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

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. JAN HOLÁŇ
AUTHOR

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

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání diplomové práce

161415

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Student:	**Holáň Jan, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Strojové učení
Název:	**Automatický přepis řeči letecké komunikace do textu**
Kategorie:	Zpracování signálů
Akademický rok:	2024/25

Zadání:

1. Seznamte se s automatickým rozpoznáváním řeči a s rozpoznávačem Whisper.
2. Seznamte se s dodanými daty. Vytvořte trénovací a testovací datasety. Seznamte se s HuggingFace knihovnou Transformers a modelem Whisper.
3. Adaptujte rozpoznávač Whisper. Iterativně vyhodnocujte přesnost přepisu. Zaměřte se na různé formy textového výstupu (např. plný a zkrácený). Prozkoumejte možnosti adaptace Whisperu pomocí promptu.
4. Adaptujte Whisper pomocí promptu (okolní volací znaky, traťové body, čísla dráhy, atd.). Vyhodnoťte dopad na přesnost přepisu.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. 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:	1.11.2024
Termín pro odevzdání:	21.5.2025
Datum schválení:	12.11.2024

Abstrakt

Tato práce se zabývá automatickým přepisem letecké komunikace do textové podoby s využitím modelu Whisper. Úvodní část představuje specifika letecké komunikace, proces přípravy dat a stručný přehled architektury modelu Whisper založeného na principu transformerů.

Hlavní část práce se zaměřuje na experimentální využití promptů, konkrétně na začlenění volacích znaků jako klíčových informací v promluvách. Předpokládá se, že pro každou komunikaci jsou známy volací znaky letadel, se kterými může být vedena. Byly testovány různé typy promptů s cílem analyzovat, jak model Whisper zpracovává tyto informace v kontextu použitých dat.

Pro testování vlivu promptů byly natrénovány různé varianty modelů na základě použitých datových sad, přičemž trénování i testování probíhalo na datech s plnou i zkrácenou formou přepisu. Výsledky ukazují, že použití promptů obecně zvyšuje kvalitu přepisu, přičemž nejlepších výsledků bylo dosaženo při použití promptů obsahujících pouze volací znaky přítomné v jednotlivých promluvách. Dále se ukázalo, že množství trénovacích dat má vliv na stabilitu modelů při provádění experimentů s promptem.

Abstract

This thesis deals with the automatic transcription of air traffic into text form using the Whisper model. The introductory part presents the specifics of air traffic, the data preparation process and a brief overview of the architecture of the Whisper model based on the principle of transformers.

The main part of the thesis focuses on the experimental use of prompts, specifically the incorporation of call signs as key information in speech. It is assumed that for each communication the call signs of the planes with which it can be conducted are known. Different types of prompts were tested to analyze how the Whisper model handles this information in the context of the data used.

To test the effect of the prompts, different variants of the models were trained on the datasets used, with both training and testing performed on both full and short form transcript data. The results show that the use of prompts generally improves transcription quality, with the best results achieved when using prompts containing only the call signs present in individual utterances. Furthermore, the amount of training data was shown to affect the stability of the models when performing experiments with the prompt.

Klíčová slova

finetuning, Whisper, Whisper Medium, NLP, zpracování přirozeného jazyka, transformery, příprava datasetů, trénování neuronových sítí, prompt, letecká komunikace

Keywords

finetuning, Whisper, Whisper Medium, NLP, natural language processing, transformers, dataset preparation, neural network training, prompt, air traffic communication

Citace

HOLÁŇ, Jan. *Automatický přepis řeči letecké komunikace do textu.* Brno, 2025. Diplomová 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 diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szőkeho, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

. . . . . .
Jan Holáň
20. května 2025

Poděkování

Děkuji svému vedoucímu, panu Ing. Igoru Szőkemu, Ph.D., za milý a vstřícný přístup při konzultacích mé práce a za velmi užitečné a věcné rady.

Obsah

1 Úvod **10**

2 Příprava dat a evaluace **11**
2.1 Evaluační metrika WER . . . 13
2.1.1 Příklad výpočtu WER . . . 13
2.1.2 WER volacích znaků . . . 14
2.2 Popis dat . . . 14
2.3 Obecná úprava datasetů . . . 15
2.4 A-PiMod . . . 16
2.5 HIWIRE . . . 17
2.6 MALORCA . . . 18
2.7 NATO . . . 18
2.8 UWB . . . 19
2.9 ATCO2 . . . 20
2.9.1 Úprava datasetu ATCO2 pro trénování související s využitím promptu 22
2.10 Souhrn trénovacích a testovacích datasetů . . . 26
2.10.1 Základní vyhodnocení datasetů – forma *true-case* . . . 27

3 Whisper a použité prostředí, neuronové sítě a transformery **29**
3.1 Model Whisper . . . 29
3.1.1 Struktura . . . 29
3.1.2 Prompt . . . 31
3.1.3 Verze Whisperu . . . 33
3.2 Neuronové sítě . . . 33
3.2.1 Loss funkce . . . 34
3.3 Transformery . . . 34
3.4 Encoder-decoder architektura Whisperu . . . 36
3.5 Prostředí . . . 37
3.5.1 Prostředí pro základní adaptaci Whisperu . . . 37
3.5.2 Prostředí pro kapitoly Finetuning I.,II.,III. . . . 38
3.6 Schéma a popis trénování . . . 39

4 Základní adaptace Whisperu **42**
4.1 Volba hyperparametrů pro trénování . . . 42
4.2 Vzájemná evaluace a trénování modelů na různých datasetech s plnými přepisy 43
4.3 Natrénování modelu na všech datasetech s plnými přepisy . . . 45
4.4 Závěr . . . 45

5 Finetuning I. – Adaptace Whisperu na plné přepisy **47**
5.1 Nastavení hyperparametrů trénování 47
5.2 Konvence pojmenovávání modelů 48
5.3 Trénování 49
5.4 Testování modelů 50
5.4.1 Testy natrénovaných modelů bez promptu 50
5.4.2 Testování modelů na datech s promptem 55
5.5 Závěr 58

6 Finetuning II. – Adaptace Whisperu na plných přepisech s použitím promptu **60**
6.1 Použité modely a konvence názvů 60
6.2 Trénování 61
6.3 Testování 61
6.3.1 Testování modelů na datech s promptem 63
6.3.2 Experiment I. – testování modelů na datech bez promptu 67
6.3.3 Experiment II. – všechny špatné vs. všechny dobré volací znaky . . . 68
6.3.4 Experiment III. – stabilní vs. nestabilní pozice volacího znaku 69
6.3.5 Experiment IV. – jiné prompty 70
6.4 Závěr 73

7 Finetuning III. – Adaptace Whisperu na zkrácené přepisy bez a s použitím promptu **74**
7.1 Vyhodnocení základních modelů natrénovaných bez promptu 74
7.1.1 Testování modelů na datech bez promptu 74
7.1.2 Testování modelů na datech s promptem 77
7.2 Vyhodnocení modelů natrénovaných s promptem 78
7.2.1 Testování modelů na datech s promptem 78
7.2.2 Testování modelů na datech bez promptu 80
7.3 Závěr 81

8 Závěr **82**

Literatura **85**

A Příklad plného a zkráceného přepisu společně s vytvořenými prompty k těmto přepisům **87**

Seznam obrázků

2.1 Skladba plné a zkrácené formy volacího znaku letadla patřícího letecké společnosti Air France. 12

3.1 Architektura modelu Whisper, převzato z [9] 30
3.2 Skladba vstupu dekodéru modelu Whisper, převzato z [9]. 32
3.3 Obecná architektura transformeru [5]. 35
3.4 Digram průběhu základní adaptace Whisperu, která je popsána v kapitole Základní adaptace Whisperu. 39
3.5 Diagram trénování a pojmenování modelů, které je popsané v kapitolách Finetuning I.,II. a III. Pojmenovaní P-ALLDS, P-BASELINE a P-ALLDSATCOEN označují rodiny modelů se stejným výchozím modelem, trénovaným bez promptu. Každý model v dané rodině vychází ze stejného původního modelu s následnou adaptací na daný typ promptu, tedy **P** značí prompt, na který byl daný model v rodině trénován – můžeme psát P $\in$ {`AG,AG4B,AG35B,AG50CZB,5B,35B,50CZB`}. 40
3.6 Diagram ukazující průběh trénování modelů na datech s promptem a jejich testování na datech s promptem i bez promptu. 41

4.1 Průběh loss funkce při trénování na datasetu ATCO2. 43
4.2 Průběh loss funkce při trénování na datasetu HIWIRE. 43

5.1 Průběh trénovací loss funkce modelů $\text{ATCOEN}_{\text{I.}}$, $\text{ALLDS}_{\text{I.}}$, $\text{ALLDSATCOEN}_{\text{II.}}$, trénování proběhlo pro plný přepis. 50
5.2 Průběh validační loss funkce (modrá), WER hodnot plných přepisů (fialová) a WER hodnot volacích znaků (červená) pro modely $\text{ALLDS}_{\text{I.}}$, $\text{ATCOEN}_{\text{I.}}$ a $\text{ALLDSATCOEN}_{\text{II.}}$. 51
5.3 WER hodnoty plných přepisů, znázorněné pro jednotlivé subsety testovací sady v kontrastu s celkovou WER hodnotou plných přepisů. 52
5.4 WER hodnoty volacích znaků, znázorněné pro jednotlivé subsety testovací sady v kontrastu s celkovou WER hodnotou volacích znaků. 52
5.5 Loss funkce znázorněné pro jednotlivé subsety testovací sady v kontrastu s celkovou loss funkcí. 53
5.6 Grafy ukazující rozložený průběh WER hodnot jednotlivých subsetů testovací sady, přičemž jsou zvýrazněné vertikálními čarami ty epochy, během kterých byla získána nejnižší WER hodnota pro danou množinu dat. 54

6.1 Evaluace jednotlivých epoch pro vybrané modely. 61

6.2 Graf ukazující celkovou úspěšnost přepisu dat pro každý typ promptu rodinou modelů P-BASELINE, přičemž pro každý prompt se liší epocha modelu, ve které dosahovala metrika WER nejnižších hodnot. 62
6.3 Ilustrace ovlivnění přepisu promptem jako nástrojem na adaptaci modelu na řečníka. 66
6.4 WER hodnoty přepisu volacích znaků pro rodinu modelů P-BASELINE. . . 67

7.1 Srovnání celkového průběhu WER hodnot přepisů a volacích znaků pro modely ALLDS, ATCOEN a ALLDSATCOEN pro plné a zkrácené přepisy. . . 75
7.2 Srovnání celkového průběhu loss funkcí pro modely ALLDS, ATCOEN a ALLDSATCOEN pro plné a zkrácené přepisy. 76
7.3 Průběh WER hodnot zkrácených přepisů pro modely natrénované na prompt 50CZB společně s hodnotami pro obdobné modely, trénované na plné přepisy. 79
7.4 Průběh WER hodnot zkrácených přepisů pro modely natrénované na promptech AG a AG35B, společně s hodnotami pro obdobné modely, trénované na plné přepisy. 79

Seznam tabulek

2.1 Mezinárodní hláskovací abeceda 11
2.2 List použitých datasetů v této práci 15
2.3 Rozdělení datové sady A-PiMod 17
2.4 Tabulka rozdělení datové sady HIWIRE 17
2.5 Tabulka rozdělení datové sady MALORCA 18
2.6 Rozdělení trénovací a testovací sady podle zemí datasetu NATO. Poslední řádek obsahuje původní čas všech nahrávek dohromady (velkou část tvořilo ticho a spousta promluv přesahovala délku třicet vteřin) 19
2.7 Počty nahrávek pro trénovací a testovací sadu datasetu UWB 20
2.8 Rozdělení datové sady ATCO2 na testovací a trénovací subsety. 22
2.9 Počty volacích znaků v jednotlivých subsetech ATCO2 před a po doplnění tagů označujících callsigny. 23
2.10 Statistiky týkající se počtu slov a poměru volacích znaků ke zbylému textu pro anglickou část datasetu ATCO2. 26
2.11 Souhrn všech zpracovaných **trénovacích** datasetů 26
2.12 Souhrn všech zpracovaných **testovacích** datasetů 27
2.13 Souhrn všech zpracovaných **trénovacích a testovacích** datasetů použitých pro prvotní adaptaci modelu Whisper popsanou v kapitole Základní adaptace Whisperu. 27
2.14 Hodnoty WER (%) na modelu Vanilla Whisper Medium pro každý dataset. Použité transkripty měly formu plných přepisů. 28
2.15 Hodnoty WER (%) na modelu Vanilla Whisper Medium pro každý dataset. Použité transkripty měly formu zkrácených přepisů. 28

4.1 WER hodnoty (%) pro modely natrénované na trénovacích datasetech A-PiMod, ATCO2, HIWIRE, MALORCA a NATO vyhodnocené na každém testovacím subsetu. Oranžově jsou zvýrazněné hodnoty WER tam, kde se shodoval dataset, ze kterého vycházela trénovací i testovací sada. Šedě je pak označeno pole hodnot, kde se nacházejí výsledky vzájemného testování čtyř datasetů (ATCO2, HIWIRE, MALORCA, NATO). Použité transkripty měly formu plných přepisů. 44
4.2 WER hodnoty (%) modelu natrénovaném na všech vytvořených trénovacích sadách (A-PiMod, ATCO2, HIWIRE, MALORCA,NATO) a vyhodnoceném na všech testovacích datových sadách zvlášť. Použité transkripty měly formu plných přepisů. 45

5.1 Datasety použité při trénování základních modelů bez použití promptu a jejich pojmenování. 49

5.2 Nejlepší získané WER hodnoty modelů pro danou skupinu hyperparametrů s uvedenou epochou, během které bylo hodnot dosaženo. Zvýrazněny jsou nejnižší hodnoty. 51
5.3 Nejnižší dosažené WER hodnoty a epochy, ve kterých jich bylo dosaženo, pro jednotlivé subsety testovací datové sady. Ve třetím sloupci je ukázán průměrný rozdíl WER hodnot mezi sadami Štefánik+Ruzyně (viděná letiště) a Zurich (neviděné letiště). Poslední sloupec pro přehlednost ukazuje celkově nejlepší WER hodnoty i s epochami. 53
5.4 Porovnání úspěšnosti modelů na volacích znacích. V tabulce pozorujeme také vyhodnocení modelu **BASELINE**, tedy Vanilla Whisper Medium. Správně přepsaným volacím znakem je myšlena naprostá shoda mezi jeho formou v referenčním a predikovaném přepisu. Zobrazené hodnoty jsou považovány za **referenční** (pro model ALLDSATCOEN jsou referenční hodnoty z osmé epochy). 54
5.5 WER hodnoty přepisů (%) pro základní modely, které byly testované na všech typech promptů. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě daného základního modelu (testovaného bez promptu, viz tabulka 5.4) – zeleně zlepšení (nižší WER), červeně zhoršení. 55
5.6 WER hodnoty (%) volacích znaků pro testované prompty na základních modelech. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě bez promptu (viz tabulka 5.4) – zeleně zlepšení, červeně zhoršení. . . . 56
5.7 Porovnání úspěšnosti základních modelů **pouze** na těch přepisech z testovací sady ATCO2-EN, které **obsahují volací znaky**. Tyto hodnoty figurují jako referenční pro tabulku 5.8. 58
5.8 WER hodnoty (%) základních modelů na promluvách z testovací sady ATCO2-EN, které obsahují volací znaky. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě bez promptu (tabulka 5.7) – zeleně zlepšení (nižší WER), červeně zhoršení. 58

6.1 Modely, pro které je prováděno srovnání WER hodnot. 61
6.2 Nejnižší dosažené WER hodnoty (%) pro plné přepisy promluv a epochy, ve kterých jich bylo dosaženo, pro modely trénované na promptech AG, AG35B a 50CZB. V levém sloupci se nacházejí označení rodin modelů a každý další sloupec pak označuje model z každé rodiny natrénovaný na daný prompt. . 62

6.3 WER hodnoty (%) plných přepisů ATCO2-EN datasetu za použití promptů na modelech adaptovaných na ně. V levém sloupci je název každé rodiny a v každém dalším sloupci jsou ukázány WER hodnoty adaptovaného modelu na konkrétní typ promptu. Druhý číselný řádek pro každou rodinu modelů obsahuje rozdíl oproti referenční hodnotě modelu trénovaného bez promptu – zeleně zlepšení, červeně zhoršení. Pro rodiny P-ALLDS je referenční hodnotou WER 20.27% získaná modelem ALLDSATCOEN, pro P-BASELINE potom WER 22.48% získaná modelem ATCOEN. Oranžově je označen sloupec pro modely trénované a testované na promptu AG, modře pak sloupec s modely týkající se promptu AG35B. Třetí řádek pro každou rodinu modelů srovnává základní modely ALLDSATCOEN a ATCOEN, testovanými s promptem proti jednotlivým adaptacím modelů na prompt. K modelům P-ALLDSATCOEN neexistuje ekvivalentně trénovaný model (model trénovaný na všech dostupných datasetech a poté dvakrát na ATCO2-EN není dostupný), a tudíž pro něj nejsou zobrazena srovnání. 64
6.4 Srovnání WER hodnot (%) přepisu a volacích znaků v nich pro modely ATCOEN a AG-BASELINE. 65
6.5 WER hodnoty (%) transkripce volacích znaků za použití promptů. V levém sloupci je název každé rodiny a v každém dalším sloupci jsou ukázány WER hodnoty adaptovaného modelu na konkrétní typ promptu. Druhý řádek pro každou rodinu modelů ukazuje rozdíl oproti referenční hodnotě bez promptu (ATCOEN pro rodinu P-BASELINE, ALLDSATCOEN pro rodiny P-ALLDS, tabulka 5.4) – zeleně zlepšení, červeně zhoršení. Třetí řádek je rozdíl proti testům základních modelů na datové sadě s promptem (viz tabulka 5.5). 66
6.6 WER hodnoty (%) **testování bez promptu** modelů adaptovaných na jednotlivé typy promptu. V levém sloupci je název každé rodiny a v každém dalším sloupci jsou ukázány WER hodnoty adaptovaného modelu na konkrétní typ promptu. Druhé číselné řádky ukazují rozdíly oproti testům rodin P-ALLDS, P-BASELINE a P-ALLDSATCOEN s promptem (tabulka 6.3). 68
6.7 WER hodnoty (%) testování modelů, které byly adaptované na prompt AG, s promptem 35B a obráceně – výsledné hodnoty testů na komplementárním promptu jsou označeny fialově. V prvním řádku pro každou rodinu modelů nalezneme hodnoty, kterých bylo dosaženo při trénování i testování na stejném promptu. 69
6.8 WER hodnoty (%) přepisů. Testované modely byly adaptovány na prompt AG35B (vznikly modely AG35B-ALLDS, AG35B-BASELINE a AG35B-ALLDSATCOEN), ve kterém měly správné callsigny stabilní pozici (nacházely se na začátku promptu). WER hodnoty v tabulce se pak týkají testů na promptech se stabilní a randomizovanou pozicí správných volacích znaků. 70
6.9 WER hodnoty (%) volacích znaků. Testované modely byly adaptovány na prompt AG35B, ve kterém měly správné callsigny stabilní pozici (nacházely se na začátku promptu). WER hodnoty (%) v tabulce se pak týkají testů na promptech se stabilní a randomizovanou pozicí správných volacích znaků. . 70

6.10 WER hodnoty (%) přepisů pro testování rodiny P-BASELINE na dvou různých setech promptů. PROMPT DS I. obsahuje prompty používané na všech ostatních místech v této práci, PROMPT DS II. je použitý pouze zde pro srovnání. Oranžově je označen řádek s rozdílem celkových WER hodnot pro oba sety promptů. 71
6.11 WER hodnoty (%) volacích znaků pro testování rodiny P-BASELINE na dvou různých setech promptů. PROMPT DS I. obsahuje prompty používané na všech ostatních místech v této práci, PROMPT DS II. je použitý pouze zde. Oranžově je označen řádek s rozdílem celkových WER hodnot volacích znaků pro oba sety promptů, modře pak řádek s rozdílem správně přepsaných volacích znaků. 72

7.1 Nejlepší získané WER hodnoty modelů pro danou skupinu hyperparametrů s uvedenou epochou, během které bylo hodnot dosaženo. Zvýrazněny jsou nejnižší hodnoty. **Modely jsou do trénování s promptem převzaté právě v těchto epochách.** Při dalším zmiňování modelů ALLDS, ATCOEN a ALLDSATCOEN již budeme mít na mysli ty verze, které dosáhly nejnižších hodnot WER tak, jak jsou vyznačeny v této tabulce. 75
7.2 Nejnižší dosažené WER hodnoty a epochy, ve kterých jich bylo dosaženo, pro jednotlivé subsety testovací datové sady. Ve třetím sloupci je ukázán průměrný rozdíl WER hodnot mezi sadami Štefánik a Ruzyně (viděná letiště) a Zurich (neviděné letiště). Poslední sloupec opět pro přehlednost ukazuje celkově nejlepší WER hodnoty s epochami. 76
7.3 Porovnání úspěšnosti natrénovaných základních modelů na volacích znacích. V tabulce pozorujeme i vyhodnocení modelu **BASELINE**, tedy Vanilla Whisper Medium. Správně přepsaným volacím znakem je myšlena naprostá shoda mezi jeho formou v referenčním a predikovaném přepisu. 77
7.4 WER hodnoty (%) přepisů pro základní modely, které byly testované na všech typech promptů. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě: BASELINE (84.74 %), ALLDS (34.02 %), ATCOEN (36.48 %), ALLDSATCOEN (33.22 %) – zeleně zlepšení (nižší WER), červeně zhoršení. 77
7.5 WER hodnoty (%) modelů trénovaných s prompty na zkrácených přepisech ATCO2-EN datasetu, testovaných s použitím promptů. Každý model je testován na tom promptu, na jakém byl trénován. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě modelu trénovaného bez promptu – zeleně zlepšení, červeně zhoršení. Pro modely P-ALLDS je referenční hodnotou WER 33.22 % získaná modelem ALLDSATCOEN, pro P-BASELINE potom WER 36.48% získaná modelem ATCOEN. Oranžově je označen sloupec pro modely trénované a testované na promptu AG, modře pak sloupec s modely týkající se promptu AG35B. Třetí řádky srovnávají výsledky testů základních modelů ALLDSATCOEN a ATCOEN na datech s promptem (viz tabulka 7.4), proti výsledkům rodin modelů P-ALLDS a P-BASELINE. K modelům P-ALLDSATCOEN neexistuje ekvivalentně trénovaný referenční model (model trénovaný na všech dostupných datasetech a poté dvakrát na ATCO2-EN není dostupný) a tudíž pro něj nejsou zobrazena srovnání. . . . 80

7.6 WER hodnoty (%) získané z testování bez promptu modelů adaptovaných na jednotlivé typy promptu. Druhé číselné řádky ukazují rozdíly oproti testům stejných modelů s promptem. 81

Kapitola 1

Úvod

Letectví provází od počátku nezbytná komunikace. Ať se jedná o přistávání či vzlet, je důležité, aby si pilot byl jistý dráhou pro přistání, povětrnostními podmínkami i třeba možností letadlo zaparkovat. Na druhé straně scény potom stojí řídící na věži, kteří musí mít dokonalý přehled o stavu letiště a vzdušného prostoru. U malých letišť nebývá provoz tak hustý, zatímco u letišť velkých, například ve Frankfurtu, mohou přistávat letadla každé dvě minuty. Takový provoz vyžaduje bezchybnou komunikaci s minimem přeslechů, a tedy nutných zopakování. V dnešní době pro tyto případy přichází na scénu právě automatický přepis letecké komunikace, který by zabezpečil lepší informační přenos.

