modelu. Nevýhodou ovšem je, že větší hodnoty batch size často vedou k nižším přesnostem modelu na testovacích datech.

Častým pojmem spojovaným s batch size je tzv. *batch normalizace*, která provádí normalizaci na základě průměru a směrodatné odchylky skrze jednotlivé příznaky všech vzorků dat v rámci jednoho batche a tím napomáhá konvergenci učení.

Dalším druhem normalizace, která se také při učení neuronových sítí používá, je tzv. *vrstvová normalizace*. Ta na rozdíl od batch normalizace provádí normalizaci napříč všemi příznaky pro jednotlivé vzorky. Je tedy nezávislá na velikosti batch.

Na obr. 3.5 je zobrazen rozdíl ve směru normalizace mezi batch a vrstvovou normalizací. Batch normalizace se používá nejčastěji u konvolučních neuronových sítí, zatímco vrstvová normalizace u rekurentních neuronových sítí a v architektuře transformer.



Obrázek 3.5: Rozdíl ve směru normalizace mezi batch a vrstvovou normalizací.

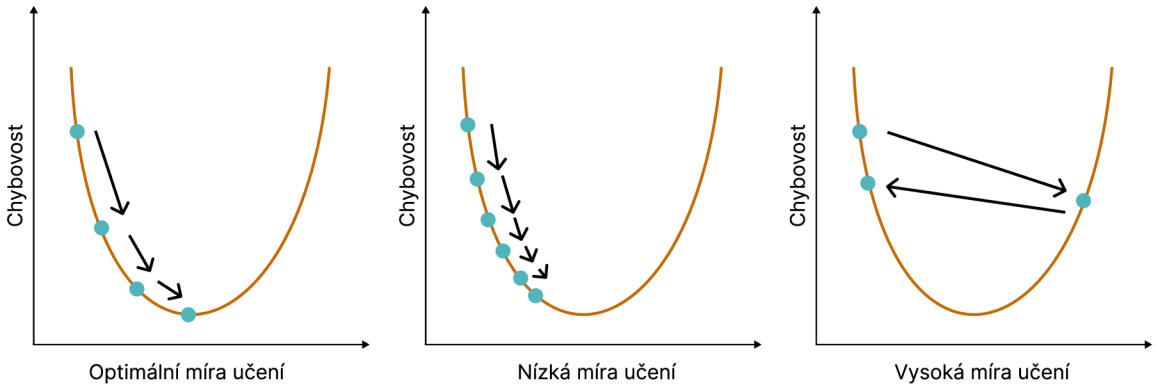
Learning rate

Learning rate, neboli míra učení, značí, o jak velký krok se změní hodnoty vah při jejich aktualizaci. Ovlivňuje tedy, jak rychle se model učí. Při nastavení nízké hodnoty se váhy

upravují až příliš opatrně a model buď vyžaduje delší čas učení, nebo se snadno zastaví v lokálním minimu, ze kterého se mu už nepodaří dostat.

Příliš vysoká hodnota může znamenat, že model opět nikdy nenalezne globální minimum. V tomto případě ale proto, že ho snadno může „přestřelit". Pokud je hodnota learning rate nastavená opravdu výrazně vysoko, může dojít i k divergenci učení a k výsledné horší přesnosti, než byla před učením. Důsledky nevhodného nastavení míry učení znázorňuje obrázek 3.6.

Při učení modelů se nejčastěji míra učení během učení snižuje, což má modelu pomoci nalézt globální minimum, pokud je k němu po uplynulé době učení blízko.



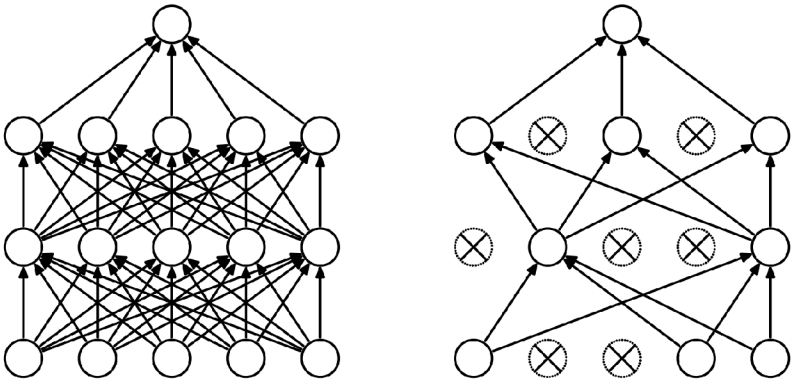
Obrázek 3.6: Nevhodně zvolená hodnota míry učení může způsobit nesprávné učení modelu.

Learning rate warmup

Tento hyperparametr souvisí s mírou učení. Jde o postupné zvyšování hodnoty míry učení v raných fázích trénování modelu až na cílovou hodnotu [6]. Metoda zvyšování míry učení pomáhá zachovat stabilitu učení a předchází tomu, aby se síť přeučila ze začátku trénování, kdy ještě není nijak adaptovaná na trénovací data. Nastavování tohoto hyperparametru probíhá nejčastěji jako počet kroků nebo relativní část doby učení, po kterou se má míra učení zvyšovat, než dosáhne svého maxima.

Dropout

Dropout je další metoda pomáhající zvýšit robustnost modelu a zabránit jeho přeučení. Funguje na základě dočasné deaktivace náhodných neuronů včetně všech jejich vstupních a výstupních spojení v síti (viz obrázek 3.7). Po deaktivaci těchto neuronů vznikne v podstatě menší verze daného modelu, která je trénována. Při evaluaci modelu jsou ale přítomny všechny neurony s tím, že pokud p je pravděpodobnost daného neuronu, že bude při trénování zachován, tak jeho výstupní váhy budou při evaluaci touto pravděpodobností p vynásobeny [25]. Pomocí hyperparametru se většinou nastavuje percentuální část z celkového počtu neuronů sítě, která bude během trénování deaktivována.



Obrázek 3.7: Neuronová síť před a po aplikaci metody dropout. Převzato z [25].

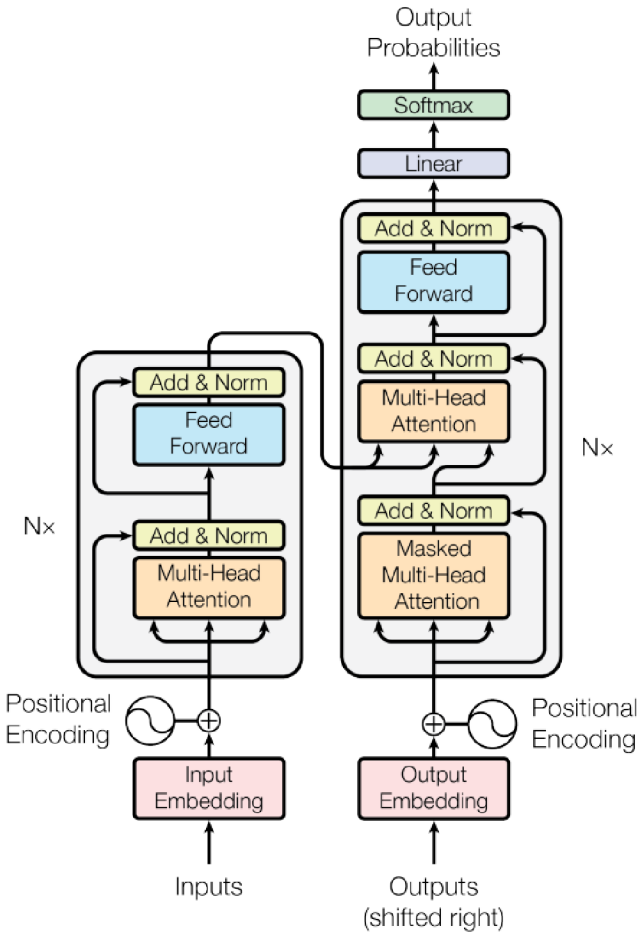
Nastavování správných hyperparametrů pro nejlepší učení neuronové sítě není přímočaré. Většinou jeden hyperparametr přímo souvisí s jedním či více jinými. Proto existuje několik přístupů ke zjištění optimálního nastavení, např. *grid search*, kdy jsou vyzkoušeny všechny možnosti nastavení hyperparametrů a pro každou možnost je provedeno trénování a vyhodnocení úspěšnosti. Tato metoda je ale použitelná především u modelů, které se učí velmi rychle, nebo nemají mnoho možností nastavení hodnot hyperparametrů. U modelů jako je Whisper, kde jedno trénování trvá několik hodin, je to téměř nerealizovatelné. V této práci proto prvotní nastavení hyperparametrů vychází ze vzorového skriptu pro učení Whisperu a následně jsou postupně vyzkoušeny jednotlivé změny hlavních hyperparametrů s cílem zjistit, zda mají na učení pozitivní vliv. Pokud ano, bude jejich nastavení ponecháno při dalších trénováních se změnou jiných hyperparametrů. Tímto způsobem sice pravděpodobně nenalezneme globálně nejlepší kombinaci nastavení na rozdíl od metody grid search, ale pomocí systematického přístupu ke změně hyperparametrů by mělo být možné přesnost modelu do určité míry vylepšit.

Kapitola 4

Transformery a model Whisper

Architektura transformer dnes patří mezi nejpoužívanější v oblasti modelů pro zpracování přirozeného jazyka. Byla představena v roce 2017 výzkumníky společnosti Google [27] a je známá především pro svůj výkon a efektivitu učení modelů, které ji využívají.

Transformery se skládají ze dvou hlavních částí, enkodéru a dekodéru, tvořících tzv. *encoder-decoder* strukturu (obr. 4.1). Ta se běžně používá pro zpracování řeči, textu či obrazu.



Obrázek 4.1: Schéma transformer architektury v originálním článku. Převzato z [27].

4.1 Zpracování vstupu

Zpracování vstupu začíná jeho převedením na tzv. *input embeddings*. To znamená, že jednotlivé prvky vstupu (např. malá časová okna vstupu, pokud se jedná o zpracování zvuku), jsou převedeny na n-dimenzionální vektory, jejichž vzájemná pozice v tomto n-dimenzionálním prostoru představuje jejich podobnost. Čím blíže jsou si dva prvky v tomto prostoru, tím podobnější jsou.

Při zpracování řeči před převodem na embeddings ještě probíhá nejčastěji převedení na Mel spektrogram. Převedení na embeddings je pak zajištěno pomocí malé konvoluční neuronové sítě.

4.2 Určení pořadí prvků

Klasické end-to-end modely jsou založené na rekurentních neuronových sítích, které zpracovávají data postupně v sekvenci, a díky tomu znají pořadí prvků, které zpracovávají. Jedním z rysů transformer architektury je ovšem to, že zpracovávají vstup po větších částech zároveň, a proto nemají o pořadí prvků žádné informace. Ty je k nim proto potřeba přidat. Jednou z nejčastějších metod pro řešení tohoto problému je tzv. *sinusoidal positional encoding*.

Nechť pos je pořadí prvku a d dimenze embeddings vektoru. Potom, i-tá složka pozicového vektoru daného prvku $\overrightarrow{PE}_{(pos,i)}$ je následující [26]:

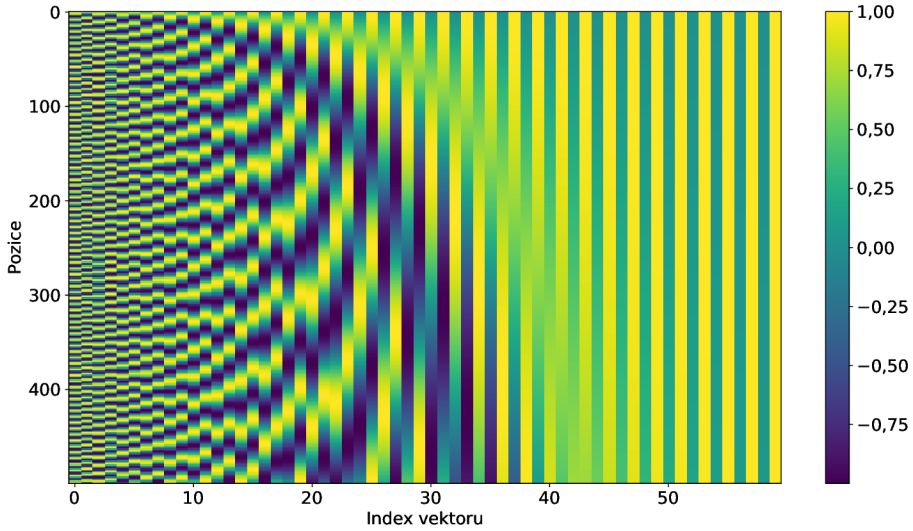
$$\overrightarrow{PE}_{(pos,i)} = \begin{cases} sin\left(\frac{pos}{10000^{\frac{2k}{d}}}\right) & \text{pro } i = 2k \\ cos\left(\frac{pos}{10000^{\frac{2k}{d}}}\right) & \text{pro } i = 2k+1 \end{cases} \tag{4.1}$$

Tento výsledný vektor si také můžeme představit následovně:

$$\overrightarrow{PE}_{(pos,i)} = \begin{bmatrix} sin\left(\frac{pos}{10000^{\frac{2\cdot 1}{d}}}\right) \\ cos\left(\frac{pos}{10000^{\frac{2\cdot 1}{d}}}\right) \\ \cdots \\ sin\left(\frac{pos}{10000^{\frac{2\cdot d/2}{d}}}\right) \\ cos\left(\frac{pos}{10000^{\frac{2\cdot d/2}{d}}}\right) \end{bmatrix} \tag{4.2}$$

Pro lepší demonstraci si můžeme dále vektor představit jako obdobu binárních čísel. První hodnota vektoru funguje podobně jako nejméně významný bit (LSB) binárního čísla, jelikož se jeho hodnota mění nejvýrazněji, zatímco poslední hodnota, kde $k = d/2$, slouží jako nejvíce významný bit (MSB).

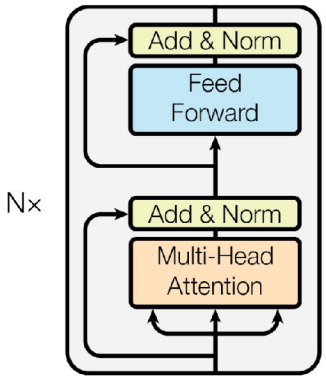
Příklad hodnot pozicového vektoru s dimenzí 60 pro prvních 500 prvků lze vidět na obr. 4.2. Každý řádek představuje hodnoty všech indexů vektoru pro prvek na dané pozici.



Obrázek 4.2: Příklad hodnot pozicového vektoru s dimenzí 60 pro 500 prvků.

4.3 Enkodér

Enkodér je složen z několika identických vrstev, přičemž každá vrstva obsahuje dvě další vrstvy – *multi-head attention* a *feed forward* vrstvu (obr. 4.3). Před předáním dat od jedné vrstvy do druhé je vždy ještě přítomna vrstvová normalizace výstupu.



Obrázek 4.3: Jedna z N vrstev enkodéru v transformer architektuře. Převzato z [27], upraveno.

Multi-head attention vrstva hledá vztahy mezi jednotlivými prvky vstupu, konkrétně na bázi tzv. *self-attention*, tedy porovnávání každého prvku se všemi ostatními. Vstup je zpracováván paralelně několika bloky tzv. hlav (proto i název multi-head attention).