Tato práce nachází inspiraci a částečně navazuje na bakalářskou práci Veroniky Nevařilové [7], jejímž cílem bylo adaptovat modely na získané nahrávky komunikace mezi řídícími a piloty. Jak se však ukazuje, podmínky jednotlivých letišť, za kterých probíhá nahrávání, jsou natolik zásadní, že jen stěží můžeme pomýšlet na generalizaci přepisu pro všechna letiště.

Motivace pro zlepšení výsledků dosavadních modelů vychází z nutnosti korektně překládat dráhy a volací znaky, jakožto klíčové informace. Dokud nebudou v této oblasti přepisovací mechanismy pracovat na úrovni velmi blízké sta procentům, bude obtížné se na získané transkripce spoléhat.

V této práci je použit model Whisper (verze medium), vytvořen společností OpenAI [9]. Díky velkému datasetu, na kterém byl tento model natrénován, poskytuje solidní základ pro rozpoznávání řeči v letecké komunikaci. Nicméně velkou překážku tvoří pro model nízká srozumitelnost leteckých nahrávek.

Pro trénování je použita série datasetů, získaných od vedoucího práce. Práce si klade za cíl vyzkoušet zlepšení přesnosti přepisů nejen díky množství a kvalitě dat, ale také skrz předání tzv. *promptu*, tedy přídavné informace předané modelu za účelem zlepšení kvality přepisu.

První kapitola se věnuje v úvodu letecké komunikaci a jejím náležitostem. Dále je v ní vysvětlen způsob přípravy dat a použité datasety. Ukázána je také metrika WER použitá v této práci pro vyhodnocování úspěšnosti modelů.

Druhá kapitola se týká principu fungování modelu Whisper, přičemž jsou v ní stručně popsány neuronové sítě, transformery a architektura encoder-decoder modelů. Nachází se zde též popis trénování a evaluace modelu Whisper tak, jak byly prováděny v této práci.

Ve třetí kapitole nalezneme základní trénování modelu Whisper s jeho vyhodnocením. Ve čtvrté, páté a šesté kapitole se nacházejí popis a průběh trénování a testování modelu Whisper na plné a zkrácené přepisy s využitím promptu pro zlepšení přepisů.

V závěru práce jsou potom shrnuty důležité výsledky a jejich vysvětlení.

Kapitola 2

Příprava dat a evaluace

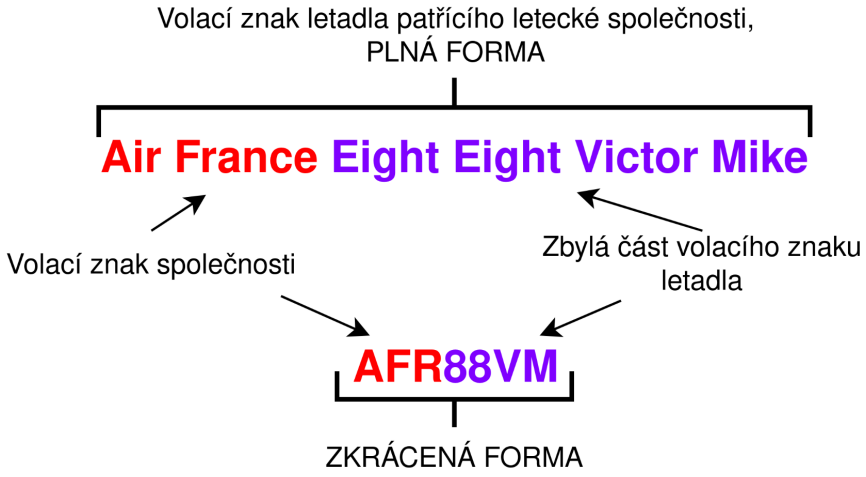
V této kapitole se nachází stručné vysvětlení principů letecké komunikace, dále je popsán způsob přípravy dat a výčet použitých datasetů.

Letecká komunikace je přenášená řeč pilota a řídícího na věži přes VHF kanál. Jeho prostřednictvím jsou předávány důležité informace, kterými jsou především volací znaky (neboli *callsigny*), waypointy a označení drah. Aby byla minimalizována šance neporozumění během komunikace, je zavedeno několik opatření. Prvním z nich je používání letecké abecedy (ukázána v tabulce 2.1) obsahující slova, která jsou vyřčena namísto jednotlivě vyslovovaných písmen. Dalším opatřením je tzv. *readback*, tedy zopakování důležitých údajů (waypointů, callsignů, čísel drah,...) naslouchající stranou. Pokud by došlo k nepřesnosti, informace se zopakuje.

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

Waypoint je ustálené označení zeměpisné polohy. Používá se k vymezení trasy navigace v dané oblasti nebo letové dráhy letadla (z webu [11]). Příkladem můžou být waypointy *AKEVA* nebo *KUVIX*.



Obrázek 2.1: Skladba plné a zkrácené formy volacího znaku letadla patřícího letecké společnosti Air France.

Označení drah je obvykle číselné. Dráha se může objevit v komunikaci například následujícím způsobem: „approach runway two four“.

Dalšími informacemi, které jsou velmi často v letecké komunikaci předávány, jsou například tlak vzduchu (v komunikaci pod označením QNH), radiové frekvence nebo aktuální výška letadla. Zmíněný tlak vzduchu – QNH – tvoří v letecké komunikaci výjimku svým vyslovováním *po písmenech*, tedy bez použití letecké abecedy.

K volacím znaků se vyjadřuje následující část textu.

Volací znaky

Volací znaky jsou unikátní označení letadel skládající se z písmen a čísel. Letecké společnosti mají jinou strukturu volacích znaků, než je tomu u letadel ostatních. Volací znak je vždy vyslovován za použití letecké abecedy, přičemž o takto vysloveném volacím znaku mluvíme jako o plné formě volacího znaku. Jeho zkrácená forma má tvar převedení vyslovených slov na čísla a písmena. Pro běžné letadlo, nepatřící letecké společnosti, může označení vypadat následovně:

- plná (vyslovená) forma – *Oscar Kilo Bravo One Three*,
- zkrácená forma – OKB13.

Pokud se jedná o letadlo, které letecké společnosti patří, tvoří první slova volací znak společnosti – tato slova však nekorespondují s leteckou abecedou a jejich zkrácení je takzvaným ICAO kódem. Příklad plné a zkrácené formy volacího znaku letadla letecké společnosti je ukázán na obrázku 2.1.

Promluvy, které dispečeři letových věží vedou s piloty *by měly* obsahovat volací znaky, aby bylo zřejmé, kterému letadlu jsou dané instrukce předávány. Ty ovšem bývají občas samotnými mluvčími zkráceny. Toto zkracování má obvykle formu vypuštění nějakých částí volacího znaku, takže namísto vysloveného *„Hotel Bravo Zulu Hotel Yankee“* už zazní pouze *„Hotel Yankee“*. Toto se ovšem děje pouze v případě, že si jsou řídící nebo pilot jistí, že se v okolí nenachází jiné letadlo, se kterým by mohl být takto zkrácený volací znak zaměněn.

2.1 Evaluační metrika WER

Tato sekce vychází z knihy [5]. Během testování v následujících kapitolách je vždy využita metrika WER – Word Error Rate. Jedná se o standardní metriku využívanou pro vyhodnocování výstupu ASR modelů (modely využívané pro automatický přepis řeči na text). Její princip spočívá v porovnání, které popisuje, jak moc se liší textový výstup modelu od referenčního výstupu. Toto porovnání se ovšem děje pouze na základě editační vzdálenosti bez znalosti kontextu a významu slov, a tedy může docházet k případům, kdy je WER hodnota nevypovídající. Níže je vzorec pro výpočet WER 2.1.

$$\text{WER } [\%] = 100 \times \frac{I + S + D}{N} \tag{2.1}$$

kde

- I – *Insertions (vložení)* je počet vložených slov navíc oproti referenčnímu textu,
- S – *Substitutions (nahrazení)* je počet nahrazených slov oproti referenčnímu textu,
- D – *Deletions (vymazání)* je počet vymazaných slov oproti referenčnímu textu.
- N je celkový počet slov v referenčním textu.

Hodnota WER je typicky prezentována v procentních bodech.

Mezi další používané metriky patří zmínit CER – Character Error Rate. Ta pracuje na obdobné bázi jako WER, ale místo slov počítá se znaky.

2.1.1 Příklad výpočtu WER

Nyní ukážeme příklad výpočtu WER hodnoty, který budeme vztahovat na následující referenční text a hypotetický výstup modelu:

Referenční text	Hello	this	is	Alpha	Charlie	X-Ray	Zulu		Bravo
Výstup modelu	Ou	this		Alpha	Charlie	raised	Zulu	lay	Bravo

- **Vložení:** lay
- **Nahrazení:** Hello → Ou, X-Ray → raised
- **Vymazání:** is

Výpočet WER

$$\text{WER } [\%] = 100 \times \frac{S + D + I}{N}$$

Kde:

- $S = 2$ (nahrazení)
- $D = 1$ (vymazání)

- $I = 1$ (vložení)
- $N = 8$ (počet slov v referenčním textu)

$$\text{WER } [\%] = 100 \times \frac{2+1+1}{8} = 100 \times \frac{4}{8} = 50\%$$

2.1.2 WER volacích znaků

V této práci je v pozdějších kapitolách zmiňována hodnota WER volacích znaků, v grafech označována jako CAL WER. Jedná se o nejnižší WER hodnotu nalezenou během posouvání okna s referenčním volacím znakem po celém přepisu (výstup modelu). Algoritmický proces, kterým jsou volací znaky vyhodnocovány při testování modelů, je popsán v kapitole *Whisper a použité prostředí, neuronové sítě a transformery* v sekci Prostředí (3.5.2). Níže je ilustrován výpočet WER hodnoty volacího znaku na stejném referenčním textu uvedeném v sekci výše.

Referenční volací znak: Alpha Charlie X-Ray Zulu Bravo

Výstup modelu:	Ou	this	Alpha	Charlie	raised	Zulu	lay	Bravo
Okno na poz. 0	Alpha	Charlie	X-Ray	Zulu	Bravo			
Okno na poz. 1		Alpha	Charlie	X-Ray	Zulu	Bravo		
Okno na poz. 2			Alpha	Charlie	X-Ray	Zulu	Bravo	
Okno na poz. 3				Alpha	Charlie	X-Ray	Zulu	Bravo

WER hodnota pro okno na pozici 0 je počítána pro referenční volací znak a část přepisu pod oknem, tedy *Ou this Alpha Charlie raised*. Obdobně je výpočet prováděn pro další pozice okna. Získané WER hodnoty pro tento příklad by byly:

Okno na poz. 0	100.0 %
Okno na poz. 1	60.0 %
Okno na poz. 2	**40.0 %**
Okno na poz. 3	60.0 %

Výsledná WER hodnota referenčního volacího znaku *Alpha Charlie X-Ray Zulu Bravo* by pro přepis *Ou this Alpha Charlie raised Zulu lay Bravo* byla 40.0 %.

2.2 Popis dat

Data použitá v této práci byla obdržena od vedoucího práce. Jsou získána z různých zdrojů a jejich struktura je velmi různorodá. Z toho důvodu bylo potřeba každý dataset zpracovat zvlášť. K dispozici pro trénování bylo získáno celkem šest datasetů, z nichž jeden byl bez přepisů a zbylé obsahovaly dvojice *WAV* soubor a k němu patřící přepis – ten ovšem v odlišném formátu podle datové sady. Vzhledem k nejednoznačné struktuře souborů s přepisy, jejichž obsah navíc mohl být rozšířen o další informace získané z přiložených souborů, bylo nutné vytvořit pro každý dataset speciální skript pro převedení transkriptů do formátů, které jsou požadovány jako vstup pro trénovací skript modelu Whisper. Z této formy byly následně vytvořeny datasety, které bylo snadné načíst během trénování. Příklad požadovaného formátu dat je následný:

```
{
    "audio": {
        "path": "/path/to/audio/LKPR_RUZYNE_Radar_120_520MHz_20201026_161349
            .WAV",
        "array": array([0.00000000e+00, 0.00000000e+00, ..., -5.18798828e
            -04]),
        "sampling_rate": 16000
    },
    "full_ts": "Radar CSA One Delta Zulu established CSA One Delta Zulu
        roger contact Ruzyne Tower one three four decimal five six zero
        three four five six zero CSA One Delta Zulu pekny den",
    "short_ts": "Radar CSA 1DZ established CSA 1DZ roger contact Ruzyne
        Tower 134.560 34560 CSA 1DZ pekny den"
}
```


Celkem tvoří použitelná část všech dat dohromady pro testování a trénování přes **36 hodin**. Celková délka obdržených záznamů je mnohem delší, avšak jejich objemná část není použitelná vzhledem k neexistenci přepisů, nevyužitelnému tichu nebo úsekům, které sice mají přiřazené přepisy, ale jejich doba trvání přesahuje třicet vteřin (**model Whisper může najednou přepsat pouze třicet vteřin audia**). Použité datové sady jsou zobrazeny v tabulce 2.2 s poznámkou odkazující se na projekt, ze kterého data pocházejí.

A-PiMod	projekt A-PiMod[1] – Applying Pilot Models for Safer Aircraft
HIWIRE	projekt HIWIRE[2]
MALORCA	projekt MALORCA[3]
NATO	zdroj [4]
UWB	projekt Inteligentní technologie pro zvýšení bezpečnosti letového provozu [15]
ATCO2	projekt ATCO2 [6]

Tabulka 2.2: List použitých datasetů v této práci

V následujících sekcích jsou popsány odlišnosti a specifika jednotlivých datasetů. Názvy odpovídají projektům, ze kterých data pocházejí. Nutno poznamenat, že nebylo možné z osobních zkušeností ověřovat, zda přepisy odpovídají promluvám v nahrávkách, a zpravidla tedy nebyly ani nijak zdokonalovány.

2.3 Obecná úprava datasetů

Jednotlivé přepisy u jednotlivých datasetů nebyly vždy v žádané podobě. Především se jedná o zajištění takzvané *plné* a *zkrácené* formy přepisu. Tím je myšleno následující: pokud je v náhravce vyslovena věta *„Oscar Kilo Papa Mike Bravo descend flight level one hundred“*, je toto zároveň její plná forma, zatímco zkrácená bude mít tvar *„OKPMB descend flight level*

[1] https://www.tcd.ie/cihs/our-research/current-projects/apimod.php
[2] https://www.academia.edu/24112615/The_HIWIRE_database_a_noisy_and_non_native_english_speech_corpus_for_cockpit_communication
[3] https://www.malorca-project.de/wp/

100“. Každá z těchto forem bude následně sloužit pro trénování jiného modelu (kdy každý z nich bude přepisovat letecké promluvy na danou formu). Kontrola a úprava transkripcí jednotlivých datasetů se skládala ze dvou kroků:

1. Zajištění, že extrahovaný přepis bude v korektní plné formě. To vyžadovalo, aby žádná jeho část nebyla zkrácená. Tedy například nesmělo dojít ke stavu uložení plné formy ve tvaru *„**Oscar** KPM **Bravo** descend flight level 1 **hundred**“*. Takové chyby by totiž nutně vedly ke zmatení modelu během trénování. Další nezbytností bylo odstranění speciálních značek z přepisu, které sloužily pro označení různých specifických částí promluvy. Například se jednalo o části obsahující volací značky (**callsign**), nebo části s numerickou hodnotou.

2. Korektní plné přepisy byly zkráceny do plné formy. K tomu byl ručně vytvořen slovník číslovek a slov letecké abecedy i s variantami případných překlepů, takže slovník obsahoval více klíčů pro jedno písmeno. Níže je uvedena jeho část:

   ```
   {
       "alpha": "A", "alfa": "A", "bravo": "B", "charlie": "C",
       "charly": "C"
   }
   ```

   Pro nalézání podobností slov ve zkracovaném textu s klíči tohoto slovníku byla využita knihovna RapidFuzz[4]. Zároveň bylo využíváno označení ve formě *tagů* v poskytnutých přepisech, díky čemuž mohla být lépe přepisována slova v částech označených například jako *callsign* (hledala se podobnost s čísly a písmeny letecké abecedy) nebo *value* (hledala se podobnost jenom s čísly).

2.4 A-PiMod

Dataset neobsahoval žádné přepisy, pouze dva WAV soubory, které obsahovaly větší množství promluv. Při částečném zjišťování obsahu WAV souborů se ukázalo, že se jejich obsah překrývá, a proto byl pro trénování využit pouze větší z nich. Protože soubor byl časově příliš dlouhý, bylo nutné jej nastříhat, což bylo provedeno triviálně podle ticha. Výsledek je ukázán v tabulce 2.3.

Kvůli absenci přepisů bylo rozhodnuto vytvořit pro tuto práci přepisy strojově s využitím již dříve (v bakalářské práci [7]) natrénovaného modelu[5] Whisper. Kvalita nicméně byla velmi nízká, což bylo způsobeno zřejmě uzpůsobením použitého modelu především na český jazyk.

Z následně vzniklého datasetu nebyla žádným způsobem vydělena testovací sada a celý obsah byl **ponechán pro trénování** (i přes nepřesnost přepisů).

[4] https://github.com/rapidfuzz/RapidFuzz
[5] https://huggingface.co/BUT-FIT/whisper-ATC-czech-full

Rozdělení	Počet nahrávek	Čas
Trénovací sada	105	00:13:29
Celkový čas dostupných dat	–	00:50:23

Tabulka 2.3: Rozdělení datové sady A-PiMod

2.5 HIWIRE

Tento dataset nebyl poskytnut v celé velikosti – chybělo zastoupení některých jazyků popsaných v dokumentaci k této datové sadě.

Zpracovány byly promluvy několika desítek mluvčích z Francie, Řecka a Španělska. Promluvy byly vždy v angličtině pronášeny nerodilými mluvčími. Nejedná se o reálně vedené promluvy mezi letcem a věží. Jednotlivé nahrávky pocházejí ze studia (za použití mikrofonu pro *blízkou mluvu*) a následně k nim byl přidán šum ve třech různých úrovních. Tento šum byl nahrán v kokpitu letadla Boeing 737 během běžné cesty letadla. Pro tuto práci bylo umožněno pracovat pouze s přidanou nízkou hladinou šumu (druhé dvě modifikace nebyly dodány), což způsobuje snadnou srozumitelnost řeči. Přesto by měla data posloužit alespoň pro naučení formy letecké komunikace.

Přepisy všech nahrávek jsou umístěny v jediném souboru, ze kterého je bylo nutno extrahovat pomocí regulárního výrazu. Přepisy neobsahovaly žádná vyznačení volacích znaků či jiných hodnot, a proto nebylo možné zkracovat text se zaměřením na daná místa, ale pouze obecně. Všechny nahrávky měly délku pod třicet vteřin, a tedy nebylo nutné dělat žádné úpravy.

Rozdělení

V datasetu neexistovalo explicitní rozdělení na trénovací a testovací sadu. Vzhledem k přesně uvedeným informacím o pohlaví řečníka a jeho rodném jazyce u každé nahrávky bylo rozdělení dat provedeno tak, aby mateřský jazyk a pohlaví řečníků byly rovnoměrně zastoupeny v testovací i trénovací sadě. V tabulce 2.4 je výsledek rozdělení (s počty nahrávek) na tři testovací sady podle mateřské řeči jednotlivých řečníků a jednu trénovací sadu. Z původního datasetu nebyla odebrána žádná data.