Každá hlava zpracovává část vstupu. Více hlav zpracovávajících jednotlivé části vstupu umožňuje nalézat vzory a vztahy mezi daty v různých „podprostorech" a z různých pozic, což je efektivnější na rozdíl od situace, kdy je hlava pouze jedna [27]. Počet hlav závisí na konkrétních modelech stavěných na této architektuře, přičemž s rostoucím počtem hlav je možné sledovat složitější vlastnosti a závislosti. Např. nejmenší Whisper model s názvem Tiny má hlav 6, zatímco model Large disponuje již 20 hlavami v každé vrstvě [19].

V každé hlavě se nachází tři matice obsahující váhy, které se nazývají *query*, *key* a *value*. Vstup zpracovávaný konkrétní hlavou je násoben s každou z těchto matic – vzniknou tři matice Q, K a V. Tyto matice jsou zpracovány následovně:

$$Attention(Q, K, V) = softmax(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})V, \tag{4.3}$$

kde d_k je dimenze matice *key*. Výsledek celé multi-head attention vrstvy pak probíhá spojením všech výstupů hlav a vynásobení výsledku s poslední maticí vah W^O:

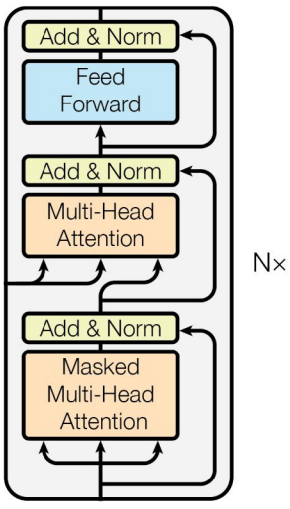
$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O, \tag{4.4}$$

kde $head_i$ je výstup attention funkce každé hlavy [27].

Výstup multi-head attention vrstvy prochází následně přes plně propojenou feed-forward neuronovou síť, která svůj výstup předá do další multi-head attention vrstvy.

4.4 Dekodér

Dekodér obsahuje opět, stejně jako enkodér, několik stejných vrstev. Každý blok dekodéru však obsahuje tři podvrstvy (obr. 4.4).



Obrázek 4.4: Jedna z N vrstev dekodéru v transformer architektuře. Převzato z [27], upraveno.

První podvrstvou je opět multi-head attention vrstva. Ta ale zpracovává, narozdíl od enkodéru, který pracuje se vstupem, předešlý výstup celého dekodéru. Ten má v sobě opět přidán pozicový vektor jednotlivých prvků. Průchod přes multi-head attention vrstvu probíhá stejně jako u enkodéru. Problém je ovšem v tom, že chceme, aby dekodér predikoval

následující token pouze na základě již vygenerovaných tokenů. Proto je vstup do softmax funkce ještě maskován. To se provádí jednoduchým vložením velmi záporných hodnot (ideálně $-\infty$) do matice na místa, která souvisí s následujícími tokeny. Díky tomu nebudou brány v potaz[1] a dekodér bude predikovat následující token pouze na základě již vygenerovaných tokenů [27].

Výstup první multi-head attention vrstvy je, jako všechny výstupy vrstev, normalizován a předán do druhé multi-head attention vrstvy.

Druhá vrstva obsahuje multi-head attention blok, který na rozdíl od všech ostatních multi-head attention bloků pracuje s různými vstupy. Pro výpočet matice Q je použit výstup z maskované multi-head attention vrstvy, zatímco matice K a V jsou získány za využití výstupu enkodéru [27]. Dále zpracování probíhá stejně jako u jiných multi-head attention bloků.

Výsledek z druhé vrstvy pokračuje do vrstvy třetí, která je také poslední. Jedná se opět o klasickou feed-forward neuronovou síť.

Výstup z dekodéru je transformován na vektor o velikosti slovníku (např. slov, pokud generujeme text) a po průchodu softmax funkcí výstupní vektor obsahuje pravděpodobnostní hodnoty pro jednotlivé prvky, které jsou následně generovány. S novým vygenerovaným prvkem, který měl v pravděpodobnostním vektoru nejvyšší hodnotu, a zbytkem již vygenerované výstupní sekvence dekodér poté pracuje při generování následujícího prvku.

4.5 Whisper

Whisper je open source model pro automatické rozpoznávání řeči představen společností OpenAI[2] v roce 2022. Byl trénován na 680 000 hodinách vícejazyčných dat, z toho pouze 192 hodin na českém jazyce [19]. Umožňuje jak přepis záznamů mluvené řeči různých jazyků do textu, tak textový překlad záznamů v těchto jazycích do angličtiny. Je trénován na nahrávkách s přepisy, které jsou dostupné na internetu, a tedy proto, že nebyl primárně učen na žádném konkrétním datasetu, je vhodný pro učení pro specifická použití.

4.5.1 Technologie

Whisper je stavěn na architektuře transformer (obr. 4.5). Vstup se vzorkovací frekvencí 16 kHz je rozdělen do částí po 30 sekundách. Pokud je vstup kratší než 30 sekund, je doplněn nulami do této délky (tzv. *zero padding*). Následně je převeden na Mel spektrogram. Ten je zpracován konvoluční sítí s GELU aktivační funkcí na input embeddings, ke kterým jsou přidány ještě pozicové vektory, a výsledek je předán do enkodéru.

Zpracování je prováděno klasickým způsobem transformer architektury, přičemž používanou aktivační funkcí v modelu je cross entropy a výstup dekodéru je pomocí BPE tokenizeru převeden na výsledný text. Kromě klasických tokenů reprezentujících text je na výstupu také několik speciálních netisknutých tokenů označujících například začátek přepisu, jazyk přepisu, typ výstupu (překlad či přepis) či časové známky. Jeho slovník, obsahující přes 50 000 tokenů, pak vypadá zhruba následovně:

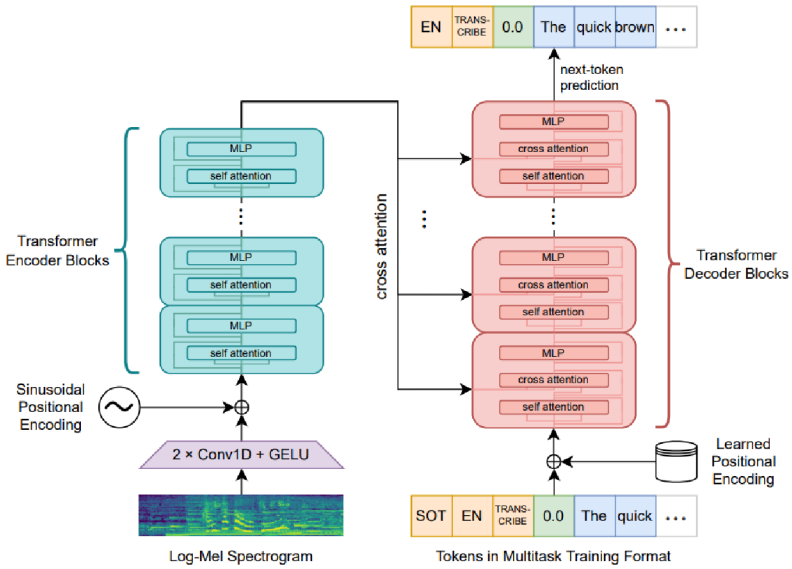
```
"Ġaccomplishments": 25943,
"ĠAnalytics": 25944,
```

[1] Dle předpisu funkce softmax popsané v kapitole 3 je zřejmé, že čím menší jednotlivé hodnoty budou, tím více se na výstupu softmax funkce budou blížit nule.

[2] https://openai.com/

```
"Ġshaping": 25945,
"reiben": 25946,
"Ġbachelor": 25947,
"Ġfingert": 25948
```

Písmeno Ġ je znak, kterým se v BPE znázorňují mezery. Pokud je tedy řetězec, který představuje token, většinou na začátku slova či je slovo sám o sobě, bude před ním pravděpodobně tento znak. Pokud se jedná např. o kořen slova, kterému často předchází i nějaká předložka, tj. místo mezery nějaká jiná písmena, znak Ġ na začátku mít nebude.



Obrázek 4.5: Transformer architektura modelu Whisper. Převzato z [19].

Whisper má 5 svých variant: Tiny, Base, Small, Medium a Large. Tyto modely se liší svými vlastnostmi, přičemž čím menší model, tím méně parametrů. Pro tuto práci byl vybrán model Whisper Medium, který disponuje 769 miliony parametry. Pro porovnání, největší model, Large, obsahuje přes 1,5 miliardy parametrů a model Tiny „pouhých" 39 milionů. Dle oficiálního článku [19] se v chybovosti Medium model od verze Large příliš neliší a navíc je výhodný v tom, že obsahuje dvakrát méně parametrů, díky čemuž ho lze rychleji a jednodušeji trénovat. Výchozí Whisper Medium (tzv. *baseline*) na datasetu Common Voice 9[3] dle téhož článku dosahuje pro české nahrávky WER 18,8 % a na datasetu Vox Populi[4] 18,4 %.

Samotný model byl převzat z Hugging Face[5], společností zabývající se tvorbou různých nástrojů pro modely strojového učení. Na svých stránkách mimo jiné poskytují velké množství datasetů nebo právě modelů.

[3] https://huggingface.co/datasets/mozilla-foundation/common_voice_9_0
[4] https://huggingface.co/datasets/facebook/voxpopuli
[5] https://huggingface.co/

Kapitola 5

Příprava dat a procesu trénování

Klíčovou částí učení modelů strojového učení je získání a předzpracování dat. V případě učení s učitelem, což je případ učení Whisperu, je to i doplnění vzorových výstupů k těmto datům. Následně je potřeba také nastavit a co nejvíce zautomatizovat proces přípravy datasetů a učení. Právě těmito aspekty se bude následující kapitola zabývat.

5.1 Trénovací data

Jak již bylo zmíněno v kapitole 2, záznamy letecké komunikace patří k poměrně nedostupným datům. Nejen, že je složité získat dataset nahrávek letecké komunikace s přidanými vzorovými přepisy, problém dělá už i získání samotných nahrávek bez přepisů.

Na internetu existují webové stránky, které zpřístupňují leteckou komunikaci různých letišť, bohužel je vše ale přenášeno pouze živě, navíc často s velmi nízkou kvalitou a žádné vhodné záznamy se veřejně neukládají. Nejjednodušším řešením bylo tedy požádat přímo některé z letišť, zda by nebylo ochotné tato vzácná data poskytnout.

5.1.1 Letiště Kunovice

Letiště Kunovice (LKKU) se jevilo jako ideální možnost. Jde o neveřejné mezinárodní letiště, které ve všední dny v časech 8-16 hodin poskytuje službu řízení letového provozu (ATC), jejímž cílem je zabraňování srážkám letadel zabezpečováním předepsaných rozstupů a předáváním informací o provozu [29]. Mimo tuto provozní dobu ve všedních dnech a o víkendech poskytuje na vyžádání letištní letovou informační službu (AFIS), kdy se známému provozu předávají informace o letišti, stavu pohybové plochy, druhu provozu, překážkách na letišti a v jeho blízkosti a meteorologických podmínkách [29]. Nejde tedy o službu řízení a od ATC se proto ve způsobu komunikace mezi stanovištěm věže a letadlovými stanicemi liší.

Na věži (stanovišti letových provozních služeb) pracují dva řídící letového provozu, kteří se střídají v poskytování služeb především v provozní době letiště, a několik dispečerů AFIS. Celkem se tedy na věži průběžně mění více než 10 lidí, samozřejmě s různou frekvencí služeb.

V blízkosti letiště sídlí místní aeroklub, který provozuje motorová i bezmotorová letadla. Dále se zde nachází několik leteckých servisů, výrobců sportovních letadel, kteří exportují letadla i do zahraničí, a přímo na letišti sídlí výrobce letadel L-410 Turbolet, kterému toto letiště také patří a využívá jej pro zkušební lety.

Všechny tyto aspekty, jako jsou rozmanitá frazeologie, velký počet různých mluvčích, různorodý provoz i kombinace české komunikace s anglickou by byly ideální pro učení co nejobecnějšího rozpoznávače letecké komunikace.

5.1.2 Získaná data

Kunovice, stejně jako všechna ostatní letiště, disponují záznamovým zařízením. Toto zařízení mimo hovorů na telefonních linkách a různých radiových kanálech zaznamenává také všechna vysílání řídícího letového provozu/dispečera AFIS i letadel na frekvenci letiště a tato data ukládá na záznamové médium ve formátu WAV.

Pro trénování modelu Whisper bylo letištěm Kunovice poskytnuto poměrně velké množství dat. Celkem byly získány záznamy 47 dní z roků 2022 a 2023, které dohromady mají více než 50 hodin, přičemž poměr českých a anglických nahrávek je zhruba 80:20. Ojediněle se pak u vysílání pilotů vyskytuje také slovenština.

5.2 Formát přepisů a anotace dat

Důležitým krokem před začátkem práce na anotaci dat bylo určit, v jakém tvaru by bylo vhodné záznamy komunikace přepisovat. Přepis všech slov přesně tak, jak šla ve vysílání po sobě (dále *plný tvar*), totiž automaticky neznamená ideální stav – jak bylo ukázáno v kap. 2, např. číselné hodnoty se v takové formě přepisu hledají jen velmi obtížně. Spolu s např. zkráceným označením drah a zkrácenými volacími znaky by se proto řídící jak přímo na věži, tak i zpětně při procházení záznamů, v přepisu orientoval mnohem rychleji a snadněji.

5.2.1 Analýza formátu přepisů letecké komunikace

Určení formátu zkráceného tvaru přepisu bylo konzultováno s kunovickými řídícími letového provozu i dispečery AFIS. Výsledkem byl seznam informací, které by byly v přepisu nahrávky ideální pro zkrácení.

Zkrácení postihlo prakticky všechny číselné údaje, přičemž jednotky těchto údajů byly zachovány v původním tvaru. Dále se zkracují volací znaky letadel a označení pozice dráhy (dráha levá, střední a pravá) se přidává ke směru dráhy:

Předmět zkracování	Plný přepis	Zkrácený přepis
Označení dráhy	nula dva levá	02L
Směr větru	dva šest nula stupňů	260 stupňů
Síla větru	jedna nula uzlů	10 uzlů
Atmosférický tlak	QNH jedna nula jedna šest	QNH 1016
Vzdálenost	tři míle	3 míle
Výška	tři tisíce tři sta stop	3300 stop
Čas	pět nula minut	50 minut
Frekvence	jedna tři šest čárka dva sedm pět	136,275
Volací znaky	Oscar Kilo Alfa Bravo Charlie Jedna Nula	OKABC10

Tabulka 5.1: Příklady zvoleného zkracování komunikace.

Běžně používané zkratky (QNH, CAVOK, apod.) jsou u obou tvarů přepisů ve stejném, zaužívaném, tvaru. Příklad přepisu na obou formách může vypadat následovně:

Plný přepis: `Oscar Kilo Alpha Bravo Charlie dráha nula dva střední vzlet povolen vítr nula jedna nula stupňů pět uzlů`

Zkrácený přepis: `OKABC dráha 02C vzlet povolen vítr 010 stupňů 5 uzlů`

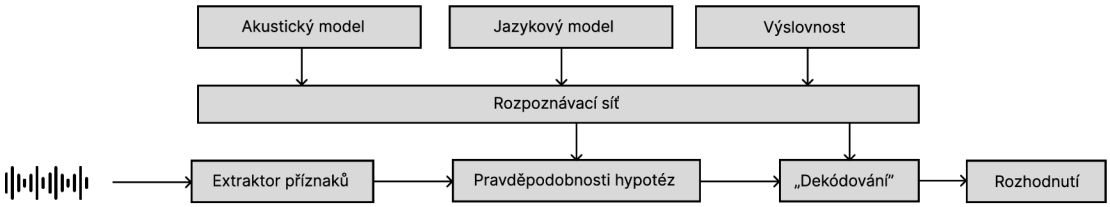
Je zřejmé, že model strojového učení, který je předtrénovaný na datech normální řeči a je naučen přepisovat každé slovo tak, jak bylo vyřčeno, bude mít s učením se na zkrácené formě větší problém než s formou plnou. Proto je učení prováděno na obou formách paralelně a jedním z bodů analýzy bude porovnat, jak si model s těmito rozdílnými přepisy vede.

5.2.2 Anotace dat

Vzhledem k velkému množství dat bylo potřeba co nejvíce zautomatizovat proces přepisu dat. Proto bylo pro anotaci využito verze nástroje SpokenData[1] vyvinuté pro projekt ATCO2. Anotování probíhalo v několika etapách.

Nejdříve vždy proběhla automatická anotace dat pomocí automatického rozpoznávače řeči českého jazyka z Fakulty informačních technologií VUT v Brně (FIT). Tento rozpoznávač je založen na hybridní architektuře CNN-TDNN-HMM [11] doplněné o n-gramový jazykový model. Byl trénován na 1043 hodinách řeči složené z monologů i dialogů, přičemž v těchto trénovacích datech se nevyskytovaly žádné promluvy letecké komunikace. Tato trénovací data jsou částečně databáze vlastněné FIT (Specon, Temic, Bison), částečně veřejně dostupné (ParCzech, Mozzila CommonVoice).

Architektura rozpoznávače je zobrazena na obrázku 5.1. K modelování akustických jednotek byla použita Factorized Time Delay Neural Networks (TDNN-F) architektura neuronových sítí. Ta dobře pracuje s dlouhým časovým kontextem a je i robustní pro menší objem trénovacích dat. Akustický model obsahuje 6 konvolučních a 19 TDNN-F vrstev. Dekodér pak využívá statickou rozpoznávací síť modelující všechny posloupnosti slov, které může rozpoznávač vygenerovat.



Obrázek 5.1: Architektura rozpoznávače použitého pro automatickou anotaci.

Pro úvodní iteraci přepisu záznamů letecké komunikace bylo využito jen 10 promluv. Automatické přepisy vygenerované tímto rozpoznávačem byly následně manuálně opraveny a přepsány do formátu, ze kterého se dal později extrahovat jak přepis plného textu, tak zkráceného (plný přepis se nacházel v závorce, které předcházel zkrácený tvar jejího obsahu). Poté byl vždy automatický rozpoznávač adaptován vedoucím této práce, během čehož byly

[1] https://www.spokendata.com/

manuální anotace použity pro adaptaci jazykového modelu a slovníku. Ten byl interpolován s původním jazykovým modelem.

Po adaptaci vždy následovala další etapa automatického přepisu s manuální anotací s následnou adaptací modelu. Díky tomu byl proces anotace pokaždé o něco rychlejší, jelikož model při automatické anotaci dělal méně chyb a manuální anotace byla etapu od etapy jednodušší. Celkem proběhlo 5 etap přepisů s adaptací, přičemž průběh WER modelu na českých a anglických nahrávkách se nachází v tabulce níže:

Etapa	1	2	3	4	5
Počet nahrávek od předešlé etapy	0	91	200	500	500
WER – český jazyk	70,1 %	53,6 %	42,1 %	41,8 %	44,9 %
WER – anglický jazyk	94,1 %	94,3 %	82,1 %	76,9 %	76,9 %

Tabulka 5.2: Průběh word error rate automatického rozpoznávače řeči na českých a anglických nahrávkách z Kunovic.

Celkem bylo anotováno zhruba 2 300 kunovických nahrávek. Posledních 1 000 však není zobrazeno v tabulce, protože rozpoznávač na nich už nebyl přeučen. Zvýšená chybovost v poslední etapě může být způsobena evaluací této chybovosti na jiném typu nahrávek letecké komunikace než v předešlých etapách – nahrávky v této etapě byly delší, složitější a vyskytovala se v nich často volná řeč.

Kromě přepisů v obou formách byly ještě při anotaci doplněny informace o jazyce a daném mluvčím. Pokud se v nahrávce jednalo o letadlo, byl do kolonky mluvčího doplněn jeho volací znak. Pokud v nahrávce mluvil řídící nebo dispečer AFIS, byli identifikováni podle hlasu a v kolonce tak bylo doplněno jejich pseudo ID. Takto bylo identifikováno 8 nejčastěji se vyskytujících mluvčích z věže. Speciální, deváté, pseudo ID bylo pro řídící či dispečery AFIS, u kterých buď nebyla jistota, o jakého mluvčího z výše zmíněných osmi se jedná, nebo pro všechny ostatní mluvčí z věže, kteří se vyskytovali pouze ojediněle.

Mimo anotaci kunovických nahrávek bylo anotováno také zhruba 150 nahrávek ze stanovišť Prahy (LKPR) a Brna (LKTB). Jejich hlavní účel byl však pro vytvoření testovací množiny (sekce 5.3), na které bude moci být model evaluován, nikoli přímo pro trénování. Nahrávky jsou pouze české a jejich přepis byl prováděn až po poslední etapě. WER automatického přepisu rozpoznávače na těchto datech bylo 43,36 %.

5.3 Testovací množiny

V průběhu anotace byly dohodnuty testovací množiny, na kterých bude trénovaný model evaluován. To pomůže k získání lepší představy o průběhu učení a přesnosti učeného modelu:

- **Nevidění mluvčí věže z LKKU** – část kunovických řídících a dispečerů AFIS byla vyčleněna pro testovací množinu, zatímco zbytek byl využit pro učení. Rozdělení bylo vytvořeno tak, aby bylo co nejvíce rovnoměrné a „spravedlivé". V každé množině se tedy nachází 4 mluvčí – jeden řídící a tři dispečeři AFIS (dva muži a jedna žena). Speciální, devátá, kategorie neznámých mluvčích z věže nebyla použita pro učení ani pro evaluaci, aby se nemohlo stát, že se omylem některý mluvčí z jedné skupiny dostane do druhé.

- **Neviděná vysílání letadel z LKKU** – do této testovací množiny spadají nahrávky vysílání letadel z Kunovic, které model při učení neviděl. Vzhledem k tomu, že někteří

piloti létají často, je pravděpodobné, že se někteří z nich vyskytnou jak v testovací, tak v trénovací množině. Jejich výskyt pouze v jedné množině však není ani cílem. U těchto nahrávek jde spíše o to, jak si model umí poradit s šumem, který je u vysílání letadel značný na rozdíl od vysílání z věže, která jsou prakticky čistá.

- **Nahrávky z LKPR a LKTB** – tato množina obsahuje české nahrávky nahrané ze stanovišť Brna a Prahy. Jejich způsob nahrávání je však jiný od dat z Kunovic (tato data jsou pouze odchytávána, nepochází přímo ze stanoviště), často jsou méně hlasité nebo značně zašuměné. Také se tam vyskytují zcela jiní mluvčí a rozdíl je mimo jiné i v obsahu nahrávek (letiště Brno například disponuje dráhami s označením 09 a 27, zatímco Kunovice 02 a 20), na který model není zvyklý.

5.4 Příprava datasetů

Po anotaci dat byly od vedoucího obdrženy surové přepisy ze SpokenData. Dalším úkolem bylo tedy tato data zpracovat – extrahovat přepisy pro plný i zkrácený tvar a vytvořit datasety pro učení i evaluaci modelu. Samotné přepisy jsou ve formě XML souborů, jejichž vzorová stavba je uvedena níže (výp. 5.1).

```
<?xml version="1.0" encoding="utf-8"?>
        <data>
                <segment>
                        <start>0</start>
                        <end>2.6</end>
                        <speaker>D</speaker>
                        <speaker_label>OKMLA</speaker_label>
                        <text>
                            Kunovice Věž OKMLA(Oscar Kilo Mike Lima Alpha)
                            finále 20C(dva nula střední)
                        </text>
                        <tags>
                            <non_english>1</non_english>
                        </tags>
                </segment>
                <segment>
                        <start>3.2</start>
                        <end>6.2</end>
                        <speaker>A</speaker>
                        <speaker_label>ATCO tower 07</speaker_label>
                        <text>
                            OKMLA(Oscar Kilo Mike Lima Alpha) 20C(dva nula
                            střední) přistání povoleno vítr klid
                        </text>
                        <tags>
                            <non_english>1</non_english>
                        </tags>
                </segment>
        </data>
```

Výpis 5.1: Příklad zpracovávaného XML souboru.

Každé vysílání v nahrávce je reprezentováno jedním segmentem. Často jich je v jedné nahrávce více. Důležitou značkou je `speaker_label`, která uchovává informace o řečníkovi. V příkladu tedy figuruje letadlo s volacím znakem OKMLA a řídící se pseudo ID č. 07. Pomocí `non_english` značky v každém segmentu je také určen jazyk vysílání. Anglická vysílání mají tuto hodnotu nastavenou na 0, česká a slovenská na 1. Aby bylo možné česká a slovenská vysílání rozpoznat, vysílání ve slovenském jazyce mají na začátku přepisu v textu navíc přidanou značku `[Slovak]`.

5.4.1 Extrakce přepisů v požadovaném tvaru

Pro účely trénování rozpoznávače, který prováděl automatický přepis nahrávek před manuální anotací na SpokenData, byl vytvořen skript, který zpracovává všechny přepisy ve formátu XML a extrahuje z nich pouze plný přepis ve formě textového souboru. Pro samotnou tvorbu datasetů a lepší organizaci však byl vytvořen skript druhý. Ten extrahuje přepisy pro plný či zkrácený tvar ve formě souborů JSON. Z XML souborů také zjišťuje identifikaci mluvčího z věže. Pokud se v nahrávce vyskytuje alespoň jedno vysílání nějakého řídícího či dispečera AFIS, je ve výstupním souboru doplněno jeho pseudo ID. V opačném případě je na místě ID doplněna hodnota `null`.

Jelikož se v jedné nahrávce často vyskytuje více než jedno vysílání, pro zajímavost a částečně i přehlednost byl výsledný přepis celé nahrávky koncipován tak, že každé vysílání z dané nahrávky je na samostatném řádku. Tím se prakticky rozlišují jednotliví řečníci vysílání – tzv. *diarizace*. To, zda se model oddělovat vysílání naučí, je ale pouze pokus, jak dokáže zvládat i jiné úkoly než pouhý přepis, a úspěšnost modelu v tomto není vyhodnocována. Příklad přepisu pro jednu takovou nahrávku se dvěma vysíláními může být následující:

```
Oscar Papa Lima dráha dva nula střední přistání povoleno vítr klid
přistání povoleno Oscar Papa Lima
```

5.4.2 Tvorba datasetů

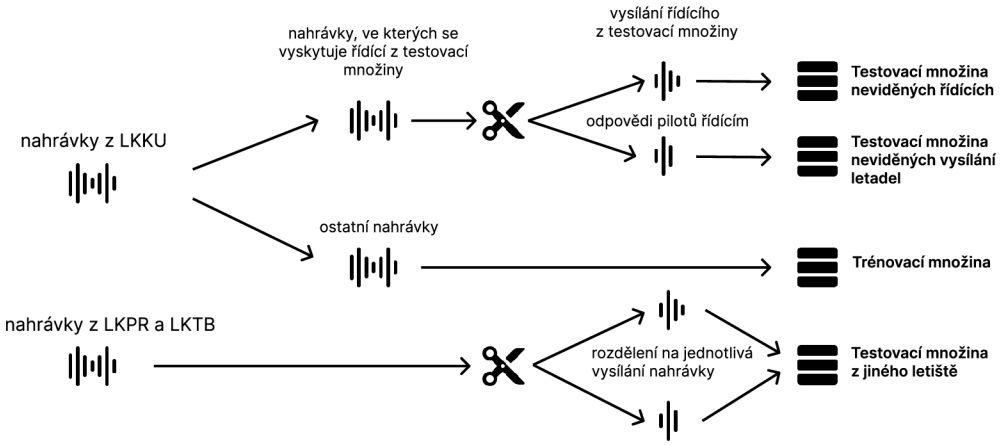
Pro tvorbu konečných datasetů slouží skript `dataset_prepare.py`. Ten načítá JSON soubory po extrakci přepisů. Ke každému přepisu spáruje nahrávku ve formátu WAV, ke které se doplní informace o vzorkovací frekvenci a cestě k dané nahrávce.

Skript pracuje ve dvou režimech. V prvním režimu tvoří trénovací dataset, a proto přepisy, u kterých je doplněno pseudo ID řídícího, které patří někomu z testovací množiny nebo je z neznámé kategorie, do datasetu nepřidá. Druhý režim je pak pro tvorbu datasetů testovacích množin. V takovém případě vytváří dataset ze všech poskytnutých nahrávek, ke kterým je ještě přidáno označení jazyka. Celý dataset je pak tvořen jako seznam složený z jednotlivých prvků nahrávek. Jedna nahrávka pro trénovací množinu (bez doplněného jazyka) je tedy v datasetu reprezentována následovně:

```
{
    'audio': = {
        'path': String,
        'array': List[Float],
        'sampling_rate': Int
        },
    'sentence': String
}
```

5.4.3 Datasety testovacích množin

Pro vytvoření datasetů s neviděnými řídícími a neviděnými vysíláními letadel jsou využity všechny nahrávky, v nichž se alespoň v jednom vysílání nahrávky vyskytuje řídící s pseudo ID z testovací množiny. Aby však bylo vyhodnocení natrénovaných modelů dle testovacích množin co nejkonkrétnější, jednotlivé nahrávky bylo vhodné rozdělit dle jazyka, kterým se v nich mluví. Ovšem v jedné nahrávce se může vyskytovat více vysílání a může se stát, že každé z nich bude v jiném jazyce. Proto jsou jednotlivá vysílání v nahrávce dle časových známek ze surových XML souborů rozstříhána. Vysílání řídícího je pak použito pro dataset s neviděnými řídícími a vysílání pilota (např. odpověď řídícímu), pokud v nahrávce je, se zařadí do množiny neviděných vysílání letadel (obr. 5.2). Díky tomu jsou všechny nahrávky využity prakticky ze 100 % a pro tvorbu datasetu neviděných vysílání letadel není nutné používat nahrávky, které lze rovnou bez jakýchkoli úprav použít pro trénování.



Obrázek 5.2: Rozdělení všech nahrávek do jednotlivých množin.

Nahrávky z LKPR a LKTB, určené pro testovací množinu z jiného letiště, jsou zpracovány stejným způsobem jako nahrávky, ve kterých se vyskytuje vysílání řídícího z testovací množiny. Jediný rozdíl je pouze v tom, že ze všech těchto nahrávek je poté vytvořen pouze jeden dataset a nedělí se na řídící a letadla.

Výsledné složení testovacích množin lze vidět v tabulce 5.3, která poskytuje i detailní informace o jednotlivých množinách. Vzhledem k tomu, že jak bylo zmíněno výše, testovací množiny neviděných řídících a neviděných vysílání letadel byly tvořeny z jedné množiny nahrávek (souborů WAV), mají tyto testovací množiny stejnou hodnotu počtu nahrávek. Celkový počet vysílání označuje počet promluv. Počet stanic reprezentuje, kolik různých letadlových stanic/řídících z Kunovic se v nahrávkách objevilo. Jelikož každý řídící z Kunovic měl své pseudo ID, každý je započítán zvlášť. Protože se ale u vysílání letadel v přepisech zaznamenává pouze volací znak a v jednom letadle se může střídat více lidí/jeden člověk může létat ve více letadlech, je toto číslo pouze orientační a nelze ho označit za počet řečníků.