Rozdělení	Počet nahrávek	Čas
Test	FR: 800 (4 muži, 4 ženy)	00:32:05
	GR: 400 (2 muži, 2 ženy)	00:28:41
	SP: 300 (2 muži, 1 žena)	00:13:06
Train	FR: 2300, GR: 1033, SP: 699	03:18:19
Celkem	5532	04:32:11

Tabulka 2.4: Tabulka rozdělení datové sady HIWIRE

2.6 MALORCA

Tento dataset obsahoval skutečné nahrávky komunikace mezi letcem a věží. Transkripty byly prováděny profesionály a celý dataset je již rozdělen na trénovací a testovací sadu, díky čemuž již nebylo nutné toto rozdělení vytvářet. Dataset obsahoval část dat, ke kterým byly vytvořeny pouze automatické přepisy bez lidské kontroly – tato část nebyla v této práci použita.

Jednotlivé nahrávky jsou samostatnými promluvami buď letců, nebo řídících na věži. Ke každé nahrávce je poskytnut soubor s plným přepisem nahrávky a soubor, ve kterém je tento plný přepis doplněn značkami, které udávají význam některých částí promluvy (jedná se především o části obsahující volací značky, číselné hodnoty či vyslovené pokyny) – tento značkami obohacený soubor s plným přepisem byl použit pro vytvoření zkrácených přepisů.

Níže je v tabulce 2.5 uvedeno rozdělení na trénovací a testovací sadu s počty nahrávek. Všechny nahrávky, kromě sady s automatickými přepisy, byly použity.

Rozdělení	Počet nahrávek	Čas
Test	1557	01:55:33
Train	6335	07:56:45
Celkem	7892	09:52:18

Tabulka 2.5: Tabulka rozdělení datové sady MALORCA

2.7 NATO

Tento dataset byl vyvinut výzkumníky z NATO pro technologii řeči a jazyka s cílem poskytnout vojensky orientovanou databázi pro studium zpracování vícejazyčné řeči. Data byla nahrána během vojenského námořního cvičení ve čtyřech zemích a obsahují výhradně anglické promluvy. Zeměmi, ve kterých se nahrávalo, jsou Kanada, Německo, Nizozemsko a Velká Británie. Nahrávky jsou také rozřazeny právě podle těchto zemí, čehož je následně využito pro vyvážené vytvoření trénovací a testovací sady.

Jednotlivé nahrávky byly příliš dlouhé a bylo nutné je nastříhat, což bylo možné díky dodaným časovým značkám u každé promluvy v dodaných přepisech. Časové úseky s delšími promluvami než 30 vteřin byly zahozeny. Ke každému původnímu WAV souboru patří jeden soubor s přepisy, přičemž ten je v záhlaví opatřen volacími značkami mluvčích, kteří v dané nahrávce promlouvají. Zároveň se identifikace daných mluvčích nacházejí u každé konkrétní zaznamenané promluvy – bylo nutné spárovat tyto identifikační značky s volacími znaky uvedenými v hlavičce souboru.

Podstatnou část nahrávek tvořilo ticho, a tudíž celková využitelná část je mnohem menší než celkový čas nahrávek. Také některé úseky byly tvořeny prázdnými přepisy a rovněž tak nebyly použity. Původní čas obdržených surových dat je pro srovnání uveden v tabulce 2.6.

Rozdělení

Díky identifikaci jednotlivých mluvčích v souborech s přepisy je možné zajistit, aby se mluvčí neopakovali v trénovací a testovací sadě. Zároveň je zajištěno přibližně stejné množství jak mužských, tak ženských promluv v testovací sadě. U trénovací sady tomu tak není,

protože celkově je zastoupení ženských a mužských mluvčích velmi rozdílné. Data z Německa obsahovala pouze mužské promluvy.

Pro tento dataset bylo rozhodnuto spojit všechny uvedené země do výsledné jedné trénovací a jedné testovací sady. V tabulce 2.6 je toto rozdělení uvedeno spolu s detailní informací o zastoupení jednotlivých zemí.

Rozdělení podle země	Počet promluv	Čas
UK test	98	00:15:02
UK train	259	00:37:53
CA test	237	00:30:36
CA train	614	01:15:52
NL test	97	00:19:02
NL train	272	00:37:21
DE test	60	00:06:41
DE train	228	00:21:14
Celkem test	492	01:11:21
Celkem train	1373	02:52:20
Původní celkový čas	–	19:56:43

Tabulka 2.6: Rozdělení trénovací a testovací sady podle zemí datasetu NATO. Poslední řádek obsahuje původní čas všech nahrávek dohromady (velkou část tvořilo ticho a spousta promluv přesahovala délku třicet vteřin)

2.8 UWB

Tento dataset byl dodán bez rozdělení na trénovací a testovací sadu. Obsahuje reálné nahrávky komunikace mezi věží a letcem a přepisy byly získány manuálně s informací o řečnících v podobě značek v přepisech, ale bez bližší specifikace, takže lze pouze rozeznat, zda mluví věž, nebo pilot. Kromě těchto tagů se v přepisech vyskytují i mnohé další, označující například nesrozumitelnou část, jazyk konkrétní části nebo označení šumu. V kulatých závorkách je potom občas uvedena výslovnost slova, přičemž před závorkami se nachází jeho přepis.

Žádná z výše zmíněných značek nebyla využitelná při extrakci plných přepisů a všechny značky bylo nutné regulárním výrazem odstranit. Problém při zpracování činil fakt, že jednotlivé transkripce obsahovaly jak zkrácený zápis některých slov, tak jejich plnou formu. Níže je uvedena ukázka přepisu, který se objevil v dodaných datech a který obsahuje kombinaci plné a zkrácené formy.

Získaný přepis: Shamrock 6 7 X keep rate of climb one thousand or more [noise]

Očekávaný tvar přepisu: Shamrock six seven x-ray keep rate of climb one thousand or more [noise]

A proto, jak bylo zmíněno v sekci Zdokonalení přepisu (2.3), bylo nezbytné v první fázi převést transkripce na korektní plné formy a následně, ve druhé fázi, z nich vytvořit formy zkrácené.

Rozdělení

V souboru s přepisy nahrávek se rovněž vyskytují časové značky, a protože jednotlivé nahrávky jsou příliš dlouhé, bylo časových značek využito pro segmentaci těchto nahrávek. K velkému množství promluv nebyly k dispozici přepisy, takže tato část datasetu nebyla vůbec použita. Ze segmentovaných nahrávek, které obsahovaly přepisy, byla ponechána pouze část jako testovací sada (sloužila jako zcela neviděná data). Trénování na datech UWB nebylo provedeno.

V tabulce 2.7 lze vidět pro srovnání testovací sadu, počet segmentovaných dat s přepisy a celkovou velikost dodaných surových dat.

Rozdělení	Počet segmentovaných promluv	Čas
Testovací sada	3899	03:44:25
Celkem surových dat	–	20:34:48

Tabulka 2.7: Počty nahrávek pro trénovací a testovací sadu datasetu UWB

2.9 ATCO2

Tento dataset byl vytvořen ze skutečných leteckých dat, ovšem na rozdíl od zbylých datasetů obsahuje doplňující soubory k jednotlivým nahrávkám. Jednotlivé přepisy byly vytvořeny lidmi. Celá datová sada je rozdělena na dvě hlavní části – anglická data a data v jiných jazycích. Data označená jako anglická však obsahují z části i promluvy v jiných jazycích, takže ty musely být přesunuty k neanglickým datům. Stejně tak v neanglické části se nacházela některá audia v angličtině – ta byla přesunuta k anglickým datům. U cizojazyčných dat převažovala francouzština.

Ke každému *WAV* souboru patří stejnojmenný *XML* soubor, ve kterém se nachází rozdělení dané nahrávky do segmentů. Každý segment pak obsahuje především transkripci části nahrávky, začátek a konec segmentu v nahrávce a informaci, zda je přepis v anglickém jazyce. Můžeme nalézt *tagy* s dalšími údaji, ty již ale nebyly použity při vytváření cílových datasetů. Jeden WAV soubor odpovídá zpravidla jednomu kontextově souvislému dialogu mezi pilotem a řídícím, což odpovídá stejnému volacímu znaku, který je zmiňován v celé nahrávce. Výjimečně se ale v nahrávce vyskytuje více volacích znaků, které upozorňují na více promluv mezi různými piloty a jedním řídícím.

Níže je ukázka struktury jednoho XML souboru:

```
<data>
<segment>
    <start>0</start>
    <end>3.77</end>
    <speaker>A</speaker>
    <speaker_label>ATCO2 radar</speaker_label>
    <text>Oscar Kilo Papa Mike Bravo descend flight level one hundred</text>
```

```
    <tags>
        <correct>0</correct>
        <correct_transcript>1</correct_transcript>
        <correct_tagging>0</correct_tagging>
        <non_english>0</non_english>
    </tags>
</segment>
```

Mezi soubory s doplňujícími informacemi, existujícími pro většinu nahrávek, patřil i seznam waypointů a volacích znaků, které by se mohly vyskytovat v daném komunikačním záznamu. Toho bylo využito pro vyzkoušení trénování za pomoci *promptu*, které je popsáno v kapitolách Finetuning II. a Finetuning III. Zároveň jsou *callsigny* v dodaných seznamech spjaty s konkrétním ICAO kódem letecké společnosti, což je využito při vytváření korektních zkrácených přepisů.

Rozdělení

Data byla rozdělena na čistě anglická data a ostatní podle tagu *non_english*, obsaženém v popisových souborech každé nahrávky. V případě, že se v různých segmentech jedné nahrávky neshodovalo označení jazyka, byl za jazyk nahrávky považován ten většinově v ní zastoupený.

Následovalo rozdělení nahrávek na testovací a trénovací sadu, protože oficiální rozdělení zde chybělo. Pro **anglická data** byla zvolena následující strategie: pro testovací sadu byly vybrány nahrávky z letiště **Zurich** a 53 % nahrávek pro letiště **Ruzyně** a 54 % pro **Štefánik**. Zbytek nahrávek byl použit pro trénování. Toto dělení mělo za úkol poskytnout představu o tom, jak dobře bude model pracovat na nahrávkách, jejichž podmínky při trénování neviděl (letiště Zurich) a na druhou stranu jestli dojde ke zlepšení přepisů v případě druhých dvou letišť (Štefánik a Ruzyně), jejichž nahrávací podmínky měl možnost vidět v trénovací sadě.

Cizojazyčných dat bylo signifikantně méně. Obsaženo bylo 125 promluv ve francouzštině a následně dalších 40 v jiných jazycích. Rozhodnuto bylo vytvořit z nich **dvě testovací sady**, z nichž jedna byla složena z 33 francouzských nahrávek, a druhá ze 40 nahrávek ve zbylých jazycích. Jako trénovací sada bylo použito 92 nahrávek opět ve francouzštině.

V tabulce 2.8 je uvedeno rozdělení datasetu ATCO2 na trénovací a testovací sady. Několik nahrávek muselo projít segmentací, protože jejich délka přesahovala třicet vteřin. Proto je v tabulce také uveden počet nahrávek v původní datové sadě a počet nahrávek po zpracování datasetu ATCO2, kdy došlo k segmentaci těch nahrávek, které byly delší než třicet vteřin.

Rozdělení	Původní počet nahrávek	Finální počet nahrávek po úpravě datasetu	Čas
FR Train	80	92	00:10:18
FR Test	33	33	00:03:39
Jiné jazyky Test	40	40	00:05:02
EN Train	1554	1579	03:00:08
EN Ruzyně Test	50	70	00:11:58
EN Štefánik Test	53	53	00:10:29
EN Zurich Test	412	412	00:49:56
Celkem	2222	2279	04:31:30

Tabulka 2.8: Rozdělení datové sady ATCO2 na testovací a trénovací subsety.

2.9.1 Úprava datasetu ATCO2 pro trénování související s využitím promptu

Tato úprava dat se týká finálního trénování modelů souvisejících s promptem – jedná se o kapitoly Finetuning II. a III.. Pro tyto části je využita pouze **anglická část datasetu ATCO2**, na kterou je souhrnně odkazováno označením ATCO2-EN.

Cílem této práce je především vyzkoušet dopad promptu na přepis, nikoli učit model normalizovat text. Z toho důvodu, a také pro zajištění celkové konzistence dat, bylo rozhodnuto nahradit v přepisech veškerá velká písmena malými – tedy zavedení formy tzv. *lower-case*. Tento krok vedl ke snížení chybovosti modelů popsaných v kapitole Základní adaptace Whisperu, který byl trénován na *true-case* formě dat. Případná potřeba *true-case* formy by mohla být snadno získána využitím jiného modelu, který by takovou konverzi z *lower-case* formy prováděl.

Doplnění označení volacích znaků

V testovací i trénovací sadě se vyskytovalo mnoho přepisů, ve kterých nebyl označen žádný volací znak. Proto bylo rozhodnuto, že se do jednotlivých souborů ručně doplní značky, označující volací znak, pokud ten bude v souboru obsažen. Tím měla být zajištěna větší relevantnost výsledků testujících dopad promptu (v následujících kapitolách), protože prompt je v této práci konstruován pouze z volacích znaků. V tabulce 2.9 je ukázán počet označených volací znaků v původním datasetu a po přídavném označování původně neoznačených volacích znaků.

Název datasetu	Počet volacích znaků		Počet promluv s vol. zn.	
	Před doplněním	Po doplnění	Před doplněním	Po doplnění
Ruzyně test	77	103	24	10
Štefánik test	78	94	15	6
Zurich test	508	566	75	32
EN Train	2213	2352	184	97

Tabulka 2.9: Počty volacích znaků v jednotlivých subsetech ATCO2 před a po doplnění tagů označujících callsigny.

Konstrukce promptu

Protože řídící i piloti občas volací znak zkrátí (například na jeho první a poslední slovo), jsou všechny tyto formy téhož volacího znaku zaznamenány v datasetu ATCO2 jako samostatné volací znaky a jsou i takto vyhodnocovány procesem popsaným dále v kapitole Whisper. Níže je ukázán tento případ na fiktivní promluvě:

> Pilot: „Hello here is **One Bravo Zulu Zulu Yankee**, altitude one thousand meters“
>
> Věž: „Hello **One Yankee**, altitude one thousand meters“
>
> **EXTRAHOVANÉ VOLACÍ ZNAKY:** One Bravo Zulu Zulu Yankee, One Yankee

Při nahlížení výsledků této práce, kterých bylo dosaženo při práci s promptem, je nutné mít výše zmíněné upozornění na paměti, protože všechny tyto formy jsou předávány do promptu, i když se jedná o jeden a tentýž volací znak.

Příprava promptu zahrnovala extrakci všech volacích znaků. Mimo ně byly extrahovány také čísla ranvejí, ovšem tyto hodnoty nakonec nebyly využity a celá práce se soustředila pouze na volací znaky.

Protože byl použit pro finální trénování pouze dataset ATCO2-EN, bylo možné získat všechny volací znaky v promluvách relativně snadno díky označeným místům v přepisech, kde se údaje vyskytovaly. Níže je k dispozici ukázka přepisu dat s označenými místy volacích znaků a dalších hodnot. Pouze zmiňme, že pozice v textu s tagy `[#value]` označovaly různé hodnoty, přičemž v ukázce níže se nacházela takto označená hladina letu.

```
[#callsign]Air France One Zero Eight Zulu[/#callsign] [#unnamed]
standby now[/#unnamed] [#command]descend to[/#command] [hes]
[#command]descend[/#command] [#value]flight level seven zero[/#value]
```

Získané volací znaky pak byly převedeny do podoby s malými písmeny a rovněž zkráceny. Samotný proces zkracování volacích znaků obsahoval více kroků:

1. **Vytvoření a načtení slovníků**: Pro správné zkrácení volacích znaků byly vytvořeny slovníky. Ze zdroje [13] byl získán seznam ICAO[6] kódů a volacích znaků jednotlivých leteckých společností. Slovník, vytvořený z těchto dat, měl následující podobu:

[6] ICAO (celým názvem International Civil Aviation Organization) je organizace přidělující čtyřmístné kódy jednotlivým letištím a trojmístné kódy aerolinkám. V této práci se využívají trojmístné kódy, které figurují ve volacích znacích letadel.

```
"ABAN": {
    "icao": "ABE",
    "airline": "Aban Air"
},
```

V něm bylo umožněno vyhledávat podle jednotlivých volacích znaků (to je klíč výše zmíněného slovníku). Obdobně byl vytvořen slovník, ve kterém na místě klíče figuroval přímo název aerolinky.

Třetí slovník byl získán přímo ze souboru *.info*, který byl jedinečný pro každou nahrávku. Ten obsahoval jak volací znaky v plné formě, tak jejich zkrácenou formu. Obsaženy byly callsigny, které se mohly vyskytovat v dané nahrávce. Poslední, čtvrtý, slovník byl vytvořen spojením všech volacích znaků dostupných z info souborů patřících jednotlivým nahrávkám.

2. **Využití slovníků s celými volacími znaky**: Extrahované volací znaky z přepisů byly přímo použity jako klíče do třetího (popř. čtvrtého) slovníku. Aby byla minimalizována chybovost vyhledávání, způsobená například větším množstvím mezer, došlo vždy k použití funkce, která normalizovala vyhledávaný volací znak do podoby malých písmen bez mezer (do stejné formy byly analogicky převedeny klíče všech dalších slovníků). Pokud došlo ke shodě nějakého klíče s vyhledávaným řetězcem, byl rovnou získán zkrácený volací znak a proces zkracování byl ukončen.

3. **Využití slovníků s ICAO kódy a volacími znaky společností**: Protože každý callsign obsahoval vždy nějaké části z letecké abecedy, byly tyto části zkráceny přímou substitucí za odpovídající písmena nebo čísla. Zůstala tak povětšinou nějaká zkrácená část a druhá nezkrácená, obvykle obsahující název společnosti. Tato druhá část byla použita jako klíč pro vyhledávání v prvních dvou definovaných slovnících (první s klíčem v podobě volacího znaku letecké společnosti, druhý s klíčem v podobě názvu aerolinky). Pokud byla nalezena shoda, nahradil ICAO kód danou část volacího znaku. Následně byl vrácen výsledný volací znak (ať úplně či neúplně zkrácen).

Finálním krokem byla konstrukce jednotlivých promptů, tedy informací předávaných modelu za účelem zdokonalení přepisu. Celkem bylo vytvořeno sedm verzí promptu s různým poměrem reálně se vyskytujících volacích znaků v dané promluvě a těch volacích znaků, které v promluvě vyřčeny nebyly. Tato sedmice promptů existovala v *plné* i *zkrácené* formě.

V této práci budeme pracovat s označením **správný volací znak**, kterým rozumíme takový, který se v promluvě vyskytuje. Špatným potom takový, který v promluvě obsažen není. Tedy pokud promluva obsahuje volací znak například *Air France Bravo Oscar One*, množina správných volacích znaků bude obsahovat přesně toto sousloví a množina špatných volacích znaků může obsahovat vše, kromě této sekvence slov, například by mohla vypadat následovně: *Air France Bravo Charlie Charlie, One One Two Zero Zero.*

Níže je uveden seznam vytvořených promptů s popisem jejich významu:

- Všechny správné volací znaky. Označení **AG**.

  Experimentální prompt, jehož smyslem bylo zjistit, zda model začne dokonale přepisovat volací znaky v případě, že bude mít jistotu, že prompt tento volací znak obsahuje. V praxi by toto ovšem nebylo použitelné.

- Všechny správné a čtyři špatné volací znaky. Označení **AG4B**.

  V praxi pravděpodobně nevyužitelný prompt. Jeho cílem bylo ověřit, jestli model zvládne správně přepsat volací znak, pokud se ten bude vyskytovat v nějaké menší množině jiných volacích znaků, které se v promluvě nevyskytují.

- Všechny správné a třicet pět špatných volacích znaků. Označení **AG35B**.

  Stejný cíl jako pro prompt výše, nicméně takto vytvořený prompt by skutečně mohl být využitý v praxi, kdy jsou modelu předány volací znaky letadel, které se v danou chvíli, kdy se promluva uskutečnila, vyskytovaly v okolí letiště a v komunikaci mohly být vysloveny.

- Všechny správné a padesát českých náhodných slov. Označení **AG50CZB**.

  Pouze experimentální prompt. Množina českých slov má za cíl zmást model, protože je zajištěno, že se žádné z nich v promluvě nebude vyskytovat.

- Pět špatných volacích znaků. Označení **5B**.

  Je dobrý předpoklad, že tento prompt bude obsahovat nějaké slovo, které se v promluvě vyskytuje a zkoumáme dopad této malé jednotky správné informace na přepis.

- Třicet pět špatných volacích znaků. Označení **35B**.

  Experimentální prompt mající podobný smysl jako v bodě výše. Je zde však větší pravděpodobnost výskytu shodných slov mezi promptem a promluvou.

- Padesát českých náhodných slov. Označení **50CZB**. Pouze experimentální prompt, který má za cíl ověřit, zda se model kompletně odnaučí využívat prompt v případě, kdy během tréninku v něm nebude obsaženo jediné slovo vyskytující se v promluvě.

Skutečné příklady promptů vytvořených k reálným promluvám jsou uvedeny v příloze A.

Je nutné dodat, že veškeré dodané série „špatných“ volacích znaků a náhodných slov, byly při tvorbě promptů, které je vyžadovaly, vybírány vždy znova a náhodně. Tedy žádné z padesáti českých slov, vyskytujících se v promptu typu 50CZB, nemusí být obsaženo v typu AG50CZB (podobně pro špatné volací znaky v jiných typech promptu).

Celá množina promptů existovala ve zkrácené a plné formě, tedy ke každé nahrávce existovalo celkem čtrnáct promptů.

Nakonec byly ještě přidány ke každé nahrávce v datasetu ATCO2 informace o vyskytujících se volacích znacích v přepisu a počtu, kolikrát byly zastoupeny. Opět byla tato informace vytvořena pro zkrácenou i plnou formu volacích znaků.

Umělé odebrání promptů z daných promluv

Při trénování za použití promptu si lze představit riziko, že si model **příliš zvykne** na existenci volacího znaku v něm a mohl by tak zkolabovat, když mu nebude předán správný volací znak. To by bylo nežádoucí chování, které by se jistě projevilo, když by volací znak nebyl v promptu k dispozici. Z toho důvodu bylo rozhodnuto vyjmout z trénovací sady u pěti procent promluv vytvořené prompty. V kapitole Finetuning III. v sekci Experiment I. (6.3.2) jsou obsaženy výsledky testování bez promptu těch modelů, které byly na prompt adaptovány.

Statistiky výskytu volacích znaků

Základní informací pro vyhodnocování dopadu promptu na přepis jednotlivých promluv je procentuální výskyt callsignů v transkriptech. Protože uvádět tento údaj jenom v absolutních či jen relativních číslech by mohlo být zavádějící, je níže uvedena tabulka2.10 s počty slov, počty písmen a procentuálním obsahem volacích znaků v přepisech.

Název sady	Ruzyně Test	Štefánik Test	Zurich Test	EN Train
Počet promluv[7]	70	53	412	1579
Celkový čas promluv	00:11:58	00:10:29	00:49:56	03:00:08
Celková délka textu bez vol. zn. [slova]	1774	1554	7948	28301
Celková délka vol. zn. [slova]	365	359	1989	8470
Celková délka vol. zn. [ve znacích]	2040	1997	11188	48012
Celkový obsah vol. zn. ve všech promluvách (vážený podle znaků)[%]	19.86%	21.84%	24.61%	29.34%
Celkový počet vol. zn.	103	94	566	2352

Tabulka 2.10: Statistiky týkající se počtu slov a poměru volacích znaků ke zbylému textu pro anglickou část datasetu ATCO2.

2.10 Souhrn trénovacích a testovacích datasetů

Pro první adaptaci modelu Whisper, která je popsána v kapitole Základní adaptace Whisperu, byly použity datasety ve formě *true-case*. Pro všechny ostatní adaptace a experimenty popsané v kapitolách Finetuning I., II. a III. měly přepisy formu *lower-case*. Dále došlo i ke změně z hlediska použitých datasetů – ty reflektují následující tabulky 2.11 a 2.12, které shrnují všechny zpracované datové sady tak, jak byly použity pro trénování a testování. Tabulka 2.13 obsahuje data, která byla použita **pouze** pro prvotní adaptaci modelu Whisper, která je popsaná v kapitole Základní adaptace Whisperu.

Název datasetu	**Subsety**	**Počet nahrávek**	**Délka v minutách**
A-PiMod	–	105	00:13:29
HIWIRE	–	4032	03:18:19
MALORCA	–	6335	07:56:45
NATO	–	1373	02:52:20
ATCO2	EN	1579	03:00:08

Tabulka 2.11: Souhrn všech zpracovaných **trénovacích** datasetů

Název datasetu	Subsety	Počet nahrávek	Délka v minutách
HIWIRE	FR	800	00:32:05
	GR	400	00:28:41
	SP	300	00:13:06
MALORCA	–	1557	01:55:33
NATO	–	492	01:11:21
UWB	–	3899	03:44:25
ATCO2	EN Štefánik	53	00:10:29
	EN Zurich	412	00:49:56
	EN Ruzyně	70	00:11:58

Tabulka 2.12: Souhrn všech zpracovaných **testovacích** datasetů

Název datasetu	Subsety	Počet nahrávek	Délka v minutách
ATCO2	FR – trénovací s.	92	00:10:18
	FR – testovací s.	33	00:03:39
	Jiné jazyky – testovací s.	40	00:05:02

Tabulka 2.13: Souhrn všech zpracovaných **trénovacích a testovacích** datasetů použitých pro prvotní adaptaci modelu Whisper popsanou v kapitole Základní adaptace Whisperu.

2.10.1 Základní vyhodnocení datasetů – forma *true-case*

Následující část textu ukazuje v tabulkách 2.14 a 2.15 výsledky čistého (neboli Vanilla) modelu Whisper na všech datasetech s přepisy v plné i zkrácené podobě. Data měla formu true-case. Testovacím prostředím byla služba Google Colab.

Vyhodnocení každé sady trvalo přibližně dvě a čtvrt hodiny a v trénovacím skriptu, který byl pouhou modifikací skriptu pro trénování, byl využit hyperparametr `per_device_eval_batch_size=16`.

Z tabulky 2.14 s výsledky evaluace pro plné přepisy lze pozorovat, že WER hodnoty se pohybují na anglických datech – tedy všech kromě subsetů *FR* a *Jiné jazyky* – na nižších hodnotách (kolem 80 %) než na zmíněných dvou cizojazyčných datasetech. To by mohlo být způsobeno nastavením modelu na angličtinu.

<table>
<tr><th>ATCO2</th><th>HIWIRE</th><th>MALORCA</th><th>NATO</th><th>UWB</th></tr>
<tr><td>Ruzyně: 76.62</td><td>FR: 81.34</td><td rowspan="5">86.27</td><td rowspan="5">81.66</td><td rowspan="5">89.89</td></tr>
<tr><td>Štefánik: 75.73</td><td>GR: 82.01</td></tr>
<tr><td>Zurich: 80.81</td><td>SP: 83.99</td></tr>
<tr><td>FR: 96.69</td><td rowspan="2"></td></tr>
<tr><td>Jiné jazyky: 107.49</td></tr>
</table>

Tabulka 2.14: Hodnoty WER (%) na modelu Vanilla Whisper Medium pro každý dataset. Použité transkripty měly formu plných přepisů.

V tabulce 2.15 s vyhodnocením pro zkrácenou formu přepisů se dle očekávání pohybují hodnoty WER výš než u plných přepisů.

<table>
<tr><th>ATCO2</th><th>HIWIRE</th><th>MALORCA</th><th>NATO</th><th>UWB</th></tr>
<tr><td>Ruzyně: 86.82</td><td>FR: 84.15</td><td rowspan="5">92.57</td><td rowspan="5">104.73</td><td rowspan="5">94.86</td></tr>
<tr><td>Štefánik: 83.35</td><td>GR: 88.08</td></tr>
<tr><td>Zurich: 86.82</td><td>SP: 85.69</td></tr>
<tr><td>FR: 102.51</td><td rowspan="2"></td></tr>
<tr><td>Jiné jazyky: 126.43</td></tr>
</table>

Tabulka 2.15: Hodnoty WER (%) na modelu Vanilla Whisper Medium pro každý dataset. Použité transkripty měly formu zkrácených přepisů.

Kapitola 3

Whisper a použité prostředí, neuronové sítě a transformery

Tato kapitola vychází především z informací obsažených v knize [5], článku, napsaného tvůrci modelu Whisper [9], a studii zabývající se testováním modelu Whisper s promptem [3]. Text je zaměřen na ASR (Automatic Speech Recognition) model Whisper. Nachází se zde také stručný popis fungování neuronových sítí a transformerů.

3.1 Model Whisper

Whisper je model pro automatické rozpoznávání řeči (ASR), který využívá architekturu typu transformer. Byl natrénován na 680 tisících hodinách zvukových nahrávek s textovými přepisy. Trénování proběhlo v režimu učení s učitelem (supervised learning), přičemž část dat byla získána metodou slabého dohledu (weak supervision). To znamená, že přepisy nebyly vždy vytvořeny ručně člověkem — část jich pocházela z automatizovaného zpracování, čímž se snížila jejich kvalita, ale zvýšil se celkový objem.

Během tréninku bylo použito standardní autoregresivní učení s tzv. teacher forcing, kdy dekodér při predikci dalšího tokenu[1] vždy jako vstup dostává skutečný předchozí token [1].

Kromě klasického přepisu řeči do textu byl model trénován i na další úlohy, jako je detekce jazyka nebo překlad řeči do angličtiny, což z něj činí víceúčelový model. Ze všech trénovacích dat bylo 170 tisíc hodin ve vícejazyčných variantách, což umožňuje použít model pro rozpoznávání řeči v 96 různých jazycích. Dalších 125 tisíc hodin tvořily nahrávky s překlady do angličtiny.

3.1.1 Struktura

Model Whisper je *sequence-to-sequence* model, což znamená, že převádí řetězec audia přímo na výstupní řetězec tokenů. Dále je založen na takzvané **encoder-decoder** architektuře. To se projevuje jeho prací se vstupním signálem tak, jak je popsáno níže v postupu práce modelu Whisper:

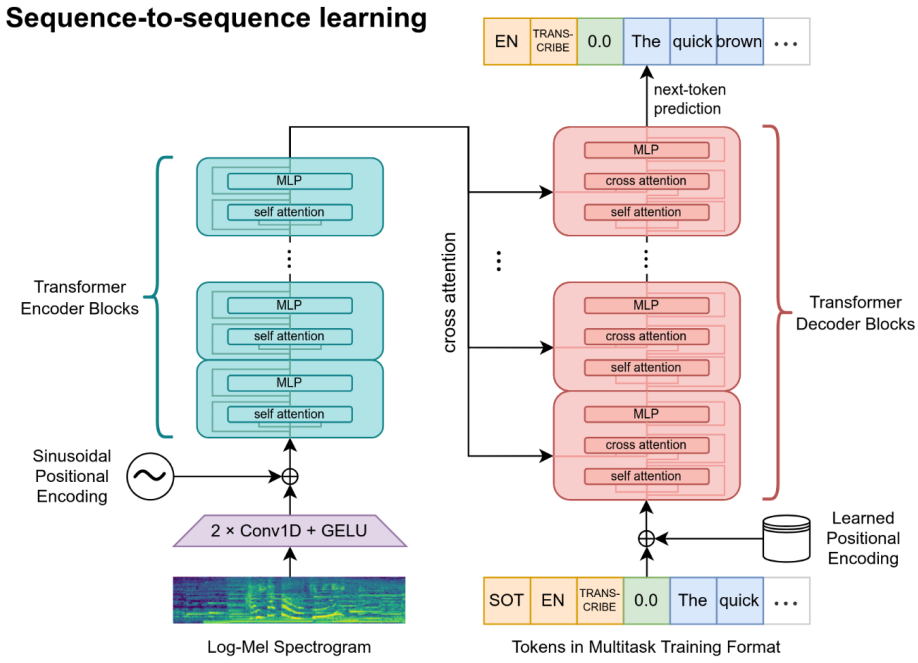
1. Surová audio data (tj. audio signály) jsou nejprve převedena na logaritmické Mel (log Mel) spektrogramy pomocí tzv. feature extractoru. Tento proces zahrnuje několik kroků [8] [10]:

[1] Tokenem rozumíme číselnou reprezentaci libovolného celku tvořeného čísly a písmeny. Nejčastěji se token chápe jako slovo.

- Vzorkování a normalizace: audio signál o maximální délce třicet vteřin je převeden na vzorkovací frekvenci 16 000 Hz a normalizován (normalizace je velmi důležitá pro správné fungování modelu)
- Rozdělení na rámce: signál je rozdělen na překrývající se rámce o délce 25 milisekund s krokem 10 milisekund.
- Výpočet Mel spektrogramu: pro každý rámec je vypočten Mel spektrogram s 80 frekvenčními kanály, který zachycuje množství zastoupení jednotlivých frekvencí v daném časovém úseku.
- Vytvoření log Mel spektrogramu: hodnoty Mel spektrogramu jsou převedeny do logaritmického měřítka, což odpovídá lidskému vnímání zvuku.

2. Log Mel spektrogram je převeden na sekvenci vnitřních stavů enkodéru.
3. Přes takzvanou *cross-attention* vrstvu je spojen výstup enkodéru a dekodér.
4. Dekodér *autoregresivně* predikuje výstupní tokeny, což znamená, že dochází k predikci pouze jednoho tokenu jak na základě aktuálního stavu dekodéru, tak na tokenech vygenerovaných v předchozích krocích. V této fázi je získán řetězec čísel, například *[4545, 7985, 2354]*.
5. Tokenizer potom dekóduje vygenerované tokeny na slova.

Názorně je architektura ukázána na obrázku 3.1.



Obrázek 3.1: Architektura modelu Whisper, převzato z [9]

Architektura modelu Whisper odpovídá konceptu **deep fusion**, tedy zapojení jazykového modelu přímo do architektury dekodéru. Díky tomu je možné trénování **end-to-end** způsobem – celý proces od vstupu po výstup se optimalizuje najednou pomocí jediné ztrátové funkce. Model generuje tokeny postupně pomocí dekodéru, přičemž každý token je následně

převeden na text pomocí tokenizéru. Tato úzká integrace jazykového modelu zajišťuje vyšší flexibilitu a obvykle i lepší výsledky ve srovnání s přístupy jako je *shallow fusion*, kde je jazykový model napojen až při inferenci[2].

Při trénování (a stejně tak při případném doladění) byla jako ztrátová funkce použita tzv. *Cross Entropy*.

Whisper je specifický tím, že nevyžaduje použití tzv. *attention masky*[3]. Tyto masky jsou běžně potřeba u jiných modelů k určení, která část vstupu je pouze doplněná (tzv. *padding*) a nemá být brána v potaz při výpočtu. Whisper naproti tomu očekává vstupy délky maximálně třicet vteřin – pokud jsou kratší, doplní se nulami (tichými úseky), pokud jsou delší, oříznou se. Model je schopný případné doplnění rozpoznat jako ticho a automaticky ho ignorovat.

3.1.2 Prompt

Tato sekce vychází především z článku popisující testování promptu na modelu Whisper [14]. Prompt je technika, při které je modelu předán rozšiřující kontext, přičemž se očekává zlepšení jeho výkonnosti. Prompt sám o sobě figuruje nejen u modelu Whisper, ale především v jazykových modelech, například v dnes velmi populárním modelu ChatGPT. Zde má ovšem formu informace, podle které se řídí obsahová forma celého výstupu. Tedy ChatGPT reaguje na prompt následovně (citace modelu[4]):

Prompt: Co je to ATC v kontextu letadel?

ChatGPT: ATC znamená *Air Traffic Control*, tedy řízení letového provozu. Jedná se o službu zajišťovanou pozemními dispečery, jejichž úkolem je navádět letadla během letu i na zemi, aby se předešlo kolizím a aby byl zajištěn plynulý a bezpečný provoz ve vzdušném prostoru. ATC komunikuje s piloty prostřednictvím rádiové komunikace a dává jim instrukce ohledně výšky, směru, rychlosti nebo přistání. Je to klíčová součást civilního i vojenského letectví.

Další text se bude zabývat problematikou použití promptu v kontextu modelu Whisper.

Dosavadní výzkum promptu

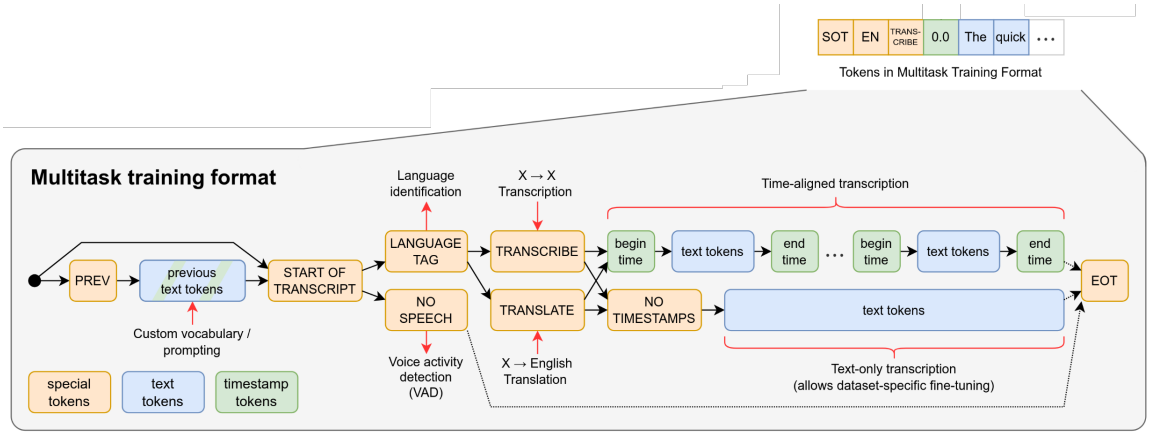
Mnoho prací se zabývalo využíváním promptu pro zvýšení výkonnosti modelu Whisper, ať se jednalo o změnu stylu přepisu (např. z tradiční čínštiny na jednoduchou), přepis textu písně či rozpoznání dialektu.

Zároveň se ukazuje, že i prompt s nesprávnými, či poškozenými informacemi může mít pozitivní dopad na kvalitu přepisu. To vede k přesvědčení, že v otázce porozumění promptu modelem Whisper není k dispozici jednoznačná odpověď, protože zlepšení jeho výkonnosti nutně neimplikuje porozumění promptu (pokud by kvalita a správnost promptu úměrně ovlivňovala výstup modelu, pak by se s největší pravděpodobností jednalo o situaci, kdy by skutečně Whisper rozuměl promptu).

[2] Inferencí modelu rozumíme jeho použití na nových datech.

[3] Při trénování i testování model opakovaně vypisoval upozornění ohledně chybějící attention masky, které varovalo před možnou nepřesností výstupu. I když byla maska nakonec předána, v kódu bylo možné ověřit, že model ji nepoužívá k maskování vstupních dat. Její účel byl pouze v možnosti augmentace, která byla využita jen při zapnutí této funkce v konfiguraci modelu. V této práci k použití attention masky nedošlo.

[4] https://chatgpt.com



Obrázek 3.2: Skladba vstupu dekodéru modelu Whisper, převzato z [9].

Vstup dekodéru

Pro Whisper model má prompt formu doplňujících informací, které vstupují přímo do dekodéru [2]. Jak je ukázáno na obrázku 3.2, dekodér Whisperu akceptuje tři hlavní skupiny tokenů. Ta první je označena jako *previous text tokens* a odpovídá těm tokenům, které zastupují prompt. Druhá dává modelu údaje o požadované činnosti (transcribe, translate,...) a jazyku, týkající se vstupního audia. Třetí předaná informace, využívaná hlavně při trénování, obsahuje přepis vstupních dat, který se má model za úkol naučit. K tomuto přepisu je možné přidat tokeny časových značek, které napomáhají modelu přesně určit místa v audio nahrávce, ke kterým se jednotlivé přepisy vážou.

Příklad vstupu dekodéru

Vstupní sekvence tokenů při trénování a využití promptu tedy může mít následující formu:

```
<|startofprev|> air france one zero eight zulu
<|startoftranscript|><|notimestamps|><|en|><|transcribe|>
air france one zero eight zulu standby now descend
to descend flight level seven zero air france one zero eight zulu
<|endoftext|>
```

V ní token `<|startofprev|>` uvozuje prompt, sekvence tokenů

```
<|startoftranscript|><|notimestamps|><|en|><|transcribe|>
```

označuje začínající přepis a určuje modelu způsob práce se vstupem, přičemž následují tokeny chtěného přepisu dat, zakončené tokenem `<|endoftext|>`, který celý řetězec uzavírá. Takto vytvořený blok dat vstupuje do dekodéru, kde je zpracováván.

Prompt a prefix

Podle oficiálního příspěvku na stránce GitHub[5], týkající se přímo modelu Whisper, je k dispozici vysvětlení týkající se **promptu** a **prefixu**:

[5] https://github.com/openai/whisper/discussions/117

- **prompt** – Jedná se o část označenou v obrázku 3.2 jako *previous text tokens*. Její první možnou úlohou je poskytnout modelu kontext z dříve přepsaných dat. Whisper může najednou akceptovat pouze třicetivteřinové audio, a proto je umožněno při přepisech delšího audia poskytnout kontext daného segmentu, čímž můžeme rozumět přímo přepsaná data v předchozím segmentu.

  Druhým využitím může být předání slov v promptu, která pomohou modelu dosáhnout požadovaného stylu přepisu.

- **prefix** – Tato část se vztahuje k oblasti označené na obrázku 3.2 jako *time-aligned transcription*. Autor příspěvku na GitHubu upozorňuje na možnost využití prefixu například při živém přepisu delších mluvených projevů – typicky při televizních rozhovorech, reportážích nebo konferencích.

  Při živém přepisu se očekává, že model bude zpracovávat zvuk po částečně se překrývajících segmentech. Aby byl zachován kontext mezi jednotlivými částmi, předává se dekodéru kromě aktuálního zvukového segmentu také přepis předchozího segmentu, a to včetně časových značek. Díky tomu může model lépe navázat na předchozí text a udržet konzistenci v průběhu přepisu.

V této práci se využívá promptu jako přídavné informace, která obsahuje údaje, které se v přepisovaném audiu vyskytují, a tím dosáhnout kvalitnějšího přepisu. Jedná se o prompty, které obsahují volací znaky jednotlivých letadel. Příprava těchto promptů byla popsána v kapitole Příprava dat (2.9.1).

3.1.3 Verze Whisperu

Whisper má více verzí (vytvořených jeho zakladateli a popsaných v [9]), které se liší počtem parametrů, což je důsledkem různého počtu hlav, vrstev a jejich šířek. Ze všech vytvořených verzí je nejpoužívanější *Medium* se 769 miliony parametrů. Verze Small (a dále všechny menší) je se svými 244 miliony parametrů příliš malá pro lepší výsledky, což lze číst v článku popisující model Whisper [9]. Verze Large je pak dvojnásobná počtem parametrů (1550 milionu) proti Medium, ale její výsledky, kterých dosáhla na různých datasetech ať už v přepisu nebo překladu, již nejsou o mnoho lepší. Proto je v této práci pro finetuning[6] použitá zmíněná verze Medium. Její trénování proběhlo také v jiné práci na podobné téma, která rovněž vznikla na fakultě FIT VUT [7].