V množině neviděných nahrávek se objevily i 3 slovenské nahrávky. Ty tam byly pro zajímavost zachovány, jelikož slovenské nahrávky zůstaly i v trénovacích datasetech. Slovenských dat je však tak málo, že se modely na slovenských nahrávkách vůbec nezlepšovaly, a proto nejsou ani zmíněné při evaluaci modelů.

Množina	Nev. řídící	Nev. vys. letadel	Nahr. z jin. letiště
Počet nahrávek	330	330	142
Vysílání celkem	340	188	236
z toho česky	275	152	236
z toho anglicky	65	33	0
z toho slovensky	0	3	0
Počet slov (plná forma)	3 920	2 277	3 347
Počet slov (zkr. forma)	2 621	1 616	2 209
Počet stanic	4	81	43
Počet minut	26,0	15,5	20,95

Tabulka 5.3: Složení testovacích množin.

5.4.4 Evaluace modelů

Posledním hlavním pomocným skriptem je evaluátor modelu. Nechává přepsat nahrávky z testovacích množin modelem, následně je porovná s referenčním přepisem a na výstupu tiskne WER pro každou testovací množinu. Ve výstupní tabulce pro danou testovací množinu je také rozdělení dle jazyka a jednotlivá chybovost na těchto jazycích.

5.5 Použité technologie pro učení

Hugging Face poskytuje přímo na svém blogu článek s příkladem skriptu pro učení Whisperu[2]. Jelikož byl Whisper vydán v roce 2022 a jedná se tedy o poměrně nový model, na internetu není mnoho implementací pro jeho učení. Avšak i přesto naprostá většina z nich používá obdobu tohoto vzorového skriptu a je považován za poměrně kvalitní. Proto z něj skript pro učení Whisperu v této práci vychází také.

Skript používá ve velké míře knihovnu Transformers[3]. Ta je určena pro trénování modelů zpracování řeči, počítačového vidění, zvuku a dalších. Nabízí podporu několika desítek konkrétních modelů.

5.5.1 Prostředí

Vzhledem k tomu, že Whisper je poměrně velký model a na běžných počítačích tyto modely téměř nelze trénovat, pro ověření funkčnosti trénovacího skriptu a doladění detailů byl Whisper nejdříve trénován na Google Colab[4], který poskytuje prostředí s grafickými kartami. Toto prostředí je sice vhodné na jednorázové trénování, ale v delším časovém horizontu je téměř nepoužitelné a ne příliš uživatelsky přívětivé. Proto byla trénování následně přesunuta na univerzitní výpočetní cluster, díky kterému bylo možné spouštět více instancí trénování zároveň bez nutnosti dohledu na ně.

[2] https://huggingface.co/blog/fine-tune-whisper
[3] https://huggingface.co/docs/transformers/index
[4] https://colab.research.google.com/

Kapitola 6

Učení

Před učením bylo vyhodnoceno WER na základním modelu Whisper Medium pomocí evaluačního skriptu na všech třech testovacích množinách (tab. 6.1). Celková hodnota WER zvýrazněná v tabulce je počítána váženým průměrem hodnot WER jednotlivých test. množin vzhledem k počtu nahrávek v těchto testovacích množinách z Kunovic (LKKU). Toto WER bude hlavní hodnota pro porovnávání modelů při trénování, jelikož nahrávky, na základě kterých je toto WER získáno, pocházejí z prostředí, na které je model trénován. Díky zohlednění počtu nahrávek v jednotlivých testovacích množinách z Kunovic je výsledné WER obrazem chybovosti při reálném provozu v Kunovicích, kde je poměr vysílání řídících/letadel podobný jako v těchto testovacích množinách. WER dle jazyků je také vypočítáno váženým průměrem vzhledem k počtu nahrávek jednotlivých množin z Kunovic. Množina nahrávek z jiného letiště pak slouží pro informaci, jak dobře dokáže model generalizovat. V této množině se totiž nejedná pouze o jiné mluvčí, ale také o jiný způsob komunikace a často frazeologii, která se v Kunovicích nepoužívá.

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Plné přepisy					
Čeština	91,6 %	88,4 %	90,5 %	**100,8 %**	84,0 %
Angličtina	151,5 %	127,4 %	143,4 %		-
Zkrácené přepisy					
Čeština	106,2 %	100,5 %	104,2 %	**120,0** %	100,0 %
Angličtina	197,7 %	166,1 %	187,1 %		-

Tabulka 6.1: WER základního Whisperu na testovacích množinách pro plné a zkrácené přepisy.

Velká chybovost v angličtině je z velké části způsobena nastavením pro český přepis. Ten je nastaven hned ze dvou důvodů. Prvním je, že bez nastaveného jazyka Whisper model téměř nedokázal rozeznat jazyk, natož správně přepsat některá slova. Buď nepřepsal nahrávky vůbec, nebo některá slova, kterým rozuměl, přepsal anglicky, i přesto, že v kódu byl model nastaven do režimu `transcribe`, nikoli `translate`. Také míchal několik různých jazyků jako arabštinu, ruštinu či čínštinu. Často byl přepis nahrávky složen pouze ze speciálních znaků. Tyto jevy pomohlo vyřešit právě nastavení českého jazyka pro přepis.

Dalším důvodem bylo, že přetrénované modely bez nastavení jazyka z neznámého důvodu začaly odmítat přepisovat jakékoli záznamy letecké komunikace a vracely pouze prázdné řetězce. Navíc se tato situace děla pouze u modelu Medium – např. Small model s tímto žádný problém neměl, i když specifikace jazyka nastavena nebyla.

Kvůli specifikaci jazyka se pak výchozí model snažil všechny, i anglické, nahrávky přepisovat pouze česky. Aby ale byla zjištěna pravá chybovost na angličtině, byl model evaluován také při nastavení jazyka anglického. U anglických plných přepisů v poté dosáhl model chybovosti 77,7 %, u zkrácených přepisů 84,3 %. Při učení a dalších evaluacích byl však už vždy nastaven jazyk český.

Nastavení jazyka ale nebyl jediný problém, který Whisper měl. Chybovost také do velké míry zvyšovalo, když se u modelu občas objevily tzv. *halucinace*, kdy se zacyklil a začal generovat jedno nebo více slov pořád dokola nebo pouze s menšími změnami:

Referenční přepis: `Kunovice Věž Oscar Kilo Victor Whiskey Charlie Tři Devět finále dráhy v Boršicích a plné přistání`
Vygenerovaný přepis: `...výšli na zvuk, když se věříte na finálu, výšli na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na finálu, věříte na zvuk, když se věříte na...`

Celý přepis měl hodnotu WER 762,5 % a je zřejmé, že pár podobných přepisů dokáže chybovost celé množiny výrazně ovlivnit. Když se z evaluace odebraly všechny přepisy s WER větší než 300 %, které většinou byly právě halucinace, celková chybovost modelu na plných přepisech se snížila na 85,6 % a na zkrácených přepisech na 98,3 %.

6.1 Učení po iteracích

Jelikož anotace dat probíhala po jednotlivých etapách, byla po částech zprvu vyhodnocována také chybovost modelů na postupně získaných datech. Iterace byly celkem tři:

- **1. iterace** – přepisy z prvních tří etap, celkem 572 nahrávek
- **2. iterace** – přepisy z předchozích etap s přepisy ze čtvrté etapy, 346 nových nahrávek, celkem 918 nahrávek
- **3. iterace** – přepisy z předchozích etap s přepisy z páté etapy, 644 nových nahrávek, celkem 1562 nahrávek

Detailní informace o nahrávkách, na kterých bylo prováděno trénování v jednotlivých iteracích, lze vidět níže v tabulce 6.2.

Tím, že anotovaných dat bylo celkově málo, bylo potřeba co nejvíce nahrávek, které jsou použitelné pro trénování (není v nich řídící patřící do testovací množiny neviděných řídících), zachovat a neubírat z nich kvůli např. tvorbě testovacích množin pro učení. Proto nahrávky z Kunovic, ze kterých vznikly testovací množiny pro evaluaci modelů, byly také ve formě datasetů použity v testovací množině pro učení. Protože ale těchto nahrávek, které se nedaly použít pro trénování, bylo dostatečné množství již po druhé iteraci a zvětšování objemu testovací množiny by vedlo k delšímu času evaluace při trénování i poté evaluačním

Iterace	1. iterace	2. iterace	3. iterace
Počet nahrávek	572	918	1 562
Vysílání celkem	713	1 146	2 497
z toho česky	607	975	2 083
z toho anglicky	92	150	355
z toho slovensky	14	21	59
Počet slov (plná forma)	8 454	13 857	40 299
Počet slov (zkrácená forma)	5 762	9 503	28 845
Počet stanic	174	231	367
Počet minut	59,8	98,4	289,8

Tabulka 6.2: Detailní informace o nahrávkách použitých pro trénování v jednotlivých iteracích.

skriptem, ze třetí iterace tyto nahrávky již pro testovací množinu při učení ani při evaluaci evaluačním skriptem použity nebyly.

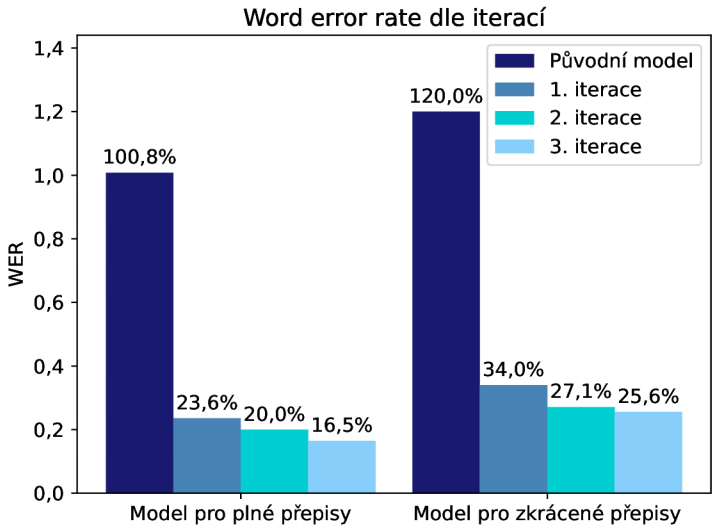
Nastavení trénovacích hyperparametrů bylo inspirováno původním skriptem převzatým z Hugging Face s několika změnami. Délka trénování byla oproti původnímu skriptu změněna na epochy místo kroků – počet kroků si model volí sám podle množství dat. Warmup byl v původním skriptu nastaven na 10 % doby trénování (500 kroků z 5000), zde byl zvýšen na 12 %. Míra učení se při výchozím nastavení po dosažení maxima lineárně snižuje až do konce trénování. Batch size je nastaveno na hodnotu 1 s tím, že hyperparametr `gradient_accumulation_steps` je nastaven na hodnotu 16. Díky němu je možné trénovat modely na případných větších hodnotách batch size i na méně výkonných strojích tím, že úprava vah se provádí až po několika iteracích batch.

Vyhodnocování WER je nastaveno, aby se provádělo každou epochu a na konci trénování byl načten model, který měl tuto hodnotu nejlepší. To může pomoci v případech, pokud by model nejlepších hodnot dosáhl uprostřed trénování a poté by se začal zhoršovat, např. kvůli přetrénování.

```
per_device_train_batch_size=1
gradient_accumulation_steps=16
learning_rate=1e-5
warmup_ratio=0.12
fp16=True
gradient_checkpointing=True
evaluation_strategy="epoch"
save_strategy="epoch"
load_best_model_at_end=True
metric_for_best_model="wer"
num_train_epochs=15
```

Trénování probíhalo vždy přetrénováním (tzv. *fine-tuning)* základního Whisperu Medium na všech dostupných datech v dané iteraci. Tímto způsobem model uvidí data vícekrát, což by mu mohlo pomoci s učením. V první iteraci byl počet epoch s ohledem na data nastaven na 15 a v každé další iteraci byl tento počet se zvětšujícím se objemem dat o 15 epoch zvýšen. Poslední iterace tedy probíhala 45 epoch. Vývoj průměrné chybo-

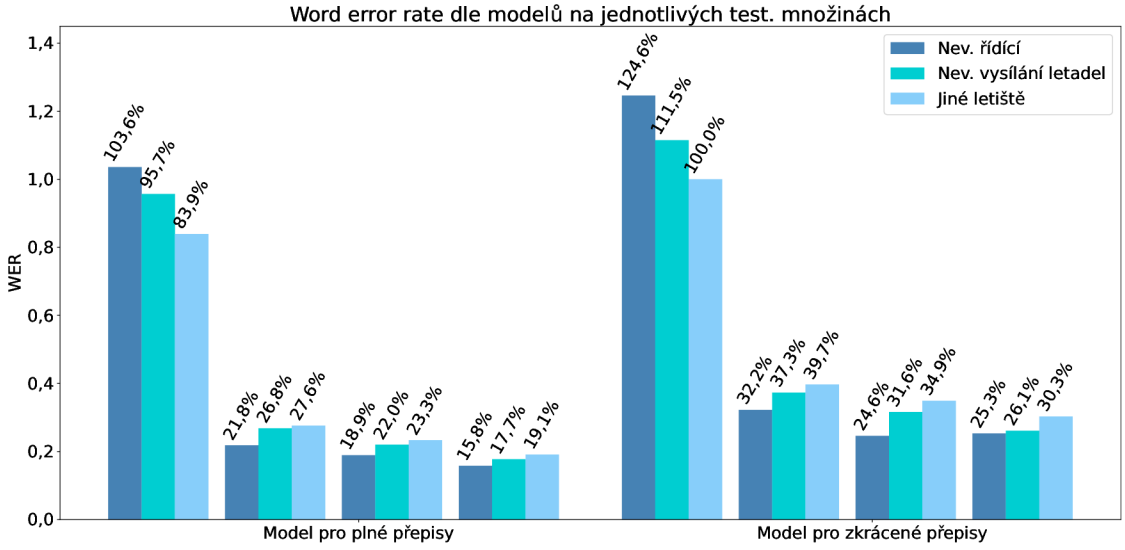
vosti na testovacích množinách LKKU po jednotlivých iteracích byl pro obě formy přepisů následující:



Obrázek 6.1: Průběh hodnot WER na nahrávkách z LKKU v jednotlivých iteracích.

Jak lze vidět, chybovost se po jednotlivých iteracích postupně snižovala. Po první iteraci byl pokles nejrazantnější. Rozdíl mezi posledními iteracemi už není tak velký, ačkoli pořád zaznamenal model pro plné přepisy relativní zlepšení 17,5 % a model pro krátké přepisy 5,5 %.

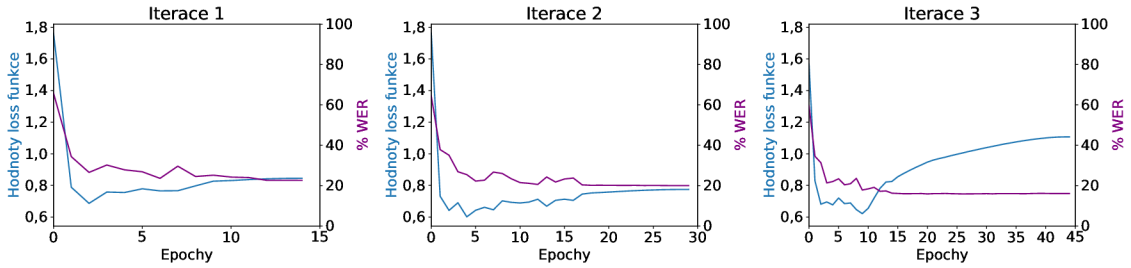
Konkrétní hodnoty WER iterací pro testovací množiny z LKKU navíc i s přidanou test. množinou z jiného letiště jsou zobrazeny na obrázku 6.2.



Obrázek 6.2: Průběh hodnot WER na hlavních testovacích množinách (LKKU) i na množině z jiných letišť pro plné i zkrácené přepisy po jednotlivých iteracích.

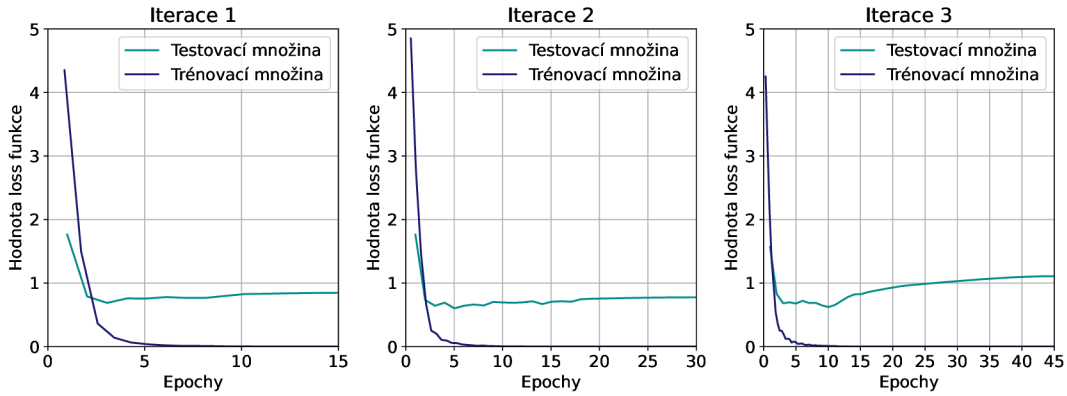
Výsledky na jednotlivých testovacích množinách se odráží od toho, jaké zastoupení měly dané typy nahrávek v trénovací množině. Frazeologii řídících z Kunovic viděl při učení model nejčastěji, proto je na ní model nejpřesnější i na neviděných řídících. Naopak nejhůře si model vede na testovací množině z jiného letiště, která není zastoupena v trénovací ani testovací množině učení. I tak se ale přesnost modelu na množině z jiného letiště po každé iteraci o něco zlepšila, což znamená, že se model zatím neadaptuje příliš na kunovická data a dokáže do velké míry generalizovat.

Tato chybovost je sama o sobě pro objem dat, na kterých byl model trénován, poměrně dobrá a už nyní v ní Whisper výrazně předběhl model z FIT. Během všech trénování se ale nepříznivě vyvíjely hodnoty loss funkce na testovací množině, která vždy ze začátku prudce klesla, po pár epochách však začala narůstat (obr. 6.3). S tímto vývojem ale už nekorelovaly hodnoty WER. Ty také ze začátku prudce klesaly, jejich pokles však na rozdíl od hodnot loss funkcí pokračoval i dále během trénování, kdy pak zkonvergoval k jedné hodnotě. Tento jev je nejvíce viditelný ve třetí iteraci, kdy se sice hodnoty jeho loss funkce na testovací množině s průběhem učení začnou výrazně zvyšovat, hodnoty WER však po celé učení klesají a jsou nejnižší ze všech tří iterací. Toto může být problém použití cross entropy aktivační funkce pro modely rozpoznávání řeči zmíněný v kapitole 3.



Obrázek 6.3: Průběh hodnot loss funkce a WER na testovací množině během učení modelu pro plné přepisy v jednotlivých iteracích.

Co se týče hodnot loss funkce na trénovací množině, ta se vždy po celou dobu trénování snižovala. Porovnání hodnot loss funkcí pro trénovací a testovací množiny lze vidět na obrázku 6.4.



Obrázek 6.4: Průběh hodnot loss funkce na testovací a trénovací množině během učení modelu pro plné přepisy v jednotlivých iteracích.

Oba modely, pro plné přepisy i pro ty zkrácené, měly při evaluaci ze všech tří testovacích množin nejlepší výsledky na neviděných řídících. V češtině na této testovací množině model pro plné přepisy dosáhl 16,0 % a v angličtině dokonce 14,9 % i přesto, že anglických nahrávek bylo při trénování mnohem méně. O něco hůře mu anglický jazyk šel na testovací množině neviděných letadel. Na rozdíl od češtiny, kde zůstala prakticky stejná chybovost 16,1 %, v angličtině došlo ke zhoršení na 20,8 % (tab. 6.3). Nejhorší výkon na českých nahrávkách měl model pro plné přepisy očekávaně na množině z jiného letiště, kde dosáhl chybovosti 19,1 %.

Co se týče modelu pro zkrácené přepisy, ten měl na množině neviděných řídících chybovost v češtině 26,0 %, v angličtině pak 22,2 %. Neviděná vysílání letadel mu šla v češtině kupodivu poměrně dobře, v angličtině chybovost však byla daleko horší. Na českých nahrávkách z jiného letiště pak měl chybovost 30,3 % (tab. 6.3).

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Plné přepisy					
Čeština	16,0 %	16,1 %	16,0 %	**16,5 %**	19,1 %
Angličtina	14,9 %	20,8 %	16,9 %		-
Zkrácené přepisy					
Čeština	26,0 %	22,7 %	24,8 %	**25,6 %**	30,3 %
Angličtina	22,2 %	36,8 %	27,1 %		-

Tabulka 6.3: WER základního Whisperu na testovacích množinách pro plné a zkrácené přepisy.

Dosažená chybovost obou modelů je sama o sobě poměrně dobrá. Oba modely po trénování bez problému, na rozdíl od původního modelu, rozeznávají jazyk, kterým se v nahrávce mluví. Také se vytratilo zacyklení a generování stejných slov pořád dokola. Hyperparametry však při tomto prvotním trénování byly nastaveny „naslepo". Proto další otázkou bylo, zda by bylo možné výkon Whisperu o něco vylepšit jejich jiným nastavením.

Všechny detailní výsledky trénování v následujících sekcích jsou k dispozici na jednom místě v příloze A.

6.2 Experimenty s učením na plné formě přepisů

Jak bylo zmíněno na závěr kapitoly 3, experimentování s hyperparametry lze provádět různými způsoby. Vzhledem k tomu, že se ale Whisper trénuje vždy několik hodin a různých hyperparametrů je nespočet, strategie byla zvolena následovně: v každém experimentu bude model učen na stejných datech – na všech dostupných po anotacích, tedy zhruba 1500 nahrávek. V každém experimentu také bude měněn pouze jeden hyperparametr. Následně bude úspěšnost natrénovaného modelu vyhodnocena a pokud bude mít lepší hodnoty než aktuální nejlepší model, bude se v dalších trénováních pracovat s touto hodnotou daného hyperparametru.

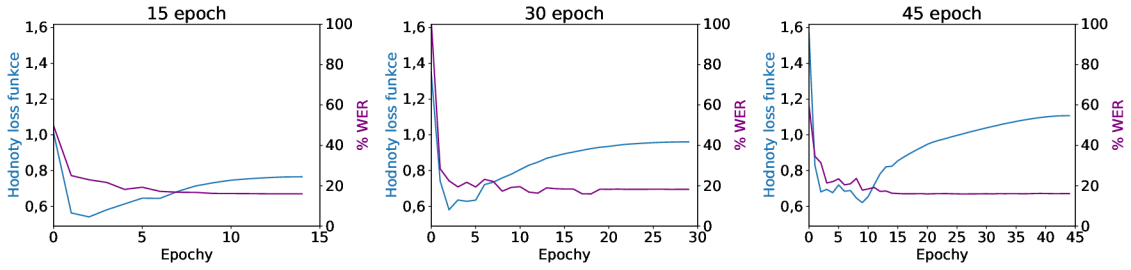
I přesto, že nastavení hyperparametrů ovlivňuje učení nejvíce, drobné rozdíly mezi chybovostí se běžně děly i při dvou učeních na stejných trénovacích datech a parametrech, což ztěžovalo určení, které hyperparametry mají opravdový vliv. Pro lepší představu o vlivu nastavení hyperparametrů proto bylo provedeno 5 stejných trénování modelu pro plné pře-

pisy s výchozím nastavením hyperparametrů s cílem zjistit, jak velké odchylky dělá sám model. Výsledky všech trénování pak byly vyhodnoceny a relativní rozdíl mezi nejlepším a nejhorším modelem činil 4,6 %. Rozdíly WER výrazně větší než tato hodnota pak proto budou moci být považovány za způsobené danou změnou hodnoty hyperparametru.

6.2.1 Experiment 1 – epochy

Rostoucí hodnoty loss funkce během učení po dobu 45 epoch sice mohou být způsobeny nevhodnou volbou cross entropy funkce pro úlohy rozpoznávání řeči, ale také tím, že se model příliš adaptuje na trénovací data. To se při trénování zatím neprokázalo, protože hodnoty WER na testovací množině při učení nijak nenarůstaly. Také se s rostoucím počtem epoch při iteracích nezvyšovaly ani hodnoty WER na testovací množině z jiného letiště, která vůbec při trénování použita nebyla ani v testovací množině. I přesto je ale možné, že pokud bychom snížili počet epoch učení, měl by model podobnou hodnotu WER a svými odpověďmi by si navíc byl více jistý. Proto byla jako první experiment vybrána právě změna počtu epoch.

Byla spuštěna dvě trénování – 15 epoch a 30 epoch. Jejich průběh a porovnání s původními 45 epochami lze vidět na obr. 6.5. Růst hodnot loss funkce se změnou počtu epoch nevyřešil, vždy trénování skončilo s hodnotami většími o několik desítek procent vyššími než jejich minimální dosažené hodnoty. Rozdíly WER při různém počtu epoch také nebyly příliš rozdílné. Po vyhodnocení evaluačním skriptem na neviděných řídících a vysíláních letadel z Kunovic stále vedl model trénovaný 45 epoch s průměrným WER 16,5 %. Model s 30 epochami totiž dosáhl 16,9 % a model s 15 epochami 17,0 %. [1]

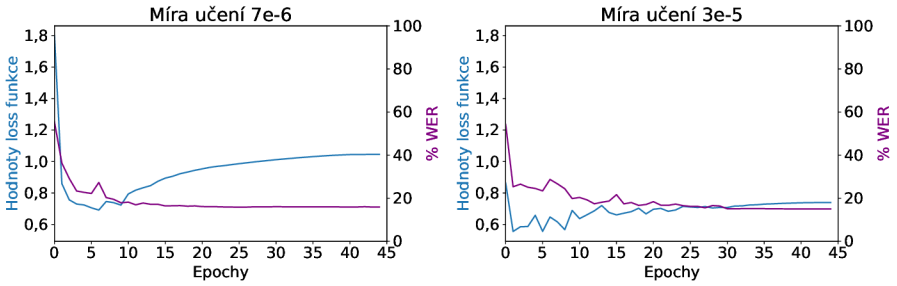


Obrázek 6.5: Průběh hodnot loss funkce a WER na testovací množině během učení modelu na různém počtu epoch.

6.2.2 Experiment 2 – míra učení

Dalším hyperparametrem, který by mohl ovlivnit přesnost modelu, byla míra učení. Výchozí hodnota byla $1 \cdot 10^{-5}$. Trénování byly spuštěny dvě, jedno s menší hodnotou, $7 \cdot 10^{-6}$, a druhé s větší, $3 \cdot 10^{-5}$ – model při předešlých učeních neprojevoval na hodnotách WER známky přeučení, proto byla míra učení prvního trénování snížena pouze o kousek, zatímco u druhého byla míra zvýšena více. Průběh hodnot loss funkce a WER na testovací množině znázorňují grafy v obr. 6.6.

[1] Na obrázku je průběh WER na testovací množině během trénování. I přesto, že je testovací množina během trénování stejná jako ta v evaluačním skriptu, hodnoty WER se z neznámého důvodu vždy o něco málo lišily. Proto je z důvodu jednotnosti pro přesné vyhodnocení WER použit vždy pouze evaluační skript.



Obrázek 6.6: Průběh hodnot loss funkce a WER na testovací množině při trénování s mírou učení $7 \cdot 10^{-6}$ a $3 \cdot 10^{-5}$.

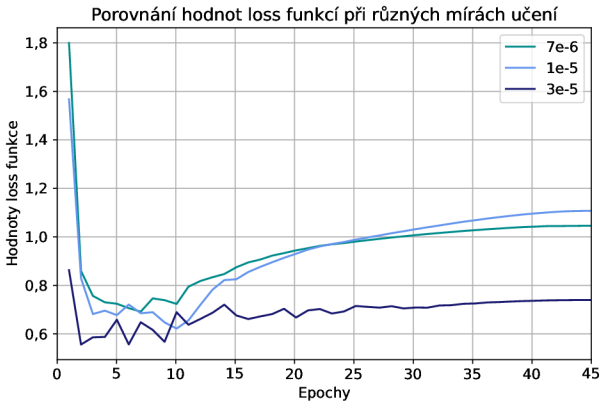
Model s mírou učení $7 \cdot 10^{-6}$ dopadl hůře než model s výchozím nastavením hyperparametrů, jeho průměrná hodnota WER na neviděných řídících a letadlech byla 16,6 %. Zvýšení míry učení u druhého modelu se však vyplatilo, protože nakonec dopadl lépe. Podařilo se mu dosáhnout průměrného WER 14,9 %, čímž se oproti modelu s výchozí mírou učení $1 \cdot 10^{-5}$ relativně zlepšil o více než 9 %. To je zhruba dvakrát tolik, než byla naměřena odchylka několika trénování při stejných parametrech. Toto nastavení se tedy prokázalo jako účinné a v dalších trénováních bude proto hodnota míry učení nastavena na $3 \cdot 10^{-5}$.

Jediná množina, na které se tento model lehce zhoršil, byla neviděná anglická vysílání letadel. Detailní hodnoty s barevným označením **absolutního** zlepšení/zhoršení oproti modelu s výchozím nastavením hyperparametrů lze vidět v tabulce 6.4.

Jazyk	**Neviděí řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	14,2 % (−1,8%)	14,5 % (−1,6%)	14,3 % (−1,7%)	**14,9 %** (−1,6%)	19,0 % (−0,1%)
Angličtina	13,9 % (−1,0%)	21,1 % (+0,3%)	16,3 % (+0,6%)		-

Tabulka 6.4: WER modelu pro plné přepisy s mírou učení $3 \cdot 10^{-5}$.

Dalším zajímavým výsledkem zvýšení míry učení bylo, že se hodnoty loss funkce mnohem více stabilizovaly a nerostly tak výrazným způsobem. Od svého minima se po zbytek učení nedostaly nad hodnoty 0,8, na rozdíl od modelů s mírou učení $1 \cdot 10^{-5}$ i $7 \cdot 10^{-6}$. Ty se oba se svými hodnotami loss funkce na testovací množině dostaly i nad 1,0 (obr. 6.7).



Obrázek 6.7: Průběh hodnot loss funkcí při trénování s mírou učení $7{\cdot}10^{-6}$, $1{\cdot}10^{-5}$ a $3{\cdot}10^{-5}$.

6.2.3 Experiment 3 – warmup

Pro zajímavost bylo dále vyzkoušeno zvýšení doby warmupu z 12 % na 25 %. To by mohlo o něco více přispět k robustnosti modelu tím, že se jeho váhy budou ze začátku učení upravovat méně a maximální míry učení bude dosaženo až ve čtvrtině trénování.