3.2 Neuronové sítě

Velmi používanou variantou neuronových sítí je takzvaná dopředná neuronová síť, kde figuruje $1 \dots n$ po sobě jdoucích vrstev, kde každá má $1 \dots n$ neuronů. Každý neuron každé vrstvy je propojen s každým neuronem z vrstvy předcházející i následující pomocí hran, které mají váhy. Data jsou následně vkládána na první vrstvu a procházejí celou sítí. Trénování probíhá skrz upravování vah podle požadované formy výstupu neuronové sítě. Je možné využít variabilní algoritmy pro učení sítí – společný základ většiny z nich spočívá v matematické aplikaci gradientního sestupu (Gradient Descent). S dostatečným množstvím neuronů a díky aktivačním funkcím každé vrstvy, které jsou zpravidla funkcemi nelineárními, je možné síť učit velmi složité vzory.

[6] Pojem finetuning chápeme jako adaptaci již natrénovaného modelu na specifická data.

Výše zmíněná architektura a princip učení, takzvaný *backpropagation*, je využit u řady forem neuronových sítí, mezi které patří mimo jiné:

- hluboké (DNN – Deep Neural Networks) – mají mnoho vrstev neuronů.
- konvoluční (CNN – Convolutional Neural Networks) – jejich využití se naskýtá zpravidla při zpracování obrazu. Základním stavebním kamenem je konvoluční jádro, díky němuž je možné zpracovávat více hodnot ze vstupu zároveň.
- rekurentní (RNN – Recurrent Neural Networks) – pro tyto sítě je charakteristická zpětná vazba mezi neurony jednotlivých vrstev.
- transformery – vycházejí z RNN.

Existuje mnoho dalších modifikací a variant neuronových sítí, ty ale nejsou pro tuto práci podstatné.

3.2.1 Loss funkce

Pro úspěšné trénování neuronové sítě je nutné použít vhodnou ztrátovou funkci, která bude reflektovat odchylku mezi výstupem modelu a cílovými hodnotami v kontextu účelu dané neuronové sítě, a umožní tak efektivní optimalizaci vah pomocí zpětné propagace (tzv. *backpropagation*). Různé loss funkce jsou vhodné pro různé účely, přičemž mezi známé patří:

- Střední kvadratická chyba (MSE – Mean Square Error) – vhodné pro regresní účely,
- Křížová entropie (Cross-Entropy) – vhodná ke klasifikaci (například obrázků). Měří odlišnost predikovaného a referenčního výstupu sítě. V této práci tato funkce figuruje jako loss pro model Whisper. Obecně je křížová entropie vhodná a široce používaná při trénování *sequence-to-sequence* modelů na klasifikačních úlohách [3]. Její vzorec 3.1 vypadá následovně:

$$\text{Cross-Entropy Loss} = - - \sum_{t} y_t \log(\hat{y}_t) \tag{3.1}$$

  kde

  - $\hat{y}_t$ představuje referenční třídu, do které měla být data (či *token*, v případě této práce) klasifikována
  - y_t je predikovaná pravděpodobnost klasifikace dat do třídy t

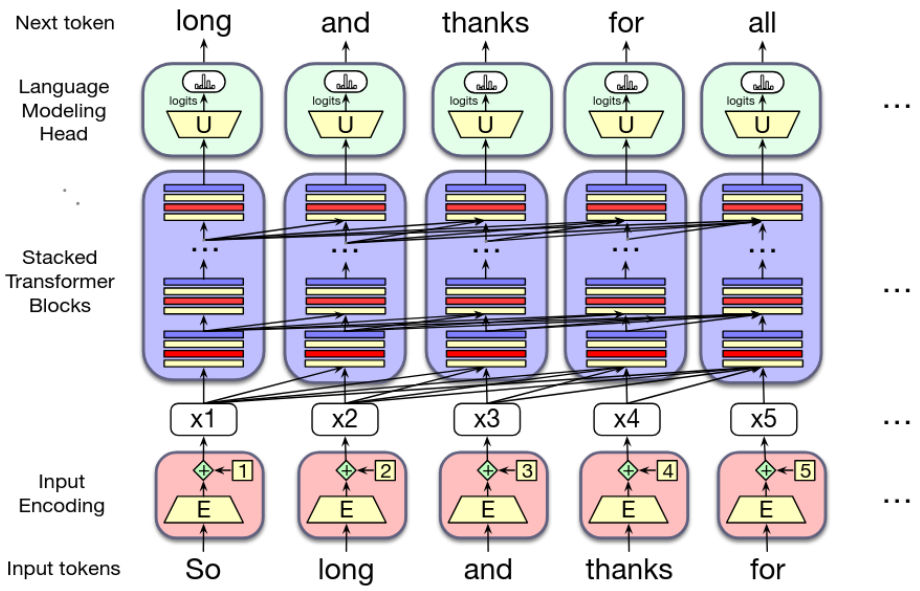
3.3 Transformery

Sekce vychází především z článku od tvůrců architektury transformer [12] a knihy [5]. V této sekci jsou obecně popsány transformery a princip jejich práce v jazykových a ASR modelech.

Nyní se vyjádříme k základním stavebním prvkům modelu Whisper. Jedná se o encoder-decoder architekturu. Tu je možné implementovat různými přístupy, zpravidla se však používají rekurentní (RNN) sítě nebo transformery – na těch je vystavěn model Whisper.

Transformer definujeme jako neuronovou síť se specifickou strukturou, kterou lze přirovnat k rekurentním sítím (RNN), které se vyznačují zpětnými vazbami mezi neurony

jednotlivých vrstev (v kontrastu k dopředným sítím). Jeho struktura je založená především na mechanismech *self-attention* a *multi-head attention*. Na obrázku 3.3 pozorujeme obecnou stavbu transfomeru, která je složena z několika hlavních komponent: **bloků**, **input encoding** komponenty, **positional encoding** mechanismu a **language modeling head** (dále *hlava*).



Obrázek 3.3: Obecná architektura transformeru [5].

Práce probíhá následovně: vstupní řetězec je převeden na posloupnost tokenů. Každý z nich je potom díky input encoding komponentě převeden na embedding vektor (za použití matice E), který následně vstupuje do posloupnosti bloků transformeru. K tomuto vektoru se přidává také výsledek positional encoding mechanismu, který zajišťuje využití informace o pozici (absolutní nebo relativní) daného tokenu ve skupině zpracovávaných tokenů.

Každý blok mapuje vstupní vektor x_i, odpovídající sloupci i, na výstupní vektor h_i. V bloku, kterých je v jednom sloupci obvykle dvanáct a více, se nachází čtyři vrstvy: první normalizační, multi-head attention, druhé normalizační a následující plně propojená.

Multi-head attention mechanismus je složen z více self-attention mechanismů. V každém z nich figuruje trojice matic:

- **Query** matice W^Q – zjišťuje vztah aktuálního elementu k předešlým vstupům (tzv. *queries*).
- **Key** matice W^K – udává podobnost předešlých vstupů k aktuálnímu elementu (tzv. *keys*).
- **Value** matice W^V, – váhuje a sčítá hodnoty výstupů předcházející vrstvy bloků pro výpočet výstupu aktuálního prvku (tzv. *values*).

Celý self-attention proces vedený za účelem výpočtu jednoho self-attention výstupu a_i s jedním vstupem x_i lze popsat následujícími rovnicemi:

$$\mathbf{q}_i = \mathbf{x}_i \mathbf{W}^Q \tag{3.2}$$

$$\mathbf{k}_j = \mathbf{x}_j \mathbf{W}^K \tag{3.3}$$

$$\mathbf{v}_j = \mathbf{x}_j \mathbf{W}^V \tag{3.4}$$

$$\text{score}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \tag{3.5}$$

$$\alpha_{ij} = \text{softmax}\left(\text{score}(\mathbf{x}_i, \mathbf{x}_j)\right) \quad \forall j \leq i \tag{3.6}$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j \tag{3.7}$$

Matice jsou během trénování upravovány tak, aby se jednotlivé části modelu zaměřovaly na různé typy informací nebo vzory ve vstupu. Multi-head attention mechanismus vede k postupnému zachytávání různých aspektů kontextu daného vstupu, jehož celkové pochopení je tak zpřesňováno každým blokem.

Za každým sloupcem se nachází *language modeling head* – hlava, která zpracovává embedding vektor h_N^L, který je výstupem posledního bloku N-tého sloupce. Na něj aplikuje *unembedding* matici $U = E^T$, čímž získá *logit* vektor u, který obsahuje skóre pro každý záznam v použitém slovníku (máme na mysli slovník obsahující mapování tokenů na jednotlivé celky písmen a číslic). Z tohoto logit vektoru je získána distribuce pravděpodobností pro celý slovník použitím funkce *softmax* (proces je popsán rovnicemi 3.8).

$$\mathbf{u} = \mathbf{h}_N^L \mathbf{E}^T \tag{3.8}$$

$$\mathbf{y} = softmax(\mathbf{u}) \tag{3.9}$$

$$softmax(\mathbf{u}_j) = \frac{e^{\mathbf{u}_j}}{\sum_i e^{\mathbf{u}_i}} \tag{3.10}$$

Způsob, jakým je při generování vybíráno další slovo ze slovníku, může být tzv. *greedy* – tedy vždy se zvolí to slovo, které má aktuálně nejvyšší pravděpodobnost. Tento přístup je jednoduchý, ale může vést k suboptimálním výsledkům, protože volba slova na pozici n se může později ukázat jako nevhodná při výběru slova na pozici $n + 1$. Tento problém řeší tzv. *beam search* (paprskové hledání), který místo jediné možnosti sleduje zároveň několik nejpravděpodobnějších větví výběru. V každém kroku si model uchová k nejlepších částečně vygenerovaných sekvencí (kde k je tzv. šířka paprsku) a během zpracovávání vstupu je rozšiřuje. Díky tomu může později zvolit takovou cestu, která vede k celkově nejpravděpodobnějšímu výstupu, a to i v případě, že některý krok této cesty neměl v daný moment nejvyšší pravděpodobnost.

3.4 Encoder-decoder architektura Whisperu

Tato architektura modelů, na které je postaven i model Whisper, vyžaduje při trénování množinu dvojic ve formátu vstup–referenční výstup. Enkodér zpracuje vstupní data x a na výstupu poskytne kontextovou reprezentaci h. Struktura enkodéru odpovídá klasickému transformeru popsanému v předchozí sekci. Dekodér následně pracuje s kontextem h a postupně generuje výstupní sekvenci y. Je konstruován podobně jako běžný transformer, ale

navíc obsahuje **cross-attention** vrstvu. Ta má stejnou strukturu jako multi-head attention, avšak s tím rozdílem, že *keys* a *values* pocházejí z výstupu enkodéru, zatímco *queries* jsou počítány z předchozí vrstvy dekodéru.

Na rozdíl od jazykových modelů, které pracují s textem jak na vstupu, tak na výstupu, ASR modely jako Whisper mají na vstupu log Mel spektrogram (obecně však ASR modely mohou přijímat i surové audio, spektrum, ap.). Ze spektrogramu se získá posloupnost t **feature-vektorů**, které vstupují dále do enkodéru (tedy, v případě modelu Whisper, do jednotlivých bloků transformeru).

3.5 Prostředí

Trénování a testování modelu Whisper vyžadovalo mít připravené prostředí ve formě skriptů, které odpovídaly za správný průběh trénování a testování. V prvotní fázi trénování, popsané v kapitole Základní adaptace Whisperu, bylo použito prostředí mírně odlišné od zbylé práce především kvůli použití služby Google Colab. V další fázi (popsané v kapitolách Finetuning I.,II.,III.) byly původní skripty značně modifikovány a zcela nově byl vytvořen skript pro testování.

3.5.1 Prostředí pro základní adaptaci Whisperu

Trénování modelu Whisper vyžaduje silný výpočetní výkon, a proto v úvahu přicházela na prvním místě služba Google Colab[7]. Ta je dostupná zdarma, ovšem výpočetní doba je omezena na 4 hodiny, což značně omezuje možnosti trénování. Druhou nevýhodou je potom omezená paměť RAM, v tomto případě s limitem 15 GB. Službu Google Colab bylo rozhodnuto použít pro přípravu, zjišťování funkčnosti trénovacích skriptů a prvotní trénování.

Základ trénování tvořil volně dostupný skript[8] z webu Huggingface[9]. Hlavní části skriptu jsou založeny na knihovnách **datasets** a **transformers**.

V trénovacím skriptu figurují následující kroky:

1. Načtení dat – ta jsou uložená po jednotlivých datasetech v *zip* souboru. K jejich načtení je použita jednořádková funkce `load_from_disk()`.

2. Načtení modelu – jako model je použit **Vanilla Whisper Medium**. Z něj jsou také načteny zbylé dva nástroje **tokenizer** a **feature extractor**. V této fázi může být načten i **checkpoint** z předešlého trénování. Toho je, v případě použití služby Google Colab, hojně využíváno především kvůli tomu, že většina datasetů je příliš velkých, než aby na nich mohl být model natrénován při rozumném počtu epoch a omezeném času najednou.

3. Příprava dat – v této fázi jsou získány *tokeny* z přepisů audia, a WAV soubory jsou převedeny do vstupní podoby pro model. Poté je ze všech dat, zamýšlených k trénování a testování, vytvořen jeden celek obsahující jednu trénovací a jednu testovací sadu.

4. Vytvoření třídy s názvem **DataCollatorSpeechSeq2SeqWithPadding**. Instance této třídy je následně použita instancí objektu **Trainer**. Jejím stěžejním úkolem je převedení získaných tokenů na pole o požadované délce přidáním *padding* hodnot -100.

[7] https://colab.research.google.com/
[8] https://huggingface.co/blog/fine-tune-whisper
[9] https://huggingface.co/

Tyto hodnoty zajišťují, že při výpočtu loss funkce nebudou zahrnuty, což je nezbytné pro správný výpočet.

5. Vytvoření funkce **compute_metrics** – slouží pro evaluaci modelu. V tomto skriptu je zásadně používaná metrika WER.

6. Definice argumentů pro trénování – v rámci třídy **Seq2SeqTrainingArguments** lze manipulovat se všemi nezbytnými argumenty.

7. Vytvoření instance třídy **Seq2SeqTrainer** s argumenty zadefinovanými dříve a spuštění trénování. Ve fázi spouštění trénování může být přidán parametr `resume_from_checkpoint`, který zajistí korektní pokračování tréninku modelu od místa, ve kterém byl uložen *checkpoint*.

3.5.2 Prostředí pro kapitoly Finetuning I.,II.,III.

Pro trénování popsaném v kapitolách Finetuning I.,II. a III. již byl využit školní výpočetní cluster. Všechny experimenty měly vždy alokovány zdroje s grafickou kartou o kapacitě paměti RAM 16GB, čemuž byl i podřízen hyperparametr *batch size*. Díky vysoké kapacitě dočasného úložiště bylo možné nechat si ukládat *checkpointy* (stav natrénovaného modelu po každé epoše) během trénování. To umožnilo provést testy až po celkovém natrénování na daný počet epoch a vybírání těch checkpointů, u kterých bylo dosaženo nejlepších výsledků (nejnižších hodnot WER).

Skripty

Z původního trénovacího skriptu ve formě python notebook byl vytvořen nový trénovací skript, který usnadňoval načítání a parsování dat pro trénování. Kroky jeho práce však zůstaly v zásadě stejné, jako tomu bylo v předcházející sekci. Zároveň skrz množství přepínačů bylo možné snadno určovat parametry trénování jako typ přepisů (plný nebo zkrácený), využití promptu (uplatnění v následující kapitole) atd.

Testovací skript poskytl obdobná usnadnění jako trénovací skript. Přidána byla především funkce testování WER hodnot volacích znaků v přepisech, která byla nezbytná pro umožnění vyhodnocení dopadu promptu. Průběh testování přepsaných volacích znaků probíhal následujícím způsobem, zapsaným algoritmicky:

1. Získej přepis audia modelem a získej referenční volací znaky obsažené v dané nahrávce.

2. Postupně pro každý referenční volací znak, obsažený v nahrávce:

   (a) Posouvej *okno* o délce počtu slov referenčního volacího znaku po přepisu (posunutí okna probíhalo po jednotlivých slovech).

   (b) Pro každý posun si zaznamenej WER hodnotu, kde referencí je volací znak, a predikcí je aktuální sekvence slov v přepisu, na kterých se nachází okno o délce počtu slov referenčního volacího znaku.

   (c) Pokud se okno nachází na konci přepisu, pokračuj, jinak posuň okno dál a opakuj krok 1.1

   (d) Seřaď získané WER hodnoty od nejmenší po největší

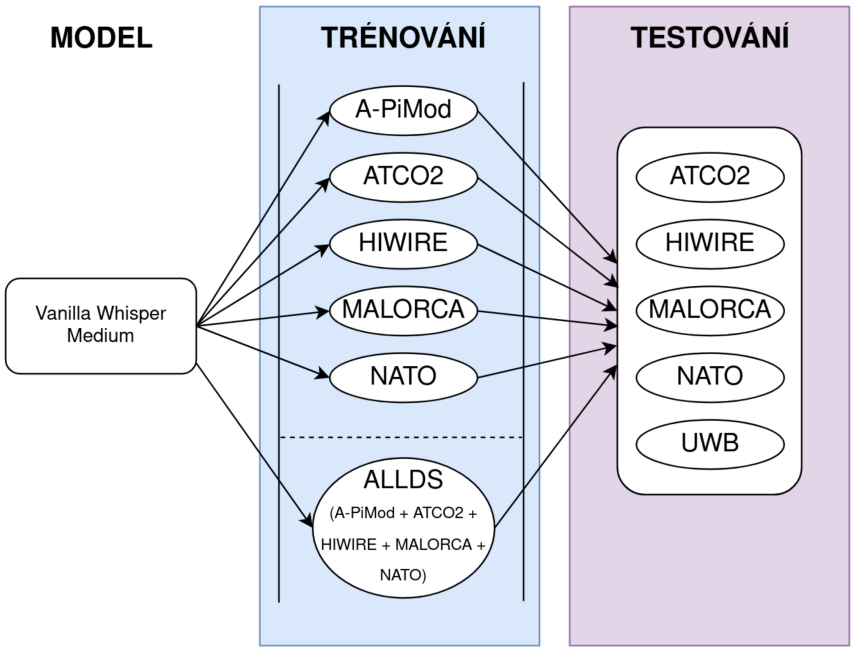
   (e) Vrať tolik WER hodnot od počátku pole s WER hodnotami, kolikrát se měl daný volací znak vyskytovat v promluvě.

(f) Vrať počet nul v poli s WER hodnotami jakožto číslo, které značí počet zcela správně přepsaných volacích znaků.

Jako pomocný byl navíc sestaven vizualizační skript, skrze který bylo možné nechat přepsat daný dataset zvoleným modelem. Výstupem skriptu byl potom soubor, ve kterém byly pod sebou zobrazeny dvojice přepisů vygenerovaný-referenční. Skript měl sloužit pro částečné ověření, že WER hodnoty, získané během různých evaluací, skutečně odpovídají kvalitě přepisu.

3.6 Schéma a popis trénování

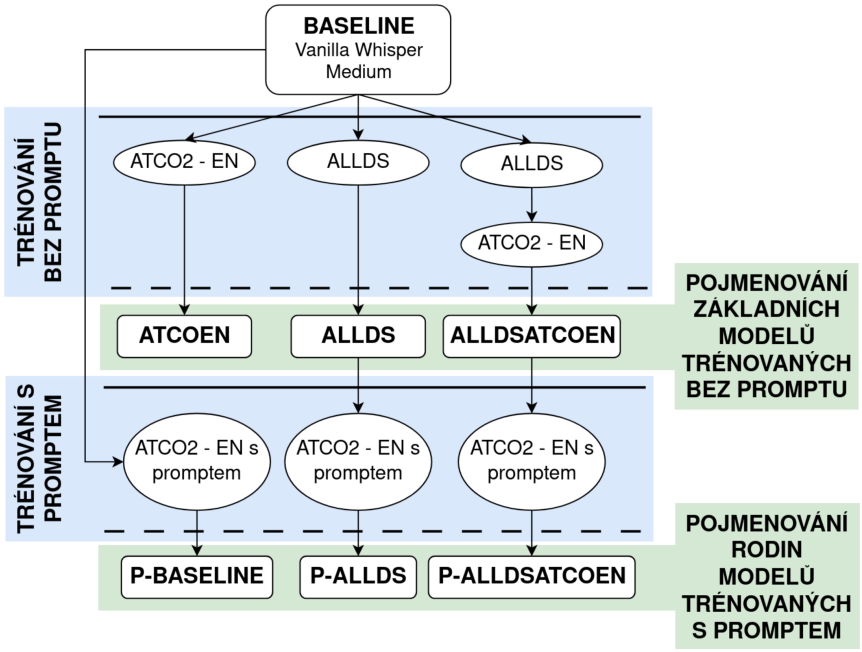
V kapitole *Základní adaptace Whisperu* proběhlo trénování a testování na prvotní upravené podobě trénovacích i testovacích datasetů s přepisy ve formě *true-case*. Prompt zde nebyl vůbec zapojen a výstupem této kapitoly je srovnání jednotlivých dostupných datových sad. Na diagramu 3.4 je ukázán postup trénování a testování, přičemž šipky, které vedou od modelu k trénovacím datasetům zároveň označují jednotlivé natrénované modely (tzn. Vanilla Whisper Medium byl natrénován samostatně na datasetu ATCO2 a potom otestován na dané množině testovacích sad).



Obrázek 3.4: Digram průběhu základní adaptace Whisperu, která je popsána v kapitole Základní adaptace Whisperu.