Tento experiment však nepřinesl lepší výsledky. Jediná podmnožina, na které se model zlepšil, byly anglické nahrávky neviděných řídících a s tím se také zlepšila celková hodnota WER pro anglický jazyk. Na všech ostatních množinách se ale model buď zhoršil, nebo zůstal stejný a celkově se absolutně zhoršil o 0,5 % oproti aktuálně nejlepšímu modelu (viz tab. 6.5).

Jazyk	**Neviděí řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	15,0 % (+0,8%)	14,8 % (+0,3%)	15,0 % (+0,7%)	**15,4 %** (+0,5%)	19,0 % (−0,0%)
Angličtina	13,5 % (−0,4%)	21,1 % (−0,0%)	16,1 % (−0,2%)		-

Tabulka 6.5: WER modelu pro plné přepisy s 25 % warmup.

6.2.4 Experiment 4 – dropout

Další věcí, která byla vyzkoušena, bylo zvýšení dropout pravděpodobnosti. Výchozí hodnota je nastavena na 10 %, pro tento experiment byla zvýšena na 30 % a 50 % dropout vrstvy.

Model s 50 % dopadl hůře, tak velké procento dropoutu bylo nejspíše příliš. Vůči zatím nejlepšímu modelu se zhoršil ve všech třech testovacích množinách – na LKKU datech dosáhl průměrné chybovosti 16,0 %, na nahrávkách z jiného letiště 20,0 %. Zajímavý byl ovšem výsledek trénování s 30 % dropout. Model se sice v chybovosti na LKKU datech zhoršil, na nahrávkách z jiného letiště však dosáhl zatím nejmenší chybovosti, a to 18,3 % (tab 6.6). Je tedy možné, že model díky zavedení dropout vrstev zvýšil svou robustnost.

Jazyk	Nevidění řídící (LKKU)	Neviděná vysílání letadel (LKKU)	WER dle jazyků (LKKU)	WER LKKU celkem	Nahrávky z jiného letiště (LKPR, LKTB)
Čeština	14,5 % (+0,3%)	14,7 % (+0,2%)	14,5 % (+0,2%)	**15,3 %** (+0,4%)	18,3 % (−0,7%)
Angličtina	13,9 % (−0,0%)	23,2 % (+2,1%)	17,0 % (+0,7%)		-

Tabulka 6.6: WER modelu pro plné přepisy s 30 % dropout.

Výsledek na nahrávkách z jiného letiště je sice při použití dropout nejlepší, avšak je to zlepšení pouze v jedné ze tří testovacích množin a celkové WER na Kunovických datech se zhoršilo. Proto tento hyperparametr v dalších trénováních nastaven nebude.

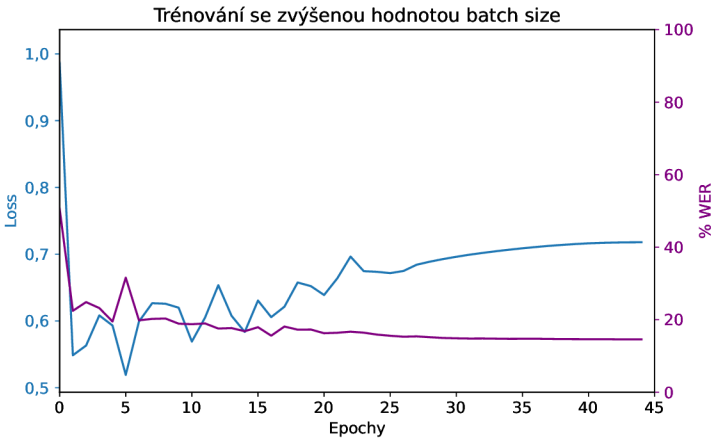
6.2.5 Experiment 5 – batch size

Posledním experimentem v nastavení hyperparametrů byla změna batch size. Původní hodnota byla nastavena na 1, pro tento pokus byla zvýšena na 2. Se zvýšením však bylo potřeba stejným násobkem zmenšit hyperparametr `gradient_accumulation_steps`, proto byl snížen na hodnotu 8. Výsledky trénování jsou rozepsány v tabulce 6.7 a průběh hodnot loss funkce na testovací množině a WER je v obr. 6.8.

Jazyk	Nevidění řídící (LKKU)	Neviděná vysílání letadel (LKKU)	WER dle jazyků (LKKU)	WER LKKU celkem	Nahrávky z jiného letiště (LKPR, LKTB)
Čeština	14,9 % (+0,7%)	13,6 % (−0,9%)	14,5 % (+0,2%)	**14,7 %** (−0,2%)	19,6 % (+0,6%)
Angličtina	12,1 % (−1,8%)	17,9 % (−3,2%)	14,1 % (−2,2%)		-

Tabulka 6.7: WER modelu pro plné přepisy se zvýšeným batch size.

Model s větší nastavenou batch size se zlepšil v angličtině na obou testovacích množinách poměrně výrazně. V češtině se zlepšil pouze na množině neviděných vysílání letadel z Kunovic, na neviděných řídících se o něco zhoršil. Celkově se model oproti doposud nejlepšímu modelu relativně zlepšil o 1,9 %. Jelikož taková míra zlepšení spadá i do rozmezí, které se stávalo při několika trénováních na stejných hyperparametrech, nelze prokazatelně určit, zda bylo toto zlepšení způsobeno změnou hodnoty batch size. Tak či tak se jednalo o poslední změnu základních hyperparametrů a při tomto nastavení bylo dosaženo největší dosavadní přesnosti, proto další pokusy zůstanou s nastavenou touto hodnotou.



Obrázek 6.8: Průběh hodnot loss funkce a WER na testovací množině během učení modelu na zvýšené hodnotě batch size.

6.2.6 Experiment 6 – zamrazení enkodéru

Další věcí, která se dala při učení nastavit, byl tzv. *freeze* (zamrazení) enkodéru. Při aktivaci tohoto nastavení se při trénování upravují pouze váhy dekodéru a enkodér zůstává nezměněn. To může mít pozitivní dopad na generovaný text a snížit jeho chybovost. Před zamrazením je ale ideální, aby byl enkodér na audio vstupy již adaptován a dokázal s nimi pracovat. Vzhledem k tomu, že výchozí model Whisper nebyl trénován na nahrávkách letecké komunikace, bylo proto nejlepší model přetrénovat při zapnutém učení enkodéru i dekodéru na menší množině dat, aby enkodér upravil své váhy, a poté na zbytku dat tento přetrénovaný model ještě jednou naučit, tentokrát již se zamrazeným enkodérem.

Přetrénování proběhlo na cca třetině dat, tedy zhruba 550 nahrávkách. Hyperparametry byly nastaveny stejně jako na nejlepším dosaženém modelu pro plnou formu přepisů, jen počet epoch byl změněn na 15. Po trénování měl model následující chybovost:

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	22,0 %	25,3 %	23,1 %	**23,1 %**	25,9 %
Angličtina	19,7 %	25,9 %	21,8 %		-

Tabulka 6.8: WER modelu pro plné přepisy po 15 epochách trénování na třetině nahrávek.

Tento model byl poté přetrénován na zbytku (zhruba 1000) nahrávek se stejnými parametry, pouze se změnou počtu epoch, kterých bylo 30, a se zamrazeným enkodérem. Trénování však nepřineslo lepší výsledky než má aktuálně nejlepší model. Model se zamrazeným enkodérem dosáhl horších výsledků ve všech testovacích množinách (viz tabulka 6.9).

Jazyk	Nevidění řídící (LKKU)	Neviděná vysílání letadel (LKKU)	WER dle jazyků (LKKU)	WER LKKU celkem	Nahrávky z jiného letiště (LKPR, LKTB)
Čeština	17,7 % (+2,8%)	18,5 % (+4,9%)	18,0 % (+3,5%)	**18,6 %** (+3,9%)	20,5 % (+0,9%)
Angličtina	16,3 % (+4,2%)	25,9 % (+8,0%)	19,5 % (+5,4%)		-

Tabulka 6.9: WER modelu pro plné přepisy po zamrazení enkodéru.

6.2.7 Experiment 7 – přidání anglických dat

Nejlepší natrénovaný model má již teď poměrně vysokou přesnost i na anglických nahrávkách. Anglických nahrávek je však v trénovacím datasetu menšina a většina je v jazyku českém. Zajímavý pokus by tedy byl zvýšit počet anglických nahrávek, zkusit na nich model přetrénovat a vyhodnotit, jaký dopad to bude mít na anglické a české nahrávky.

K tomuto účelu byla vedoucím poskytnuta ATCO2 data. Ta obsahovala anglické nahrávky z mnoha různých zemí s již vytvořenými přepisy. Protože nahrávky ze zahraničí měly různé úrovně šumu a angličtina v nich byla s různými přízvuky, pro začátek byly vytvořeny datasety pouze z anglických nahrávek pocházejících z Česka. Tato angličtina je totiž nejvíce podobná té, na které byl model učen z kunovických nahrávek. Pokud by se angličtina po natrénování na těchto nahrávkách zlepšila, mohlo by být do trénování zařazeno více dat i z jiných zemí. Je ovšem nutné dodat, že i přesto, že jsou data nejvíce podobná těm, které model již při učení viděl, obsah nahrávek není úplně stejný. Občas se vyskytují volací znaky linkových letadel (Lufthansa, Ryanair apod.) či pokyny, které se v Kunovicích prakticky nepoužívají.

Nejlepší doposud natrénovaný model byl na těchto zhruba 100 nahrávkách anglické komunikace pocházející z Česka přeučen s nejlepším nastavením hyperparametrů a s počtem 7 epoch. Trénování však nedopadlo pozitivně. Model se na anglických nahrávkách místo očekávaného zlepšení zhoršil. Mnohem větší zhoršení ale potkalo české nahrávky, kde na testovacích množinách z LKKU dosáhl 57,0 % a na nahrávkách z jiného letiště dokonce 84,9 % (tab. 6.10).

Jazyk	Nevidění řídící (LKKU)	Neviděná vysílání letadel (LKKU)	WER dle jazyků (LKKU)	WER LKKU celkem	Nahrávky z jiného letiště (LKPR, LKTB)
Čeština	66,1 % (+51,2%)	40,4 % (+26,8%)	57,0 % (+42,5%)	**49,9 %** (+35,2%)	84,9 % (+65,3%)
Angličtina	19,2 % (+7,1%)	19,8 % (+1,9%)	19,4 % (+5,3%)		-

Tabulka 6.10: WER nejlepšího modelu přetrénovaného na vytvořeném ATCO2 datasetu.

U tohoto modelu se také opět objevily halucinace či přepis českých slov anglicky, což se původně v nejlepším modelu již vůbec nevyskytovalo. Doposud měl tyto vlastnosti pouze výchozí Whisper model a přetrénováním se je vždy podařilo eliminovat. Zjevně se tedy tyto jevy objevují, pokud je model zmatený a dostává k přepisu typy nahrávek, na kterých

nebyl trénován, či na kterých byl trénován v dřívějších iteracích trénování a tím pádem už na jejich správné zpracování do určité míry „zapomněl".

To, že model zhoršil svůj výkon na českých nahrávkách, když byl při posledním učení trénován pouze na těch anglických, je očekávatelné. Řešení, které by tomuto mohlo pomoci, je přetrénovat výchozí model Whisper na nejlepším nastavení hyperparametrů na všech původních datasetech, ke kterým budou připojeny i právě tyto ATCO2 anglické nahrávky. Tím pádem se model bude učit češtinu i angličtinu zároveň a díky většímu zastoupení angličtiny by se v ní mohl o něco zlepšit.

Trénování bylo ponecháno jako u nejlepšího modelu na 45 epoch, vzhledem k tomu, že ATCO2 dat je v trénování malé množství a nijak výrazně se tím velikost trénovacího datasetu nezvětšila. Výsledky dopadly následovně:

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	16,5 % (+1,6%)	14,2 % (+0,6%)	15,7 % (+1,2%)	**16,0 %** (+1,3%)	20,8 % (+1,2%)
Angličtina	14,2 % (+2,1%)	19,8 % (+1,9%)	16,1 % (+2,0%)		-

Tabulka 6.11: WER modelu po 45 epochách trénování na kunovických datasetech i vytvořeném ATCO2 datasetu.

Zlepšení oproti přetrénovanému modelu pouze na anglických nahrávkách je razantní. Každopádně vůči nejlepšímu modelu je tento model horší jak v celkovém WER, tak i ve všech testovacích množinách. Tedy ani tato strategie nepomohla modelu lépe rozpoznávat angličtinu z testovacích množin.

6.3 Experimenty s učením na zkrácené formě přepisů

Zkrácené přepisy byly nejprve přeučeny na stejném nastavení hyperparametrů, při kterých bylo dosaženo největší přesnosti modelu pro plné přepisy. Nastavení tedy bylo následující:

```
per_device_train_batch_size=2
gradient_accumulation_steps=8
learning_rate=3e-5
warmup_ratio=0.12
fp16=True
gradient_checkpointing=True
evaluation_strategy="epoch"
save_strategy="epoch"
load_best_model_at_end=True
metric_for_best_model="wer"
num_train_epochs=45
```

Trénování na těchto hyperparametrech dopadlo poměrně různorodě pro každou testovací množinu (viz tab. 6.12). Model se výrazně zlepšil na českých nahrávkách neviděných řídících, kde dosáhl na rozdíl od původního modelu (tab. 6.3) 18,0 % místo původních 26,0 %.

Zhoršení nastalo u českých nahrávek z jiného letiště a u neviděných anglických vysílání letadel. Celkové WER má však model lepší než model původní, proto je možné ho považovat za aktuálně nejlepší.

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	18,0 % (−8,0%)	18,4 % (−4,3%)	18,1 % (−6,7%)	**20,0 %** (−5,6%)	34,0 % (+3,7%)
Angličtina	21,2 % (−1,0%)	41,0 % (+4,2%)	27,9 % (+0,8%)		-

Tabulka 6.12: WER modelu pro zkrácené přepisy trénovaného na nejlepších hyperparametrech modelu pro plné přepisy.

Přeučení modelu pro zkrácené přepisy se stejnými hyperparametry, které byly účinné u modelu pro plné přepisy, ale nemusí znamenat automaticky nejlepší výsledek. Vzhledem k tomu, že zkrácené přepisy jsou značně složitější než plné, může pro model platit jiné nastavení hyperparametrů. Proto bylo vytvořeno také pár experimentů trénování. Tentokrát byly ale vybrány ty, které prokázaly, že mají výraznější dopad na trénování – tedy epochy, míra učení a batch size. Výsledky těchto experimentů budou porovnávány, stejně jako experimenty na plných přepisech, vůči původnímu přetrénovanému modelu na 45 epochách se „slepým" nastavením hyperparametrů a následně bude nejlepší model z těchto experimentů porovnán s modelem trénovaným na nejlepších parametrech pro plné přepisy. Tím bude nakonec určen model s nejmenší chybovostí.

6.3.1 Experiment 1 – epochy

Stejně jako u modelů pro plnou formu přepisů byl jako první pro úpravu zvolen počet epoch. Trénování probíhalo na 15 a 30 epochách a bylo porovnáváno s původním modelem trénovaným na 45 epochách.

Výsledky dopadly jinak než u modelu pro plné přepisy. I přesto, že zkrácené přepisy mohou být pro model těžší a tím pádem by bylo možná nutno předložit data modelu vícekrát, aby se naučil vzory v datech, nejlépe dopadl model trénovaný pouze na 15 epochách:

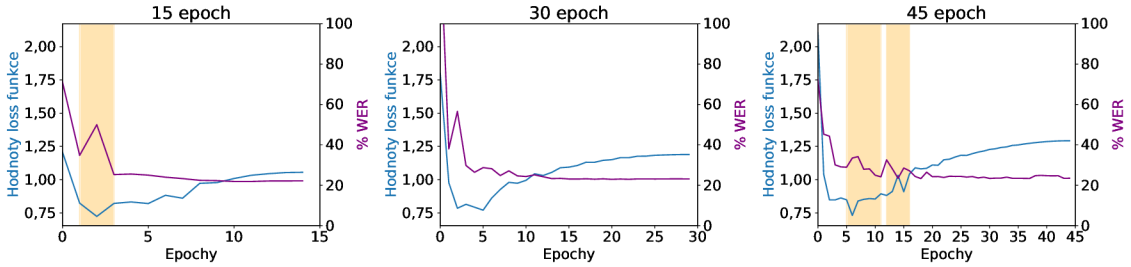
Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	21,4 % (−4,6%)	21,7 % (−1,0%)	21,5 % (−3,3%)	**22,7 %** (−2,9%)	28,2 % (−2,1%)
Angličtina	22,2 % (−0,0%)	38,1 % (+1,3%)	27,5 % (+0,4%)		-

Tabulka 6.13: WER modelu pro zkrácené přepisy trénovaného 15 epoch.

Model se zlepšil oproti původnímu modelu trénovaném na základním „slepém" nastavení hyperparametrů ve všech českých podmnožinách. U anglických nahrávek neviděných řídících zůstal model stejný a u neviděných vysílání letadel se na angličtině zhoršil. I tak

je ale celkový výsledek modelu relativně lepší o více než 11 % než původní, proto pro další experimenty bude voleno trénování na 15 epochách.

Co se týče vývoje hodnot loss funkce na testovací množině během učení, ty měly stejně jako u modelu pro plné přepisy stoupající tendenci. V některých místech (zvýrazněna oranžově) grafů v obr. 6.9 to pak dokonce vypadá, že místo toho, aby se jejich hodnoty vyvíjely podobně, jdou téměř proti sobě.



Obrázek 6.9: Průběh hodnot loss funkce na testovací množině a WER při učení modelu pro zkrácené přepisy při různém počtu epoch se zvýrazněním protichůdného vývoje metrik.

6.3.2 Experiment 2 – míra učení

Druhý měněný hyperparametr v pořadí byla míra učení. Původní model byl nastaven na hodnotu $1 \cdot 10^{-5}$. Při tomto experimentu byly hodnoty nastaveny stejně jako u totožného experimentu u modelu pro plné přepisy, tedy $3 \cdot 10^{-5}$ a $7 \cdot 10^{-6}$.

Podobně jako u modelu pro plné přepisy měl lepší výsledky model se zvýšenou mírou učení. Model se lehce zhoršil na neviděných českých vysíláních letadel a záznamech z jiného letiště, na ostatních podmnožinách testovacích množin se ale výrazně zlepšil a poprvé dosáhl chybovosti menší než 20 %, a to dokonce u dvou podmnožin (tab. 6.14). Model zaznamenal relativní zlepšení o více než 7 % – další trénování proto bude probíhat s hodnotou míry učení $3 \cdot 10^{-5}$.

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	19,4 % (−2,0%)	21,9 % (+0,2%)	20,3 % (−1,2%)	**21,0 %** (−1,7%)	29,0 % (+0,8%)
Angličtina	17,8 % (−4,4%)	31,8 % (−6,3%)	22,5 % (−5,0%)		-

Tabulka 6.14: WER modelu pro zkrácené přepisy s mírou učení $3 \cdot 10^{-5}$.

6.3.3 Experiment 3 – batch size

Poslední pokus byl proveden se změněnou hodnotou batch size. Ta byla zvýšena z původní hodnoty 1 na hodnotu 2. Na rozdíl od modelu pro plné přepisy však tato změna měla spíše negativní dopad (tab. 6.15).

Jediná testovací množina množina, ve které se model výrazněji zlepšil, byla neviděná vysílání letadel v českém jazyce. Na záznamech z jiného letiště se model sice taky zlepšil, ale

pouze o 0,1 % absolutně. Celkové WER se zhoršilo oproti nejlepšímu dosaženému modelu z 21,0 % na 21,2 %, a proto změna nastavení batch size nebyla prokázána jako účinná pro model pro zkrácené přepisy.

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	19,9 % (+0,5%)	20,4 % (−1,5%)	20,1 % (−0,2%)	**21,2 %** (+0,2%)	28,9 % (−0,1%)
Angličtina	20,1 % (+2,3%)	35,6 % (+3,8%)	25,3 % (+2,8%)		-

Tabulka 6.15: WER modelu pro zkrácené přepisy po trénování se zvýšenou hodnotou batch size.

6.3.4 Určení nejlepšího modelu pro zkrácené přepisy

Nejlepším modelem pro zkrácené přepisy natrénovaným pomocí postupné změny hyperparametrů se stal model trénovaný na pouhých 15 epochách a se zvýšenou mírou učení. Posledním úkolem bylo porovnat ho s modelem, který byl trénován na nejlepších hyperparametrech modelu pro plné přepisy, viz tab. 6.16 a tab. 6.17, ve které jsou i barevně vyznačeny rozdíly chybovosti oproti modelu na nejlepších hyp. pro plné přepisy.

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	18,0 %	18,4 %	18,1 %	**20,0 %**	34,0 %
Angličtina	21,2 %	41,0 %	27,9 %		-

Tabulka 6.16: WER modelu pro zkrácené přepisy trénovaného na nejlepším nastavení hyperparametrů pro plné přepisy.

Jazyk	**Nevidění řídící** (LKKU)	**Neviděná vysílání letadel** (LKKU)	**WER dle jazyků** (LKKU)	**WER LKKU celkem**	**Nahrávky z jiného letiště** (LKPR, LKTB)
Čeština	19,4 % (+1,4%)	21,9 % (+3,5%)	20,3 % (+2,2%)	**21,0 %** (+1,0%)	29,0 % (−5,0%)
Angličtina	17,8 % (−3,4%)	31,8 % (−9,2%)	22,5 % (−5,4%)		-

Tabulka 6.17: WER nejlepšího modelu pro zkrácené přepisy dosaženého postupnou změnou hyperparametrů.

Jednoznačně nelze nejlepší model určit, jelikož se jejich úspěšnost liší v několika množinách. Model dosažený postupnou změnou hyperparametrů má vyrovnanější výsledky skrze všechny testovací množiny na rozdíl od modelu trénovaného na nejlepším nastavení hyperparametrů pro plné přepisy. Model trénovaný na nejlepších hyperparametrech pro plné

přepisy má ale celkové WER nižší a v obou českých testovacích podmnožinách z Kunovic má chybovost nižší než 20 %, a proto byl za nejlepší model zvolen právě ten.

6.4 Vyhodnocení modelů

Chybovost obou nejlepších modelů na různých formách přepisů je poměrně nízká, i když model pro plné přepisy v ní oproti modelu pro zkrácené přepisy zcela jasně vede. Hodnocení modelů pouze dle hodnoty WER je ale příliš povrchní. Aby bylo možné modely nadále zlepšovat, je třeba se podívat na jednotlivé nahrávky a jejich přepisy a identifikovat, které věci modelům dělají největší problémy.

Oba modely si často nevěděly rady se slovy, která se ve vysíláních často nevyskytují. To je pochopitelné, protože model nemá šanci se pořádně jemné nuance ve výslovnostech daných slov naučit, pokud je neviděl několikrát řečena od různých řečníků. Obecně by se dala špatně identifikovaná slova rozřadit do tří hlavních skupin:

- **Názvy obcí a geografických míst** – tyto názvy jsou často používány piloty při hlášení aktuální polohy. Tento typ chyb byl nejčastější v testovací množině z jiného letiště – na testovacích množinách z Kunovic se do určité míry některá hlavní místa naučil. I přesto, že názvy model nikdy neviděl, se ale snažil a většinou se přepis alespoň do určité míry podobal správnému tvaru:

  `Bohunicích` → `bouřicích`
  `Tuřany` → `Tuři`
  `Medlánky` → `Medlán k`
  `Čebín` → `čebym`

- **Typy letounů** – při prvním hlášení letadla věži se obvykle sděluje typ daného letounu. S často vyskytujícími se letadly (např. Cessna, Piper) modely většinou problém neměly, větší problém však dělaly méně časté typy letadel:

  `Zephyr` → `vzletí`
  `Roko` → `Echo`

- **Volná řeč** – i přesto, že by se letecká komunikace měla udržovat pokud možno v ustálených frázích, pokud pro dané sdělení či požadavek žádná fráze neexistuje, přechází se do volné řeči. Ustálené fráze, které se v nahrávkách opakují často, se modelům dobře učí, ale sdělení ve volné řeči je pokaždé jiné a proto s tím modely mají větší problém.

Věci, které modelům naopak šly, byla čísla a písmena hláskovací abecedy. Ta se totiž vyskytují téměř v každé nahrávce. Dle přepisů nahrávek testovacích množin to vypadalo, že se modelům především na volacích znacích letadel velmi daří. Proto byl vytvořen poslední evaluační skript, který počítá jejich WER. Vzhledem k tomu, že nelze jednoduchým algoritmem z přepisů extrahovat pouze volací znaky (hláskovací abeceda se totiž nepoužívá pouze u volacích znaků letadel, ale i při identifikacích letišť, výstupních a vstupních bodech vzdušných prostorů apod.), evaluace musí být z části manuální. Skript načítá jako vstup soubor ve tvaru výstupu evaluačního skriptu pro testovací množiny. Následně předkládá uživateli referenční a vygenerovaný přepis nahrávky a uživatel zadává hodnoty, které WER sleduje – počet nahrazených, vložených, vymazaných a správných slov u volacích znaků. Hodnoty si skript ukládá a po průchodu celým souborem spočítá výslednou hodnotu WER.

Chybovost na volacích znacích byla vyhodnocena pro všechny testovací množiny. Vzhledem k tomu, že hláskovací abeceda je stejná jak v českém, tak v anglickém jazyce, bylo WER spočítáno pro všechny nahrávky každé testovací množiny dohromady a nedělilo se na jazyky. Výsledky pro model pro plné přepisy dopadly nad očekávání dobře:

Testovací množina	Nevidění řídící	Neviděná letadla	Záznamy z jiného letiště
WER vol. znaků	7,3 %	4,2 %	3,6 %

Tabulka 6.18: WER nejlepšího modelu pro plné přepisy na volacích znacích.

Překvapivě nejlepší výsledky měl model na nahrávkách z jiného letiště, kde dosáhl chybovosti pouhých 3,6 %. Na neviděných vysíláních letadel byla chybovost o něco vyšší a nejvyšší chybovost 7,3 % měla testovací množina neviděných řídících.

Vzhledem k tomu, že zkrácený přepis volacího znaku je většinou jedno slovo složené kombinací několika písmen/čísel, chybovost modelu pro zkrácené přepisy byla vyhodnocena po jednotlivých znacích. Ta dopadla následovně:

Testovací množina	Nevidění řídící	Neviděná letadla	Záznamy z jiného letiště
WER vol. znaků	10,2 %	8,0 %	11,6 %

Tabulka 6.19: WER nejlepšího modelu pro zkrácené přepisy na volacích znacích.

I přesto, že je chybovost modelu pro zkrácené přepisy na volacích znacích daleko lepší než průměrná chybovost na jednotlivých testovacích množinách, je to stále průměrně každé desáté písmeno či číslo vygenerované špatně. Pokud má volací znak letadel většinou pět znaků, znamená to, že model udělá alespoň jednu chybu v každém druhém volacím znaku.

Při vyhodnocování chybovosti modelu pro zkrácené přepisy na volacích znacích bylo vidět, že model nemá přímo nějaký přesný vzor dělání chyb. Nejmenší chybovost měla samozřejmě letadla z Kunovic, která létají často a jejichž volací znaky model při trénování tedy více vídal. Každopádně i v nich model občas dělal chyby. V testovací množině z jiného letiště, u které bylo očekáváno, že chybovost dopadne mnohem hůře, jelikož model dané volací znaky letadel nikdy neviděl, ale model překvapil. Spoustu volacích znaků určoval bez chyb. Ty chyby, které ale dělal, byly různorodé. Stávalo se zaměnění dvou písmen ve volacím znaku, vynechaná písmena i písmena navíc. U záznamů z jiného letiště mu některé volací znaky přišly nejspíše povědomé a vygeneroval volací znak letadla z Kunovic, které se v nahrávkách vyskytovalo často. I tak byla ale průměrná chybovost na volacích znacích mnohem menší než bylo očekáváno a model si na nich vedl poměrně dobře.

Co se týče vyzkoušení rozdělování přepisů jednotlivých vysílání v nahrávkách na samostatné řádky, s tím model také žádný problém neměl. Ve všech nahrávkách předložených modelu pro zajímavost, které obsahovaly více vysílání, model vždy jednotlivé přepisy rozděloval. Na základě toho byly vyzkoušeny i nahrávky vytvořené uměle, ve kterých se střídali dva řečníci (muž a žena). Cílem bylo zjistit, zda model vysílání rozděluje dle zvuku způsobeného zahájením a ukončením vysílání přes radiový kanál, či na základě změny hlasu. Uměle vytvořené nahrávky totiž byly nahrány běžným mikrofonem, a tak byla změna řečníka provedena bez doprovodného zvuku v nahrávce.

Ani s tímto však model problém neměl, vždy zvládal rozlišovat mezi různými řečníky a přepis jejich vět umisťoval vždy na samostatné řádky. Whisper se tedy ve všech aspektech přepisu, které mu byly zadány, prokázal jako velmi všestranný model zvládající několik složitějších úkolů zároveň.

Kapitola 7

Závěr

Hlavním zájmem této bakalářské práce bylo učení modelu automatického rozpoznávání řeči Whisper na českých a anglických nahrávkách letecké komunikace s cílem zjištění eventuální použitelnosti v praxi řídícími letového provozu či při vyhledávání v záznamech letecké komunikace. Za tímto účelem se podařilo vytvořit dva modely pro různé formy přepisů, plné a zkrácené. Model pro plné přepisy již během prvotního trénování dosáhl poměrně dobrých výsledků. Po experimentech s hyperparametry se ale podařilo jeho chybovost ještě snížit. Ta poté dosáhla průměrné hodnoty 14,7 %. Jeho nejsilnější stránkou byla chybovost na volacích znacích letadel, která dosáhla na množině nahrávek z jiného letiště pouhých 3,6 %

Model pro zkrácené přepisy měl vzhledem k požadovanému výstupu obtížnější podmínky pro učení. I tak ale dokázal dosáhnout průměrné chybovosti 20,0 %. S chybovostí na volacích znacích se pohyboval okolo 8 až 12 % pro všechny testovací množiny.

Celková přesnost modelu pro plné přepisy je poměrně vysoká. Pro potenciální použití na řídící věži je však model stále nevhodný, jelikož i průměrná chybovost 14,7 % je pro tak zodpovědný úkol jako přepis letecké komunikace, na který by se mohli řídící letového provozu spolehnout, stále příliš vysoká. Pro zpětné vyhledávání v přepisech záznamů letecké komunikace by se však dal model použít již teď. V tomto případě totiž nejde o tak rizikovou činnost. Navíc by se s největší pravděpodobností nejčastěji vyhledávalo pomocí volacích znaků letadel, na čemž model prokázal vysokou přesnost.

Nevýhodou modelu pro plné přepisy je ale délka přepisů v porovnání s jejich informační hodnotou. Řešením by mohlo být např. automatické zpracování plných přepisů. Pokud by se podařilo vyvinout systém, který by dokázat s dostatečnou přesností detekovat, která slova mají být zkrácená a spojená v jedno slovo a která nikoliv, velmi by to přepisy zpřehlednilo a zároveň by byla zachována přesnost modelu pro plné přepisy.