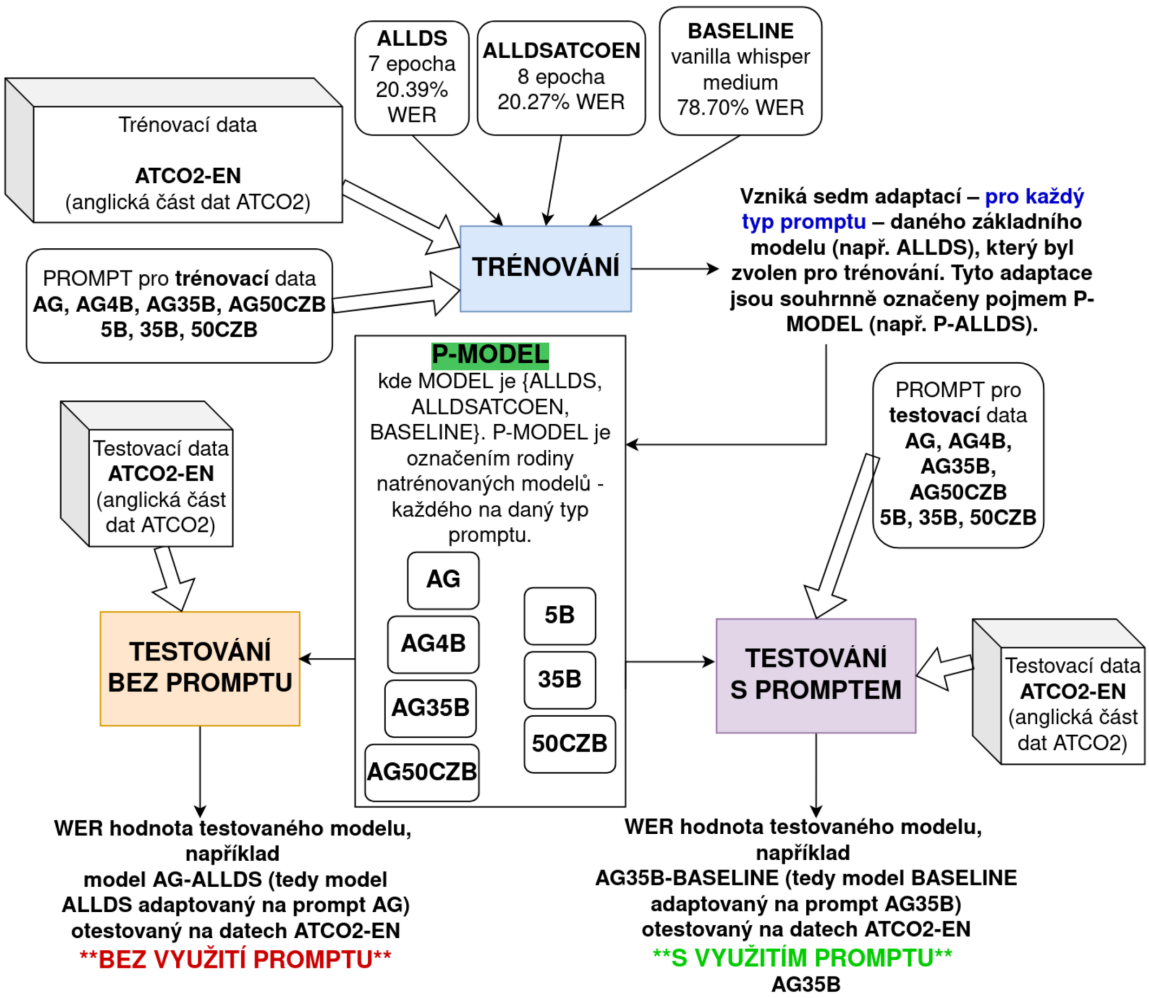
V kapitole *Finetuning I.* bylo provedeno trénování modelů bez použití promptu. Pod označením ATCO2-EN rozumíme anglickou část datasetu ATCO2, na kterou je zaměřeno celé následné trénování s promptem. Název ALLDS odpovídá označení *všech* zpracovaných datasetů, tedy A-PiMod, ATCO2-EN (zde opět pouze anglická část původních dat ATCO2), HIWIRE, MALORCA a NATO. Všechny přepisy měly formu *lower-case*. Jednotlivé získané modely pak byly použity pro následující trénování s promptem (kapitola *Finetuning II.*. Modely se lišily podle použitých datových sad, podle kterých byly potom

výsledné modely pojmenovány. Následující diagram 3.5 objasňuje použití datových sad a pojmenování natrénovaných modelů, které je dále objasněné v popisu obrázku.



Obrázek 3.5: Diagram trénování a pojmenování modelů, které je popsané v kapitolách Finetuning I.,II. a III. Pojmenovaní P-ALLDS, P-BASELINE a P-ALLDSATCOEN označují rodiny modelů se stejným výchozím modelem, trénovaným bez promptu. Každý model v dané rodině vychází ze stejného původního modelu s následnou adaptací na daný typ promptu, tedy **P** značí prompt, na který byl daný model v rodině trénován – můžeme psát $P \in \{\texttt{AG,AG4B,AG35B,AG50CZB,5B,35B,50CZB}\}$.

V dalším diagramu 3.6 je upřesněno trénování a testování modelů na datasetech s promptem. Opět je názorně ukázáno použití názvu P-MODEL, který označuje rodinu modelů vycházejících ze stejného výchozího MODELU. Každý model v rodině je dále trénován na jiný typ promptu.



Obrázek 3.6: Diagram ukazující průběh trénování modelů na datech s promptem a jejich testování na datech s promptem i bez promptu.

Kapitola *Finetuning III.* se věnuje zkráceným přepisům na stejném principu, jaký je ukázán v předchozích dvou diagramech 3.5 a 3.6.

Kapitola 4

Základní adaptace Whisperu

Následující část textu popisuje prvotní fáze trénování modelu Whisper s dostupnými daty. Důležité je podotknout, že při trénování měla data formu *true case*, tedy byla přítomna velká a malá písmena. Rovněž zde nedocházelo k trénování pomocí *promptu* a nebyla vyhodnocována přesnost přepisu volacích znaků.

Jako trénovací prostředí byl zvolen Google Colab.

4.1 Volba hyperparametrů pro trénování

Prvním krokem pro trénování byla volba hyperparametrů. Většina pochází přímo z nastavení původního trénovacího skriptu (ze stránky *huggingface*). Níže je ukázáno nastavení, se kterým probíhalo trénování modelů v této práci. Následně jsou jednotlivé parametry vysvětleny.

```
per_device_train_batch_size=4,
gradient_accumulation_steps=4,
learning_rate=1e-5,
warmup_ratio=0.12,
gradient_checkpointing=True,
fp16=True,
save_strategy="epoch",
num_train_epochs=15,
logging_steps=30,
```

Velikost dávky (per_device_batch_size) na jedno GPU je nastavena na čtyři. Protože tato hodnota přímo úměrně zvyšuje náročnost na paměť GPU, nelze ji zvýšit, aniž by trénink mohl být v pořádku dokončen pro 15 GB GPU paměti. Aby bylo možné efektivně pracovat s většími dávkami, používá se akumulace gradientů přes čtyři kroky. Tímto způsobem se efektivní velikost dávky zvýší, aniž by byla překročena paměťová kapacita GPU. Poznamenejme, že při nižších hodnotách by docházelo k prodloužení trénovacího času. Také by mohlo dojít k nepřesnému odhadu gradientu, jehož hodnota by se pravděpodobně skokově měnila.

Rychlost učení je nastavena na $1e-5$, což je běžná hodnota pro *finetuning* velkých modelů. Stabilitu tréninku v počátečních fázích zajišťuje *warm-up* fáze, která tvoří 12 % z celkového počtu kroků. Tato hodnota byla převzata z bakalářské práce na FIT VUT [7], věnující se rovněž trénování modelu Whisper, kde bylo dosaženo dobrých výsledků.

Gradient checkpointing umožňuje snížit paměťové nároky při práci s velkými modely tím, že ukládá méně mezivýpočtů, což je klíčové pro práci s omezenou pamětí.

Pro trénování je dále zapnutá optimalizace pro GPU s podporou *FP16* (přesnost hodnot s datovým typem *16bit floating point*), což zrychluje výpočty, snižuje paměťovou náročnost a zvyšuje paměťovou propustnost. To vše v kontrastu s přesností *FP32*.

Na konci každé epochy se také ukládají kontrolní body, aby bylo možné tréninkové části zpětně vyhodnocovat. V parametrech není uvedena vyhodnocovací strategie, a to z důvodu, že se trénink při vyhodnocování po každé epoše stával časově neúnosně náročným. Vyhodnocování v této kapitole proběhlo pouze u zvolených epoch až po samotném trénování.

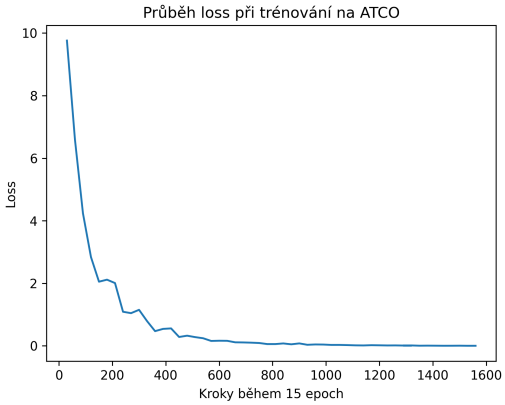
Pro podrobné sledování průběhu tréninku se logy generují každých třicet kroků, což poskytuje přehled o aktuálním stavu trénovací loss funkce, aniž by to příliš zatěžovalo proces. Jeden krok znamená zpracování jedné dávky.

Tyto parametry společně zajišťují efektivní využití dostupných zdrojů a optimalizaci výsledků modelu během tréninku.

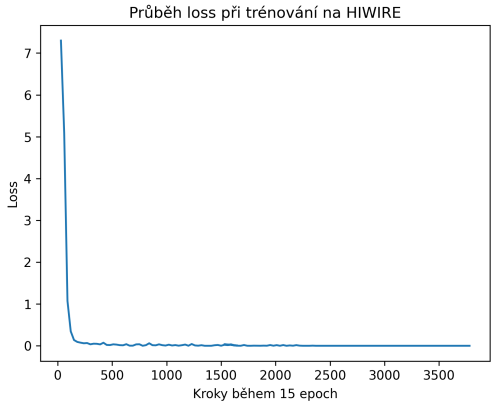
4.2 Vzájemná evaluace a trénování modelů na různých datasetech s plnými přepisy

V této fázi bylo rozhodnuto natrénovat Vanilla Whisper Medium na jednotlivých datasetech zvlášť s cílem ověřit, jak se budou vyvíjet hodnoty WER. Z testovacích i trénovacích sad byly použity pouze varianty s *plnými přepisy*.

Níže jsou ukázány průběhy dvou loss funkcí z trénování na datech ATCO2 4.1 a HIWIRE 4.2. Pozoruhodné je, jak rychle loss funkce klesá, přičemž u sady HIWIRE se loss funkce již mezi první a druhou epochou asymptoticky blíží hodnotě nula. U ostatních datasetů se tento jev neděje, hodnoty se ale přesto pohybují velmi blízko nuly. Vzhledem k tomu, že datasety (kromě A-PiMod) jsou relativně větší, lze tento jev přisuzovat spíš vysoké kapacitě a schopnosti modelu Whisper přizpůsobit se datům než přetrénovanosti.



Obrázek 4.1: Průběh loss funkce při trénování na datasetu ATCO2.



Obrázek 4.2: Průběh loss funkce při trénování na datasetu HIWIRE.

Vyhodnocení proběhlo na všech vytvořených testovacích sadách. V tabulce 4.1 lze pozorovat procentuální hodnoty WER pro jednotlivé modely natrénované na dané trénovací sadě. Šedě je vyznačena oblast pro datasety, na kterých proběhlo trénování i testování. Oranžově je zbarvena diagonála tam, kde proběhlo testování modelu natrénovaném na sadě

pocházející ze shodného datasetu jako testovací sada. Dle očekávání jsou na této diagonále hodnoty nízké, což poukazuje na fakt, že ačkoliv je testovací dataset vybrán z dostupných dat precizně, dochází k mnohem lepším výsledkům, pokud trénovací data náleží stejnému zdroji. Velmi pravděpodobně se v tomto případě bude projevovat nahrávací prostředí, opakující se slova, která jsou typická pro danou lokalitu (například označení přistávacích drah, označení příletových bodů,...) a nelze vyloučit i opakující se mluvčí.

Dále lze konstatovat, že nejlepšího výsledku celkově dosáhl model natrénovaný i otestovaný na datech sady HIWIRE, přičemž došlo i ke zlepšení u ostatních testovacích sad v porovnání s tabulkou 2.14 s hodnotami evaluace Vanilla Whisper Medium (vůči této tabulce uvažujeme v této kapitole veškerá zlepšení nebo zhoršení). Při testování došlo obecně ke zlepšení, kromě testovacích subsetů *Jiné jazyky* a *FR* z datasetu ATCO2. Zmíněné sady měly při trénování na datech MALORCA výsledky po řadě 95.04% (zde došlo pouze k nepatrnému zlepšení) a 117.51% (zhoršení o 9.3%) a na datech NATO výsledky po řadě 98.35% a 108.68% (zhoršení po řadě o 1.71% a 1.1%). Celkově nepříliš dobré výsledky na subsetu *Jiné jazyky* jsou očekávatelné, vzhledem k tomu, že kromě dat ve francouzském jazyce nejsou poskytnuta jiná cizojazyčná data. Velmi dobré hodnoty při trénování a testování na datech sady HIWIRE lze zase vysvětlit nízkou hladinou šumu v nahrávkách a snadnou srozumitelností řeči (při běžném poslechu lze rozumět zřetelně jednotlivá slova, což u jiných sad říci nelze).

Testovací sady		Modely podle trénovací sady				
		A-PiMod	**ATCO2**	**HIWIRE**	**MALORCA**	**NATO**
ATCO2	Ruzyně	43.88	29.50	65.59	60.74	75.
	Štefánik	40.98	25.46	65.38	60.81	72.48
	Zurich	49.12	28.29	69.41	65.44	73.20
	FR	74.26	47.98	93.01	95.04	98.35
	Jiné jazyky	79.34	86.83	101.80	117.51	108.68
HIWIRE	FR	38.61	39.40	0.08	40.57	29.01
	GR	41.24	41.40	0.48	40.92	28.50
	SP	42.41	39.60	0.94	39.71	32.74
MALORCA		53.43	54.29	40.93	5.38	48.86
NATO		58.29	53.17	37.65	38.84	12.09
UWB		49.29	50.95	44.03	38.64	53.55

Tabulka 4.1: WER hodnoty (%) pro modely natrénované na trénovacích datasetech A-PiMod, ATCO2, HIWIRE, MALORCA a NATO vyhodnocené na každém testovacím subsetu. Oranžově jsou zvýrazněné hodnoty WER tam, kde se shodoval dataset, ze kterého vycházela trénovací i testovací sada. Šedě je pak označeno pole hodnot, kde se nacházejí výsledky vzájemného testování čtyř datasetů (ATCO2, HIWIRE, MALORCA, NATO). Použité transkripty měly formu plných přepisů.

4.3 Natrénování modelu na všech datasetech s plnými přepisy

V této části byla provedena adaptace modelu na všech trénovacích datasetech (opět pouze pro plné přepisy). Model byl natrénován na osmi epochách. Vyšší počet epoch nebyl vyzkoušen, protože tato práce se na trénování bez promptu nezaměřuje. Navzdory očekávání došlo k velmi dobrým výsledkům – WER hodnoty všech testovacích sad, které byly získané při evaluaci natrénovaného modelu, předčily téměř vždy nejlepší výsledky v tabulce 4.1. V následující tabulce 4.2 můžeme pozorovat získané hodnoty WER pro jednotlivé testovací datasety.

<table>
<tr><th>ATCO2</th><th>HIWIRE</th><th>MALORCA</th><th>NATO</th><th>UWB</th></tr>
<tr><td>Ruzyně: 27.19</td><td>FR: 0.54</td><td rowspan="5">5.80</td><td rowspan="5">11.75</td><td rowspan="5">33.08</td></tr>
<tr><td>Štefánik: 25.00</td><td>GR: 1.35</td></tr>
<tr><td>Zurich: 26.43</td><td>SP: 1.04</td></tr>
<tr><td>FR: 51.29</td><td rowspan="2"></td></tr>
<tr><td>Jiné jazyky: 80.24</td></tr>
</table>

Tabulka 4.2: WER hodnoty (%) modelu natrénovaném na všech vytvořených trénovacích sadách (A-PiMod, ATCO2, HIWIRE, MALORCA,NATO) a vyhodnoceném na všech testovacích datových sadách zvlášť. Použité transkripty měly formu plných přepisů.

Výsledná WER hodnota na datasetu UWB ukazuje, že model hůře generalizuje na neviděná data. To potvrzuje rozdílnost kvality jednotlivých leteckých datasetů a poukazuje na obtížnost vytvořit obecný rozpoznávač letecké komunikace.

4.4 Závěr

V této kapitole byly provedeny první adaptace modelu Vanilla Whisper Medium na datech bez promptu. Všechny získané modely prošly testováním.

Trénink probíhal s daty ve formátu *true-case* a byl realizován na platformě Google Colab. Mezi důležité zvolené hyperparametry patřily velikost dávky (4), akumulace gradientů (přes 4 kroky), učící rychlost $1e-5$, warm-up fáze (12 %) a zapnutí FP16, což umožnilo efektivní využití omezené paměti.

Bylo provedeno samostatné trénování modelu na každé trénovací sadě s plnými přepisy. Evaluace následně ukázala, že nejnižších hodnot WER bylo dosaženo vždy, když testovací i trénovací data pocházela ze stejného datasetu – typicky na diagonále tabulky výsledků. Nejlepších výsledků dosáhl model trénovaný a testovaný na datech HIWIRE, kde například subset „FR“ vykázal pouhých 0.08 % WER, což je zásadní zlepšení oproti původnímu Vanilla modelu.

V porovnání s výchozím modelem Whisper Medium se modely **adaptované na jednotlivé datasety** zlepšily téměř ve všech případech. Konkrétně:

- HIWIRE (FR): Zlepšení o 38.53 % z 38.61 % na 0.08 %
- HIWIRE (GR): Zlepšení o 40.76 % z 41.24 % na 0.48 %
- MALORCA: Zlepšení o 48.05 % z 53.43 % na 5.38 %

- NATO: Zlepšení o 46.20 % z 58.29 % na 12.09 %

Výjimkou byly testovací subsety „FR“ a „Jiné jazyky“ z datasetu ATCO2, u kterých došlo ke zhoršení WER při trénování na datech MALORCA nebo NATO:

- ATCO2 (FR) – trénink na datasetu MALORCA: zhoršení o 30.68 % z 86.83 % na 117.51 %
- ATCO2 (FR) – trénink na datasetu NATO: zhoršení o 10.33 % z 98.35 % na 108.68 %

Tato zhoršení lze vysvětlit nedostatečným pokrytím cizojazyčných dat v trénovacích sadách. Naopak velmi nízké hodnoty WER na sadě HIWIRE svědčí o její vysoké kvalitě – dobře artikulované řeči a nízkém šumu.

Při vyhodnocení modelů trénovaných i testovaných na datech pocházejících ze stejného datasetu si nejhůře vedl dataset ATCO2, který zdaleka nedosáhl takových výsledků, jako tomu bylo například u dat MALORCA. Přitom obě skupiny dat pocházejí z reálného prostředí, což vede k ještě silnějšímu přesvědčení, že prostředí a kvalita snímaných dat má příliš velkou váhu při přepisování, než aby bylo možné mluvit o generalizaci přepisu napříč letišti.

Kapitola 5

Finetuning I. – Adaptace Whisperu na plné přepisy

Datasety použité v této sekci jsou výhradně ve formě *lower-case*. Pro trénování byly využity všechny vytvořené trénovací datasety, ale testování už probíhalo pouze na ATCO2-EN, tedy anglické části datové sady ATCO2. Obecným kritériem se stala otázka, jak dobře se trénované modely naučí rozpoznávat anglická data z této sady.

Tato sekce zahrnuje trénování výchozího modelu Vanilla Whisper Medium, který je zároveň označen za **BASELINE** model, na plných přepisech. Trénování probíhalo bez použití promptu, testování pak bez i s promptem (pro srovnání v následující kapitole Finetuning II.).

Protože testovací sada ATCO2-EN se skládá ze tří subsetů, jsou některé výsledky uváděny pro větší názornost zvlášť pro každý subset.

5.1 Nastavení hyperparametrů trénování

Motivací práce v této fázi bylo vytvořit pokud možno co nejpřesnější modely pro přepis. Pro trénování byly využity dvě sady hyperparametrů, u kterých, kromě počtu epoch, nedocházelo k žádným změnám. První byla otestována v rámci bakalářské práce Veroniky Nevařilové [7] a rovněž byla využita v původním skriptu pro *finetuning* modelu Whisper. Její hodnoty jsou následující:

I. skupina trénovacích hyperparametrů

```
per_device_train_batch_size=4
gradien_accumulation_steps=4
learning_rate=1.0e-5
warmup_ratio=0.12
weight_decay=0.0
num_train_epochs=30/15/8
```

Druhá trénovací sada hyperparametrů byla vytvořena přímo podle oficiálně dostupných rad týkajících se jejich ideálního nastavení. Díky nižší hodnotě hyperparametru *learning_rate* a nastavenému *weight_decay* lze předpokládat, že použití této sady bude výhodné především pro trénování, které by mohlo být ohroženo přetrénováním. Nastavení je uvedeno dále:

II. skupina trénovacích hyperparametrů

```
per_device_train_batch_size=4
gradien_accumulation_steps=8
learning_rate=6.25e-6
warmup_ratio=0.1
weight_decay=1.0e-6
num_train_epochs=30/15/8
```

Velikost dávky a počet kroků, po který se akumuluje gradient, je zmíněn v rámci *Whisper Finetuning Event*[1]. Zde bylo uvedeno doporučené nastavení pro grafické karty s dostupnou pamětí RAM o velikosti 16GB, což byl případ této práce. Nutné zmínit, že na uvedeném zdroji se nachází doporučení pro mírně odlišné nastavení hyperparametrů, než je uvedeno výše, konkrétně:

- `per_device_train_batch_size=2`,
- a `gradien_accumulation_steps=16`.

Tyto hodnoty také odpovídají celkovému počtu vzorků třicet dva (po jejichž zpracování dojde k aktualizaci vah sítě) a tudíž by toto nastavení mělo odpovídat tomu uvedenému výše.

Learning rate, tedy míra učení, vychází z doporučení[2] tvůrce skriptu pro trénování modelu Whisper. V tomto doporučení je zmíněno, že by tato hodnota měla být čtyřicetkrát menší než hodnota, která byla využita při trénování. V případě Whisper Medium byla použita míra učení 2.5×10^{-4} [9], která odpovídá čtyřicetinásobku výše nastavené hodnoty. Poznamenejme, že v témže příspěvku se autor vyjadřuje k využití *dropout* hodnoty pro malé datasety, přičemž tento hyperparametr nakonec nastaven nebyl (jeho nastavení zůstalo rovno nule), protože ve zmíněné bakalářské práci se stejným tématem nedošlo při nastavení tohoto parametru k zlepšení.

Stanovení parametru *warmup ratio* se odvíjelo opět od skriptu pro trénování Whisper, kde počet kroků, po které probíhá *zahřívání modelu*, odpovídá deseti procentům jejich celkového počtu.

Weight decay, byl nastaven na radu vedoucího práce, přičemž tato hodnota měla pozitivní dopad při jiných experimentech v rámci jeho výzkumné skupiny.

5.2 Konvence pojmenovávání modelů

Všechny trénované modely v základu vycházejí z Vanilla Whisper Medium. Ovšem jejich trénování už je nejenom prováděno s různými hyperparametry, ale především na různých datových sadách. Získané vlastnosti modelů na základě použité množiny dat mohou mít různý dopad při trénování s promptem v následující kapitole. Z toho důvodu je modelů ponecháno více a jejich pojmenování přímo odráží použité datasety. Vzniklá označení jsou vypsána níže:

- BASELINE – název pro model Vanilla Whisper Medium)
- ALLDS – model natrénovaný na všech vytvořených trénovacích datasetech)

[1] https://github.com/huggingface/community-events/tree/main/whisper-fine-tuning-event
[2] https://huggingface.co/spaces/openai/whisper/discussions/6#63c5731dfb9a6b829d898bc8

- ATCOEN – model natrénovaný na datasetu ATCO2-EN
- ALLDSATCOEN – model natrénovaný nejprve na všech vytvořených trénovacích datasetech a následně na ATCO2-EN

Protože trénování každého modelu proběhlo pro obě dříve zmíněné skupiny parametrů, jsou názvy modelů indexovány dolním indexem, aby bylo možno rozeznat, kterou skupinou parametrů byl natrénován (například $\text{ALLDSATCOEN}_{\text{II.}}$).

5.3 Trénování

Tabulka 5.1 ukazuje, jaké modely byly natrénovány. Smyslem zůstalo získat co nejnižší WER hodnotu na ATCO2-EN datech. Od tohoto cíle se také odvíjejí použité datové sady.

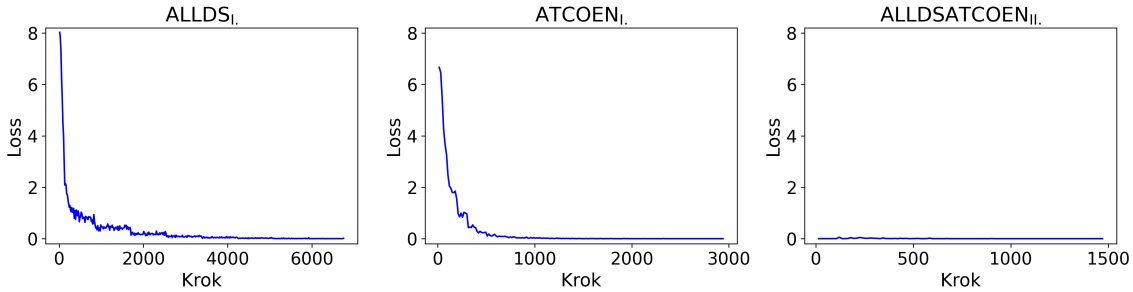
Výchozí model	**Trénovací sada**	**Počet epoch**	**Ozn. cílového modelu**
Vanilla Whisper Medium	–	–	BASELINE
Vanilla Whisper Medium	Všechny datasety, všechny jazyky	8	ALLDS
Vanilla Whisper Medium	ATCO2-EN	30	ATCOEN
ALLDS	ATCO2-EN	30	ALLDSATCOEN

Tabulka 5.1: Datasety použité při trénování základních modelů bez použití promptu a jejich pojmenování.

Pokaždé přitom proběhlo trénování zvlášť pro obě skupiny hyperparametrů uvedených výše (I. a II. skupina). Měněny byly pak pouze počty epoch.

Cílem při trénování na *všech datasetech* nebylo získat cílový model, ale pouze *předtrénovat* čistý Whisper Medium, aby došlo k uvedení modelu do kontextu promluv, a vytvořit tak základ pro následující natrénování s promptem. Proto byl také zvolený počet epoch pouze osm. Při trénování na datech ATCO2-EN byl zvýšen počet epoch na třicet, aby byla zajištěna dostatečně dlouhá doba pro adaptaci modelu. Dotrénování modelu ALLDS na ATCO2-EN datech bylo provedeno na třiceti epochách.

Na obrázku 5.1 jsou k vidění průběhy hodnot loss funkce pro trénování modelů za použití těch hyperparametrů, pro které modely dosáhly nejlepších WER hodnot.



Obrázek 5.1: Průběh trénovací loss funkce modelů ATCOEN$_{I.}$, ALLDS$_{I.}$, ALLDSATCOEN$_{II.}$, trénování proběhlo pro plný přepis.

Z dat lze pozorovat, že pokud se trénovalo na výchozím modelu Vanilla Whisper Medium, loss funkce klesala žádaným a očekávaným způsobem. Naopak při opětovném dotrénování modelu na datech ATCO2-EN, kdy model již data viděl v rámci trénování na všech datasetech, se loss funkce téměř nemění. Při přetrénování bychom očekávali růst hodnot loss funkce, to se ale neděje, a tudíž je pravděpodobné, že druhá adaptace modelu neublížila.

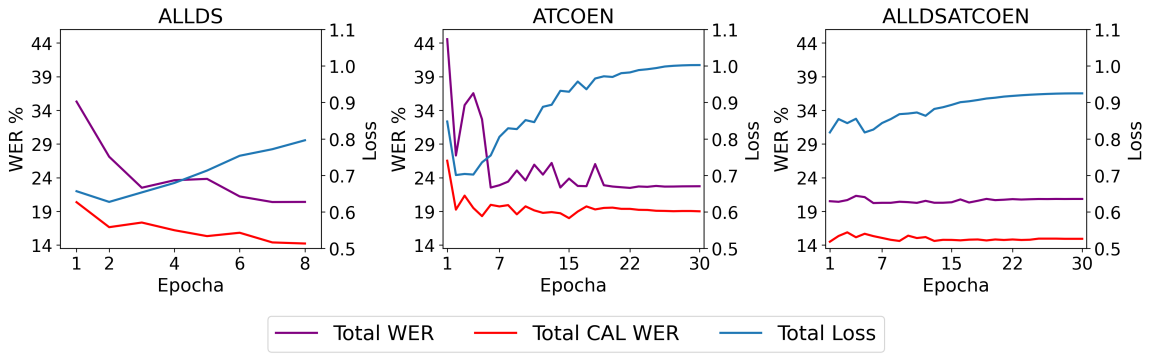
Tento formát loss funkcí se opakoval i pro další trénování. Tedy vždy platilo, že loss funkce modelů, které vycházely z Vanilla Whisper Medium, stabilně klesá, zatímco dotrénování modelů na ATCO2-EN datech, pokud model již předtím tato data viděl (tzn. jedná se o model ALLDSATCOEN), ovlivnilo loss funkci do podoby ukázané na posledním grafu na obrázku 5.1. Proto již dále v textu nebudou grafy loss funkcí uváděny.

5.4 Testování modelů

Testování se zaměřovalo výhradně na datovou sadu ATCO2-EN, která je složena ze tří subsetů – letiště Štefánik, Ruzyně a Zurich. Množiny Štefánik a Ruzyně jsou letiště, jejichž promluvy jsou částečně obsaženy v trénovací sadě. Tudíž výsledky na těchto množinách odpovídají na otázku, zda model bude pracovat lépe, pokud byl dříve adaptován na podmínky daného letiště. Oproti tomu množina dat Zurich v trénovací sadě nebyla obsažena vůbec a závěry z testovacích hodnot odpoví na otázku, nakolik lze model přizpůsobit, aby fungoval napříč podmínkami různých letišť.

5.4.1 Testy natrénovaných modelů bez promptu

Na obrázku 5.2 je průběh WER hodnot přepisů, WER hodnot volacích znaků a loss funkce na testovacích datech pro modely ALLDS$_{I.}$, ATCOEN$_{I.}$ a ALLDSATCOEN$_{II.}$.



Obrázek 5.2: Průběh validační loss funkce (modrá), WER hodnot plných přepisů (fialová) a WER hodnot volacích znaků (červená) pro modely $\text{ALLDS}_{\text{I.}}$, $\text{ATCOEN}_{\text{I.}}$ a $\text{ALLDSATCOEN}_{\text{II.}}$.

Průběh WER hodnot pro model $\text{ALLDS}_{\text{I.}}$ dokazuje, že se model velmi dobře učí a zřejmě by zde mohl být i potenciál ve vícero epochách. Zajímavé je pozorovat loss funkci, která sice během první a druhé epochy klesá, nicméně pak dojde k jejímu strmému růstu, zatímco WER hodnoty stále klesají.

Model $\text{ATCOEN}_{\text{I.}}$ již nevykazuje tak jednoznačný pokles WER hodnot. Nicméně jev, kdy jsou v nesouladu křivky loss a WER, se zde opakuje.

Pro model $\text{ALLDSATCOEN}_{\text{II.}}$ se zdá být dotrénování na ATCO2-EN datech neúčinné, s téměř konstantní křivkou WER hodnot.

Tabulka 5.2 uvádí nejlepší získané modely pro danou skupinu hyperparametrů. Uvedeny jsou též epochy, ve kterých bylo hodnot dosaženo. Z ukázaných hodnot můžeme vyčíst, že celkově nejnižší WER hodnota pro testovací sadu ATCO2-EN dosáhla hodnoty 20.24 % pro II. skupinu hyperparametrů na modelu ALLDSATCOEN. To bylo částečně očekávané, protože v parametrech této sady figuruje *learning rate* s hodnotou $6.25e-6$ společně s nastaveným *weight decay*. Tyto hyperparametry by měly mít lepší vliv na model v případě, kdy by mohlo hrozit přetrénování, v kontrastu k parametrům první skupiny, kde je *learning rate* nastaven na $1.0e-5$.

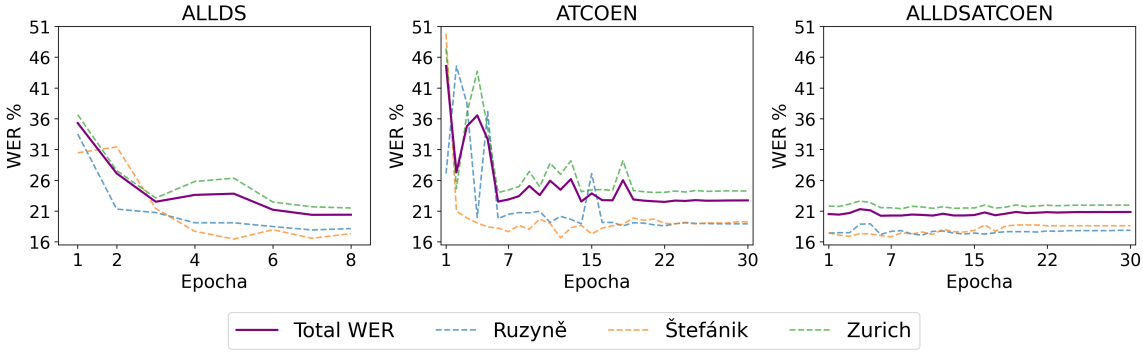
Model	**I. sk. par.**		**II. sk. par.**		**Celkem trénovacích epoch**
	WER %	ep	WER %	ep	
ALLDS	20.39	7	21.22	6	8
ATCOEN	22.48	22	23.55	19	30
ALLDSATCOEN	20.31	1	20.24	6	30
	–	–	20.27	8	30

Tabulka 5.2: Nejlepší získané WER hodnoty modelů pro danou skupinu hyperparametrů s uvedenou epochou, během které bylo hodnot dosaženo. Zvýrazněny jsou nejnižší hodnoty.

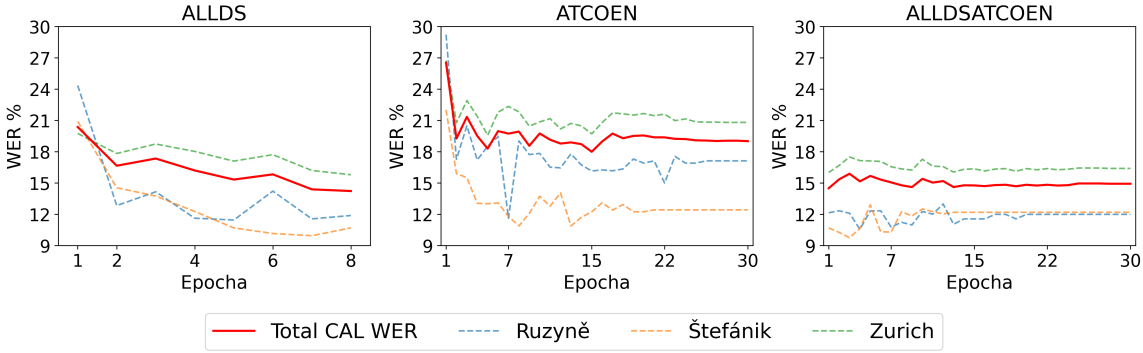
Model ALLDSATCOEN má zvýrazněny dvě nejlepší WER hodnoty pro druhou skupinu hyperparametrů, kdy nejlepší pochází z šesté a druhá z osmé epochy. Jak uvidíme v dalších grafech, model $\text{ALLDSATCOEN}_{\text{II.}}$ již nevykazuje žádné významné zlepšení a WER hodnoty

se liší jen na místě setin, případně desetin. Proto byly ponechány dvě podoby modelu s cílem zjistit, zda může mít na chování promptu (který je testován později) vliv doba adaptace navzdory téměř shodné WER hodnotě.

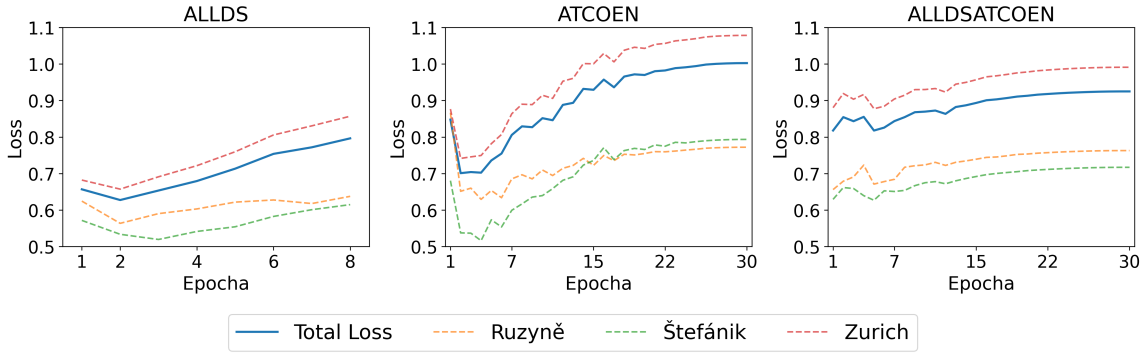
Nyní zbývá ukázat, jak si model vedl na datech z subsetů sady ATCO2-EN. Průběh WER hodnot celých přepisů a volacích znaků a hodnot loss funkce je předveden na grafech na obrázcích 5.3, 5.4 a 5.5 pro všechny nejlepší modely.



Obrázek 5.3: WER hodnoty plných přepisů, znázorněné pro jednotlivé subsety testovací sady v kontrastu s celkovou WER hodnotou plných přepisů.



Obrázek 5.4: WER hodnoty volacích znaků, znázorněné pro jednotlivé subsety testovací sady v kontrastu s celkovou WER hodnotou volacích znaků.



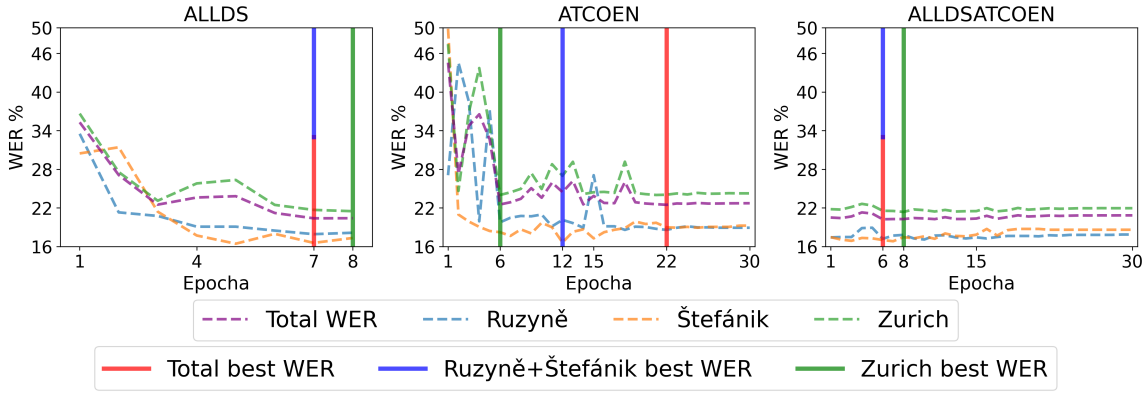
Obrázek 5.5: Loss funkce znázorněné pro jednotlivé subsety testovací sady v kontrastu s celkovou loss funkcí.

Při detailnějším rozboru WER hodnot přepisů lze pozorovat, že model skutečně lépe rozpoznává nahrávky z letišť Štefánik a Ruzyně, od kterých již měl k dispozici vzorky v trénovací sadě. Data z letiště Zurich jsou vždy hůře rozpoznaná. Tabulka 5.3 ukazuje nejlepší dosažené WER hodnoty pro různé části testovací datové sady a průměrný rozdíl během všech epoch mezi viděnými (Štefánik, Ruzyně) a neviděnými daty (Zurich) – nejdříve byl vypočítán v každé epoše vážený průměr WER pro data Štefánik a Ruzyně podle celkové délky audia a teprve potom počítán rozdíl od WER hodnot vypočítaných na datech Zurich (tento rozdíl byl zprůměrován počtem epoch).

Model	**Štef.+Ruzyně**		**Zurich**		**AVG rozdíl**	**Celkem**	
	WER %	ep	WER %	ep	WER %	WER %	ep
ALLDS	17.29	7	21.49	8	4.54	20.39	7
ATCOEN	18.51	12	24.01	6	6.76	22.48	22
ALLDSATCOEN	17.12	6	21.38	8	3.99	20.24	6
						20.27	8

Tabulka 5.3: Nejnižší dosažené WER hodnoty a epochy, ve kterých jich bylo dosaženo, pro jednotlivé subsety testovací datové sady. Ve třetím sloupci je ukázán průměrný rozdíl WER hodnot mezi sadami Štefánik+Ruzyně (viděná letiště) a Zurich (neviděné letiště). Poslední sloupec pro přehlednost ukazuje celkově nejlepší WER hodnoty i s epochami.

Na grafech na obrázku 5.6 lze sledovat epochu, v jaké se modely dostávaly k nejlepším hodnotám na jednotlivých subsetech testovací sady. Zatímco u modelů $\text{ALLDS}_{\text{I.}}$ a $\text{ALLDSATCOEN}_{\text{II.}}$ bylo dosaženo nejpřesnějšího rozpoznání promluv téměř ve stejných epochách pro všechny datové podmnožiny, u modelu $\text{ATCOEN}_{\text{I.}}$ tomu bylo přesně naopak – nejrychleji se model dobře adaptoval na letiště Zurich, o šest epoch později na letiště Ruzyni a Štefánik a až o dalších deset epoch později, tj. ve dvaadvacáté epoše, dosáhl model ideální celkové hodnoty metriky WER pro všechny subsety dohromady.



Obrázek 5.6: Grafy ukazující rozložený průběh WER hodnot jednotlivých subsetů testovací sady, přičemž jsou zvýrazněné vertikálními čarami ty epochy, během kterých byla získána nejnižší WER hodnota pro danou množinu dat.