I přesto, že trénovací objem dat, na kterých byly modely trénovány, byl velmi malý, Whisper prokázal schopnost adaptovat se rychle na jednodušší (plný), ale i složitější (zkrácený) tvar přepisu. S větším objemem dat i např. z jiných českých letišť by proto mohlo být reálné nadále snižovat chybovost obou modelů a dosáhnout tak nízké chybovosti, kdy by byly modely velmi spolehlivé pro přepisy jakékoli české i anglické letecké komunikace.

Práce byla prezentována na studentské konferenci Excel@FIT[1] a nejlepší modely pro plné i zkrácené přepisy byly publikovány na Hugging Face[2,3], kde si je může kdokoli stáhnout a používat pro přepisy nahrávek letecké komunikace nebo pokračovat ve vylepšování.

[1] https://excel.fit.vutbr.cz/
[2] https://huggingface.co/BUT-FIT/whisper-ATC-czech-full
[3] https://huggingface.co/BUT-FIT/whisper-ATC-czech-short

Literatura

[1] Allamy, H. K. and Khan, R. Z.. Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study). [online]. Prosinec 2014, [cit. 2. 4. 2024]. Dostupné z: https://www.researchgate.net/publication/295198699_METHODS_TO_AVOID_OVER-FITTING_AND_UNDER-FITTING_IN_SUPERVISED_MACHINE_LEARNING_COMPARATIVE_STUDY.

[2] Banerjee, K., C., V. P., Gupta, R. R., Vyas, K., H., A. et al. *Exploring Alternatives to Softmax Function* [online]. 2020 [cit. 16. 4. 2024]. DOI: arXiv:2011.11538. Dostupné z: https://arxiv.org/abs/2011.11538.

[3] Brownlee, J. What is the Difference Between a Batch and an Epoch in a Neural Network? *Deep learning*. Červenec 2018, [cit. 16. 4. 2024]. Dostupné z: https://deeplearning.lipingyang.org/wp-content/uploads/2018/07/What-is-the-Difference-Between-a-Batch-and-an-Epoch-in-a-Neural-Network_.pdf.

[4] Dahl, G. E., Yu, D., Deng, L. a Acero, A. Large vocabulary continuous speech recognition with context-dependent DBN-HMMS. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, s. 4688–4691 [cit. 18. 4. 2024]. DOI: 10.1109/ICASSP.2011.5947401. Dostupné z: https://ieeexplore.ieee.org/document/5947401.

[5] Davis, K. H., R., B. a Balashek, S. Automatic Recognition of Spoken Digits. *The Journal of the Acoustical Society of America*. Listopad 1952, sv. 24, č. 6, s. 637–642, [cit. 18. 4. 2024]. ISSN 0001-4966. Dostupné z: https://rauterberg.employee.id.tue.nl/presentations/bell-labs.pdf.

[6] Gotmare, A., Keskar, N. S., Xiong, C. a Socher, R. *A Closer Look at Deep Learning Heuristics: Learning rate restarts, Warmup and Distillation* [online]. [cit. 17. 4. 2024]. DOI: arXiv:1810.13243. Dostupné z: https://arxiv.org/abs/1810.13243.

[7] Heckbert, P. Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. *Computer Graphics* [online]. 1995, sv. 2, č. 1995, [cit. 20. 4. 2024]. Dostupné z: https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/99/pub/www/notes/fourier/fourier.pdf.

[8] Hugging Face. *Metric:wer* [online]. [cit. 28. 12. 2023]. Dostupné z: https://huggingface.co/spaces/evaluate-metric/wer.

[9] Janocha, K. a Czarnecki, W. M. On Loss Functions for Deep Neural Networks in Classification. *CoRR* [online]. 2017, abs/1702.05659, [cit. 16. 4. 2024]. DOI: arXiv:1702.05659. Dostupné z: https://arxiv.org/abs/1702.05659.

[10] Jordan, M. I. a Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science*. 2015, sv. 349, č. 6245, s. 255–260, [cit. 20. 4. 2024]. DOI: 10.1126/science.aaa8415. Dostupné z: https://www.science.org/doi/abs/10.1126/science.aaa8415.

[11] Kocour, M., Umesh, J., Karafiat, M., Švec, J., López, F. et al. BCN2BRNO: ASR System Fusion for Albayzin 2022 Speech to Text Challenge. In: *Proc. IberSPEECH 2022*. 2022, s. 276–280 [cit. 7. 5. 2024]. DOI: 10.21437/IberSPEECH.2022-56.

[12] Kudo, T. a Richardson, J. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing* [online]. 2018 [cit. 22. 4. 2024]. DOI: arXiv:1808.06226. Dostupné z: https://arxiv.org/abs/1808.06226.

[13] Lee, K. F. a Hon, H. W. *Large-vocabulary speaker-independent continuous speech recognition using HMM* [online]. 1988 [cit. 18. 4. 2024]. DOI: 10.1109/ICASSP.1988.196527. Dostupné z: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=196527.

[14] Lee, M. *GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance* [online]. 2023 [cit. 20. 4. 2024]. DOI: arXiv:2305.12073. Dostupné z: https://arxiv.org/abs/2305.12073.

[15] Lu, L., Shin, Y., Su, Y. a Karniadakis, G. E. *Dying ReLU and Initialization: Theory and Numerical Examples* [online]. 2023 [cit. 20. 4. 2024]. DOI: arXiv:2305.12073. Dostupné z: https://arxiv.org/abs/2305.12073.

[16] Moller, H. a Pedersen, C. S. Hearing at low and infrasonic frequencies. *Noise and health*. Medknow. 2004, sv. 6, č. 23, s. 37–57, [cit. 28. 12. 2023].

[17] Nasteski, V. An overview of the supervised machine learning methods. *HORIZONS.B*. Prosinec 2017, sv. 4, s. 51–62, [cit. 6. 4. 2024]. DOI: 10.20544/HORIZONS.B.04.1.17.P05.

[18] O'Shea, K. a Nash, R. *An Introduction to Convolutional Neural Networks* [online]. 2015 [cit. 20. 4. 2024]. DOI: arXiv:1511.08458. Dostupné z: https://arxiv.org/abs/1511.08458.

[19] Radford, A. and Kim, J. W. and Xu, T. and Brockman, G. and McLeavey, C. and Sutskever, I.. *Robust Speech Recognition via Large-Scale Weak Supervision* [online]. 2022 [cit. 26. 12. 2023]. DOI: arXiv:2212.04356. Dostupné z: https://doi.org/10.48550/arXiv.2212.04356.

[20] Rao, M., Dheram, P., Tiwari, G., Raju, A., Droppo, J. et al. *Do as I mean, not as I say: Sequence Loss Training for Spoken Language Understanding* [online]. 2021 [cit. 16. 4. 2024]. DOI: arXiv:2102.06750. Dostupné z: https://arxiv.org/abs/2102.06750.

[21] Sennrich, R., Haddow, B. a Birch, A. *Neural Machine Translation of Rare Words with Subword Units* [online]. 2016 [cit. 22. 4. 2024]. DOI: arXiv:1508.07909. Dostupné z: https://arxiv.org/abs/1508.07909.

[22] Sharkawy, A. N. Principle of Neural Network and Its Main Types: Review. *Journal of Advances in Applied & Computational Mathematics*. 2020, sv. 7, s. 8–19, [cit. 20. 4. 2024]. Dostupné z: https://doi.org/10.15377/2409-5761.2020.07.2.

[23] Sharma, S., Sharma, S. a Athaiya, A. Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*. 2020, sv. 4, s. 310–316, [cit. 20. 4. 2024]. DOI: 10.33564/IJEAST.2020.v04i12.054. Dostupné z: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf.

[24] Song, X., Salcianu, A., Song, Y., Dopson, D. a Zhou, D. *Fast WordPiece Tokenization* [online]. 2024 [cit. 22.4. 2024]. DOI: arXiv:2012.15524. Dostupné z: https://arxiv.org/abs/2012.15524.

[25] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. a Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014, sv. 15, č. 56, s. 1929–1958, [cit. 15. 4. 2024]. Dostupné z: http://jmlr.org/papers/v15/srivastava14a.html.

[26] Takase, S. a Okazaki, N. *Positional Encoding to Control Output Sequence Length* [online]. 2019 [cit. 19. 4. 2024]. DOI: arXiv:1904.07418. Dostupné z: https://arxiv.org/abs/1904.07418.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. Attention Is All You Need. *CoRR*. 2017, abs/1706.03762, [cit. 16. 4. 2024]. Dostupné z: http://arxiv.org/abs/1706.03762.

[28] Wang, D., Wang, X. a Lv, S. An Overview of End-to-End Automatic Speech Recognition. *Symmetry*. Srpen 2019, sv. 11, [cit. 18. 4. 2024]. DOI: 10.3390/sym11081018. Dostupné z: https://www.mdpi.com/2073-8994/11/8/1018.

[29] Řízení letového provozu České republiky. *VFR příručka – Česká republika, kap. 6 – Letové provozní služby* [online]. 2024 [cit. 5.5. 2024]. Dostupné z: https://aim.rlp.cz/vfrmanual/actual/pdf/gen_6_cz.pdf.

Příloha A

Chybovost modelů

A.1 Model pro plný přepis

Jazyk	Nevidění řídící (LKKU)	Neviděná vysílání letadel (LKKU)	WER dle jazyků (LKKU)	WER LKKU celkem	Nahrávky z jiného letiště (LKPR, LKTB)
Výchozí natrénovaný model					
Čeština	16,0 %	16,1 %	16,0 %	**16,5 %**	19,1 %
Angličtina	14,9 %	20,8 %	16,9 %		-
Experiment 1 – epochy					
Čeština	16,0 %	16,1 %	16,0 %	**16,5 %**	19,1 %
Angličtina	14,9 %	20,8 %	16,9 %		-
Experiment 2 – míra učení					
Čeština	14,2 %	14,5 %	14,3 %	**14,9 %**	19,0 %
Angličtina	13,9 %	21,1 %	16,3 %		-
Experiment 3 – warmup					
Čeština	15,0 %	14,8 %	15,0 %	**15,4 %**	19,0 %
Angličtina	13,5 %	21,1 %	16,1 %		-
Experiment 4 – dropout					
Čeština	14,5 %	14,7 %	14,5 %	**15,3 %**	18,3 %
Angličtina	13,9 %	23,2 %	17,0 %		-
Experiment 5 – batch size					
Čeština	14,9 %	13,6 %	14,5 %	**14,7 %**	19,6 %
Angličtina	12,1 %	17,9 %	14,1 %		-
Experiment 6 – zamrazení enkodéru					
Čeština	17,7 %	18,5 %	18,0 %	**18,6 %**	20,5 %
Angličtina	16,3 %	25,9 %	19,5 %		-
Experiment 7a – přidání anglických dat					
Čeština	66,1 %	40,4 %	57,0 %	**49,9 %**	84,9 %
Angličtina	19,2 %	19,8 %	19,4 %		-
Experiment 7b – trénování na kunovických a anglických ATCO2 datech					
Čeština	16,5 %	14,2 %	15,7 %	**16,0 %**	20,8 %
Angličtina	14,2 %	19,8 %	16,1 %		-

Tabulka A.1: WER Whisperu po jednotlivých trénováních na plných přepisech.

A.2 Model pro zkrácený přepis

Jazyk	Nevidění řídící (LKKU)	Neviděná vysílání letadel (LKKU)	WER dle jazyků (LKKU)	WER LKKU celkem	Nahrávky z jiného letiště (LKPR, LKTB)
Výchozí natrénovaný model					
Čeština	26,0 %	22,7 %	24,8 %	**25,6 %**	30,3 %
Angličtina	22,2 %	36,8 %	27,1 %		-
Model natrénovaný na nejlepších hyperparametrech pro plný přepis					
Čeština	18,0 %	18,4 %	18,1 %	**20,0 %**	34,0 %
Angličtina	21,2 %	41,0 %	27,9 %		-
Experiment 1 – epochy					
Čeština	21,4 %	21,7 %	21,5 %	**22,7 %**	28,2 %
Angličtina	22,2 %	38,1 %	27,5 %		-
Experiment 2 – míra učení					
Čeština	19,4 %	21,9 %	20,3 %	**21,0** %	29,0 %
Angličtina	17,8 %	31,8 %	22,5 %		-
Experiment 3 – batch size					
Čeština	19,9 %	20,4 %	20,1 %	**21,2** %	28,9 %
Angličtina	20,1 %	35,6 %	25,3 %		-

Tabulka A.2: WER Whisperu po jednotlivých trénováních na zkrácených přepisech.

Příloha B

Obsah přiloženého paměťového média

```
/
├── src/ ..... Zdrojové kódy
│   ├── prep/ ..... Skripty pro přípravu datasetů a práci s přepisy a nahrávkami
│   ├── train/ ..... Trénovací skript
│   ├── eval/ ..... Evaluační skripty
│   └── demo/ ..... Jednoduché rozhraní pro demonstraci modelů
├── models/ ..... Nejlepší modely pro plné a zkrácené přepisy
├── excel/ ..... Plakát, video a článek odevzdávaný na Excel@FIT
├── latex_src/ ..... Zdrojové soubory technické zprávy
└── xnevar00.pdf ..... PDF technické zprávy
```

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024

Automatic Transcription of Air-Traffic Communication to Text

Bachelor's thesis
2024

Author: Veronika Nevařilová
Supervisor: Ing. Igor Szőke, Ph.D.

Purpose

Fine tune a speech recognition model tailored for Czech-English air-traffic communication potentially usable by air traffic controllers (ATCos).

Design a shortened transcription protocol for ATCos' quicker orientation in transcriptions.

Train the model on both full and shortened transcriptions and analyze their performance.

Datasets

Czech and English air-traffic communication recordings of Kunovice airspace (LKKU).

Transcribed with help of **SpokenData.com**.

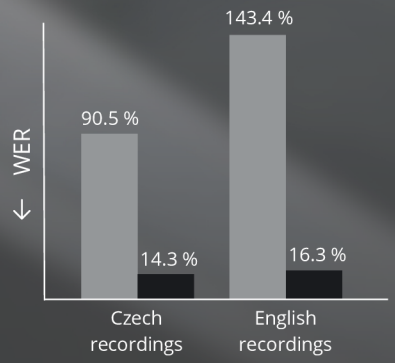
~ 5 hours
of recordings used for training on full and shortened transcriptions.

Model

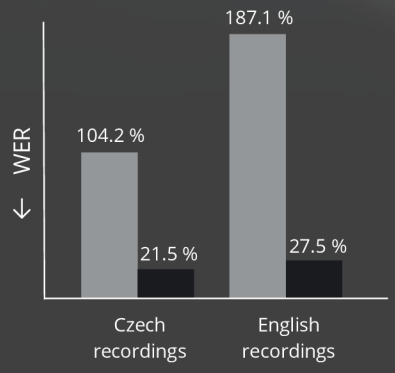
OpenAI Whisper Medium
Pre-trained on 192 hours of Czech language audio

Results

Model for full transcriptions



Model for shortened transcriptions



Whisper Medium baseline
Trained

Fig. 1,2: Word error rate of Whisper baseline and trained model on LKKU Czech and English data

Output examples

Full transcription

Oscar Kilo Alpha Bravo Charlie dráha nula dva střední přistání povoleno vítr nula jedna nula stupňů pět uzlů

Shortened transcription

OKABC dráha 02C přistání povoleno vítr 010 stupňů 5 uzlů