Nyní se ještě, vzhledem k následující kapitole, potřebujeme zaměřit na ohodnocení volacích znaků. Algoritmus vyhodnocování callsignů je detailně popsán v kapitole *Whisper a použité prostředí, neuronové sítě a transformery* v sekci *Prostředí* (3.5.2). V tabulce 5.4 můžeme číst výsledky pro jednotlivé modely, které jsou dále v této a následující kapitole považovány za **referenční**.

Model	**WER přepisů (%)**	**WER volacích zn. (%)**	**Správně přepsané vol. zn.**
BASELINE	78.70	88.13	12 / 763 (1.57%)
ALLDS	20.39	14.39	451 / 763 (59.11%)
ATCOEN	22.48	19.37	397 / 763 (52.03%)
ALLDSATCOEN			
– 6. ep	20.24	15.33	456 / 763 (59.76%)
– 8. ep	20.27	14.78	459 / 763 (60.15%)

Tabulka 5.4: Porovnání úspěšnosti modelů na volacích znacích. V tabulce pozorujeme také vyhodnocení modelu **BASELINE**, tedy Vanilla Whisper Medium. Správně přepsaným volacím znakem je myšlena naprostá shoda mezi jeho formou v referenčním a predikovaném přepisu. Zobrazené hodnoty jsou považovány za **referenční** (pro model ALLDSATCOEN jsou referenční hodnoty z osmé epochy).

5.4.2 Testování modelů na datech s promptem

V této sekci je popsáno testování výše natrénovaných modelů bez promptu na datech s promptem. Zjistíme tak, zda si modely umí vzít pomocné informace z promptu, aniž by na to byly dříve trénovány. Porovnání probíhá s WER hodnotami přepisů, které jsou zobrazené v tabulce 5.4.

Model	**AG**	**AG4B**	**AG35B**	**AG50CZB**	**5B**	**35B**	**50CZB**
BASELINE	53.31	49.65	48.25	62.27	59.49	49.02	80.96
	(-25.39)	(-29.05)	(-30.45)	(-16.43)	(-19.21)	(-29.68)	(+2.26)
$\text{ALLDS}_{\text{I.}}$	18.00	22.95	34.64	28.79	21.51	28.43	24.62
	(-2.39)	(+2.56)	(+14.25)	(+8.40)	(+1.12)	(+8.04)	(+4.23)
$\text{ATCOEN}_{\text{I.}}$	18.81	19.42	19.63	19.73	22.25	21.69	22.97
	(-3.67)	(-3.06)	(-2.85)	(-2.75)	(-0.23)	(-0.79)	(+0.49)
$\text{ALLDSATCOEN}_{\text{II.}}$							
– 6. ep	17.66	22.77	35.25	29.90	21.59	29.96	25.08
– 8. ep	17.91	20.76	27.22	21.80	20.74	25.29	22.10
srov. 8.ep	(-2.36)	(+0.49)	(+6.95)	(+1.53)	(+0.47)	(+5.02)	(+1.83)

Tabulka 5.5: WER hodnoty přepisů (%) pro základní modely, které byly testované na všech typech promptů. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě daného základního modelu (testovaného bez promptu, viz tabulka 5.4) – zeleně zlepšení (nižší WER), červeně zhoršení.

Pokud u modelu $\text{ALLDSATCOEN}_{\text{II.}}$ srovnáme osmou a šestou epochu, vidíme, že výsledky jsou jednoznačně ve prospěch využití modelu z osmé epochy (kromě evaluace na promptu AG). Konstatovat tedy můžeme, že ačkoliv se WER hodnota při evaluaci jednotlivých epoch tohoto modelu takřka neměnila, může se zcela lišit jeho reakce na použití promptu. **Jakýkoliv další odkaz na model $\text{ALLDSATCOEN}_{\text{II.}}$ bude směřovat na výsledky v jeho osmé epoše.**

Očekávané zlepšení všech modelů se skutečně dostavilo pouze při použití promptu AG. To odpovídá skutečnosti, že promptem modelu předaná slova, která se vyskytují v promluvě, mu pomáhají v přepisu. Na další typy promptu však modely již reagovaly různě, přičemž překvapivě nejlépe fungovaly prompty pro model $\text{ATCOEN}_{\text{I.}}$. Z takového chování bychom mohli usuzovat, že modelu může pomoct pouhé přidání promptu (tzn. bez adaptace na prompt), pokud byl dříve na datech ze stejného prostředí adaptován. To je potvrzeno i tím, že zcela největšího propadu dosáhl model $\text{ALLDS}_{\text{I.}}$, který data ATCO2-EN viděl pouze ve skupině *všech* trénovacích datasetů, a tudíž se na ně nemohl zaměřit.

Zde se vyskytuje ještě jedna zvláštnost, kterou můžeme pozorovat při porovnání dopadu promptu AG50CZB a 50CZB (oranžově zvýrazněná pole v tabulce 5.5) na model $\text{ALLDS}_{\text{I.}}$. Předpoklad zcela jistě zní, že výskyt správné sekvence slov v promptu se směsí náhodných slov (AG50CZB) by měl mít lepší či přinejmenším stejný dopad na model, než pouze náhodný řetězec slov (50CZB). Metrika WER nicméně odhaluje, že se model zhoršil více při

použití AG50CZB. Náhled na WER metriku spočítanou na volacích znacích (ukázáno dále v tabulce 5.6) sice ukazuje lepší výsledek pro AG50CZB (18.73%) proti 50CZB (19.38%) – vzhledem k původním hodnotám se však jedná o zhoršení. Jedním z možných vysvětlení je, že model zřejmě zvládl částečně odhalit vyskytující se volací znak v promptu, ale snažil se okolí přepisu přizpůsobit okolí promptu kolem volacího znaku. Ovšem vizuální kontrola přepisů nastíněnou ideu nepotvrdila. Přepisy pod vlivem promptu AG50CZB však obsahovaly násobně větší počet čárek, konkrétně 657 – přitom při použití promptu 50CZB se v přepisu objevilo pouze 252 čárek. Existuje potom ještě druhá možnost vlivu, a to že typ použitých náhodných slov, byť v promluvě s jistotou neexistujících, ovlivňuje přepis (připomeňme, že náhodná slova vyskytující se v promptu AG50CZB jsou jiná od těch v 50CZB, což by teoreticky mohlo vést k ovlivnění správnosti přepisu). Tato hypotéza však již nebyla potvrzena.

Ke zmíněnému problematickému srovnání promptů AG50CZB a 50CZB zbývá dodat, že modely BASELINE a $\text{ATCOEN}_{\text{I.}}$ se zlepšily při použití promptu AG50CZB a zhoršily při použití 50CZB. Model $\text{ALLDSATCOEN}_{\text{II.}}$ pak vykazuje menší zhoršení u AG50CZB než u 50CZB. Odtud pak můžeme konstatovat, že reakce modelů na diskutované dva prompty je přímo úměrná míře zaměření modelu na jiné datasety (tzn. kromě ATCO2-EN).

Model BASELINE se rovněž zlepšil při použití téměř všech promptů kromě 50CZB. Navíc zlepšení je přímo úměrné množství předaných použitelných informací a můžeme proto tvrdit, že se BASELINE chová nejvíc předvídatelně.

Vyhodnocení kvality přepisu volacích znaků

Tabulka 5.6 poskytuje přehled WER hodnot volacích znaků vyskytujících se v promluvách přepsaných jednotlivými modely.

Model	**AG**	**AG4B**	**AG35B**	**AG50CZB**	**5B**	**35B**	**50CZB**
BASELINE	21.75	30.06	33.82	40.33	51.50	42.43	96.01
	(-66.38)	(-58.07)	(-54.31)	(-47.80)	(-36.37)	(-46.70)	(+7.88)
$\text{ALLDS}_{\text{I.}}$	6.08	14.17	26.11	18.73	15.51	24.94	19.38
	(-8.31)	(-0.22)	(+11.72)	(+4.34)	(+1.12)	(+10.55)	(+4.99)
$\text{ATCOEN}_{\text{I.}}$	6.29	7.72	9.98	9.81	17.78	16.08	19.91
	(-13.08)	(-11.65)	(-9.39)	(-9.56)	(+1.59)	(+3.29)	(+0.54)
$\text{ALLDSATCOEN}_{\text{II.}}$	5.98	11.01	19.29	12.47	14.76	19.98	16.69
	(-8.80)	(-3.77)	(+4.51)	(-2.31)	(+0.02)	(+5.20)	(+1.91)

Tabulka 5.6: WER hodnoty (%) volacích znaků pro testované prompty na základních modelech. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě bez promptu (viz tabulka 5.4) – zeleně zlepšení, červeně zhoršení.

Model BASELINE vykazuje vysoké zlepšení WER hodnot volacích znaků pro všechny typy promptu kromě 50CZB. To opět odpovídá předpokládanému chování Whisperu, který prompt chápe jako přepis předcházejícího audia – prompt 50CZB v tomto případě uvádí model do úplně špatného kontextu.

U dalších modelů pozorujeme, že k horším WER hodnotám došlo mnohokrát i v případě, kde v předcházející tabulce s WER hodnotami přepisů naopak přišlo zlepšení – zcela úměrné zlepšení a zhoršení nalezneme pouze pro model BASELINE. Model $\text{ALLDS}_{\text{I.}}$ zlepšil kvalitu přepisu volacích znaků při použití promptu AG4B, AG35B a AG50CZB, i když po stránce WER hodnot přepisů se zhoršil. Podobně se vedlo i modelu $\text{ALLDSATCOEN}_{\text{II.}}$. Model $\text{ATCOEN}_{\text{I.}}$ zřejmě dokáže najít v promptu správné volací znaky, protože jeho zlepšení WER hodnot přepisu volacích znaků přišlo u všech promptů, ve kterých se správné volací znaky vyskytovaly.

Pozastavit bychom se mohli nad výsledkem modelů při použití promptu 35B a 50CZB, tedy třicet pět v promluvě se nevyskytujících volacích znaků a padesát náhodných českých slov. Modely $\text{ALLDS}_{\text{I.}}$ a $\text{ALLDSATCOEN}_{\text{II.}}$ při použití těchto promptů vykázaly markantní zhoršení výsledků jak pro celé přepisy, tak pro volací znaky natolik, že v absolutních číslech zůstaly výrazně za modelem $\text{ATCOEN}_{\text{I.}}$. Z toho můžeme vyvodit závěr, že model natrénovaný na menší objem dat je odolnější vůči „chybným“ datům v promptu.

Obecně pro všechny testované prompty můžeme pozorovat stejný vzorec pro trojici modelů $\text{ALLDS}_{\text{I.}}$, $\text{ATCOEN}_{\text{I.}}$ a $\text{ALLDSATCOEN}_{\text{II.}}$, kdy model $\text{ALLDS}_{\text{I.}}$ je vždy se svými rozdíly za modelem $\text{ALLDSATCOEN}_{\text{II.}}$, který zase zaostává za $\text{ATCOEN}_{\text{I.}}$. Z něj bychom mohli usuzovat, že prompt bez předchozí adaptace pomůže spíše modelu, který je natrénovaný na menším objemu dat (máme na mysli model $\text{ATCOEN}_{\text{I.}}$).

Ohodnocení modelů pouze na datech obsahujících volací znaky

Protože část promluv testovací sady ATCO2-EN neobsahuje volací znaky, je důležité ukázat dopad promptu pouze na ty promluvy, které volací znaky obsahují. Zejména pro tuto část promluv je očekáváno zlepšení přepisu volacích znaků při použití promptů AG, AG4B, AG35B a AG50CZB.

Pro větší názornost jsou ukázány typy vyňatých a ponechaných promluv z ATCO2-EN pro následující testování:

- PROMLUVA A: „Hello, this is Alfa Bravo Charlie One One, QNH one zero two six“ – tato promluva je ponechána, protože obsahuje volací znak (označený zeleně).
- PROMLUVA B: „Good morning, can you repeat your callsign?“ – tato promluva je odebrána, protože **neobsahuje** volací znak.

Pro následující výsledky tedy jako testovací dataset zůstává ATCO2-EN, ale figurují v něm **pouze promluvy s volacími znaky**. V tabulce 5.7 jsou uvedeny výsledky testování základních modelů bez promptu na tomto omezeném datasetu. Tabulka pak 5.8 ukazuje základní modely při testování s promptem na omezeném datasetu i s uvedenými rozdíly proti tabulce 5.7.

Model	WER přepisů (%)
ATCOEN	21.81
ALLDS	19.66
ALLDSATCOEN	19.47
BASELINE	79.07

Tabulka 5.7: Porovnání úspěšnosti základních modelů **pouze** na těch přepisech z testovací sady ATCO2-EN, které **obsahují volací znaky**. Tyto hodnoty figurují jako referenční pro tabulku 5.8.

Model	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
BASELINE	52.39	48.51	47.03	61.90	59.58	47.97	79.58
	(-26.68)	(-30.56)	(-32.04)	(-17.17)	(-19.49)	(-31.06)	(+0.51)
$\text{ALLDS}_{\text{I.}}$	17.04	22.29	34.69	28.49	20.74	28.08	24.00
	(-2.62)	(+2.63)	(+15.03)	(+8.83)	(+1.08)	(+8.42)	(+4.34)
$\text{ATCOEN}_{\text{I.}}$	17.85	18.50	18.72	18.84	21.58	21.03	22.25
	(-3.96)	(-3.31)	(-3.09)	(-2.97)	(+0.23)	(-0.78)	(+0.44)
$\text{ALLDSATCOEN}_{\text{II.}}$	16.98	19.99	26.85	21.10	19.92	24.72	21.28
	(-2.49)	(+0.52)	(+7.38)	(+1.63)	(+0.45)	(+5.25)	(+1.81)

Tabulka 5.8: WER hodnoty (%) základních modelů na promluvách z testovací sady ATCO2-EN, které obsahují volací znaky. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě bez promptu (tabulka 5.7) – zeleně zlepšení (nižší WER), červeně zhoršení.

Jedinou změnou v ukázaných rozdílových hodnotách (označené v tabulce zeleně a červeně) oproti tabulce 5.5 (ve které jsou výsledky testování na celé ATCO2-EN sadě) je zhoršení modelu $\text{ATCOEN}_{\text{I.}}$ na promptu 5B. Z toho můžeme vyvodit, že obsah tohoto promptu pravděpodobně tomuto modelu pomáhal při přepisu jiných částí promluv než volacích znaků.

5.5 Závěr

V této kapitole bylo provedeno natrénování základních modelů na plné přepisy. Pro trénování byly použity všechny vytvořené datové sady, ale testování se týkalo již výhradně datasetu ATCO2-EN, na kterém proběhlo vyhodnocení přepisů promluv a zvlášť vyhodnocení přepisů volacích znaků obsažených v těchto promluvách. Každý model byl trénován pro dvě skupiny hyperparametrů, testování bylo ukázáno jen pro tu skupinu, pro kterou model dosáhl nejlepších WER hodnot (uvedena v dolním indexu modelů).

Základní trénování ukázalo, že model $\text{ALLDSATCOEN}_{\text{II.}}$, který byl trénován na všech datasetech dohromady a následně na datasetu ATCO2-EN, dosáhl nejlepších WER hodnot ze všech získaných modelů – v šesté epoše trénování dosáhl 20.24 % WER na ATCO2-EN.

Model $\text{ATCOEN}_{\text{I.}}$ vykázal jako svou nejlepší dosaženou hodnotu 22.48 % WER, která ale byla tou nejvyšší při srovnání s modely ALLDSATCOEN, ALLDS.

Následně proběhlo testování natrénovaných modelů s promptem. Přestože modely na prompt adaptovány nebyly, došlo k nečekaným zlepšením. Nejlépe si vedl model $\text{ATCOEN}_{\text{I.}}$, který se zlepšil při použití téměř všech promptů, nejvíce o 3.37 % na promptu AG. Model BASELINE také vykazoval vysoká zlepšení, což bylo z hlediska funkce promptu u modelu Whisper očekávané. U modelu $\text{ALLDSATCOEN}_{\text{II.}}$ se projevilo, že ačkoliv výsledky dvou zvolených epoch dosahovaly přibližně stejné hodnoty WER (20.24 % WER v šesté a 20.27 % WER v osmé epoše), při provedení testování s promptem se výsledky jedné i druhé epochy velice lišily. Největší rozdíl nastal pro prompt AG50CZB, kdy verze modelu $\text{ALLDSATCOEN}_{\text{II.}}$ z šesté epochy dosáhla 29.90 % WER, zatímco z osmé epochy 21.80 %.

Kapitola 6

Finetuning II. – Adaptace Whisperu na plných přepisech s použitím promptu

Jako cílová i trénovací byla opět zvolena datová sada ATCO2-EN.

Předpoklad využití promptu v praxi spočíval v předání hypotetických volacích znaků modelu, mezi kterými by se měl vyskytovat i ten skutečně obsažený v promluvě. Očekávaný dopad, který se měl potvrdit během testování, se měl projevit ve zlepšení přepisu volacích znaků či dalších slov, která se vyskytovala v promptu nebo promluvě.

6.1 Použité modely a konvence názvů

Pro experimenty byla zvolena trojice modelů: **BASELINE**, tedy Vanilla Whisper Medium, dále vybrané modely natrénované v předešlé kapitole Finetuning I. (5) – **ALLDS** z osmé epochy (Vanilla Whisper Medium dotrénovaný na všech datasetech) trénovaný za použití parametrů I. skupiny a **ALLDSATCOEN** z osmé epochy (Vanilla Whisper Medium dotrénovaný nejprve na všech datasetech a následně na ATCO2-EN datech) trénovaný na parametrech II. skupiny (v minulé kapitole jsme pro rozlišení uváděly čísla skupin hyperparametrů v dolním indexu modelu, v této kapitole ovšem budeme chápat modely ALLDS a ALLDSATCOEN tak, jak byly nyní uvedeny a dolní index již nebudeme uvádět). Vybrané modely jsou z těch epoch, kde bylo dosaženo nejlepších WER hodnot na subsetu Zurich, protože se jednalo o výsledek na neviděném letišti.

Jak bylo zmíněno v kapitole Finetuning I. (5), i zde je zaveden pro větší přehlednost systém označení jednotlivých trénovaných modelů. Veškeré trénování probíhá s promptem a zároveň data vždy pocházejí ze sady ATCO2-EN pro testování i trénování. Pojmenování v této kapitole má dvě formy:

1. Forma – pojmenováváme celou rodinu modelů, které vycházejí ze stejného základového modelu, ale každý z nich je trénován na jiný typ promptu. Tyto rodiny zde označujeme následovně:
    - Model P-ALLDSATCOEN[1]
    - Model P-ALLDS[1]
    - Model P-BASELINE[1]

[1] K názvu modelu je přidáno na začátek písmeno **P**, které označuje, že se jedná o modely trénované za použití PROMPTU.

2. Forma – pojmenováváme jednotlivé konkrétní modely. Takové pojmenování má podobu jednoho ze tří uvedených v 1. Formě, kde za **P** je dosazen prompt, na který byl model trénován, tedy jeden z AG, AG4B, AG35B, AG50CZB, 5B, 35B nebo AG50CZB.

   Například označení AG-BASELINE ukazuje na model BASELINE (tedy Vanilla Whisper Medium), trénovaný na ATCO2-EN datech s použitím promptu AG.

Během testování jsou modely srovnávány s těmi, které byly trénovány bez promptu na stejných datasetech. Tabulka 6.1 ukazuje po řádcích dvojice modelů, které jsou v této kapitole srovnávány:

Rodina modelů trénovaná s promptem	Základní model (trénován bez promptu)
P-ALLDS	ALLDSATCOEN
P-BASELINE	ATCOEN
P-ALLDSATCOEN	*není k dispozici stejně trénovaný model bez promptu*

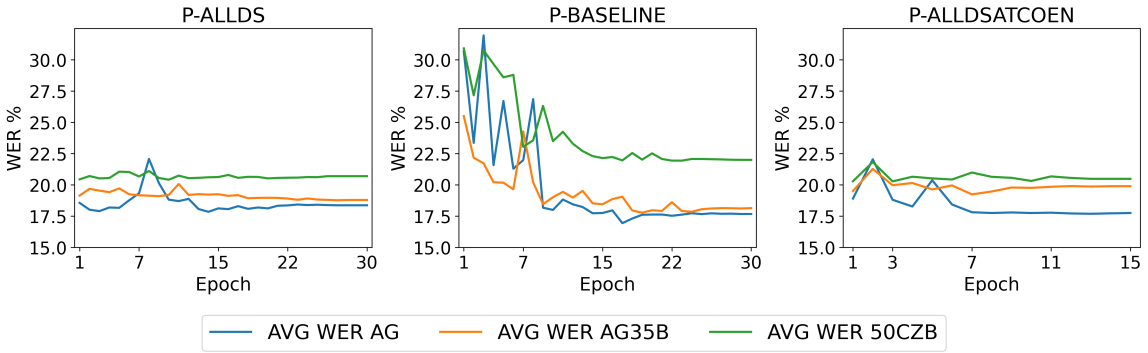
Tabulka 6.1: Modely, pro které je prováděno srovnání WER hodnot.

6.2 Trénování

Každý z modelů byl natrénován na všech sedm zmíněných typů promptů (jedná se o dataset ATCO2-EN). Pro trénování modelů P-BASELINE bylo využito trénovacích hyperparametrů I. skupiny s nastavením na třicet epoch. Modely P-ALLDS byly trénovány na prompt s parametry II. skupiny po třicet epoch. Rodina modelů P-ALLDSATCOEN byla trénována s parametry II. skupiny po dobu patnácti epoch.

6.3 Testování

Na obrázku 6.1 je ukázán průběh evaluace epoch modelů pro trénování za použití promptů AG, AG35B a 50CZB (stejné byly využity právě i během evaluace).



Obrázek 6.1: Evaluace jednotlivých epoch pro vybrané modely.

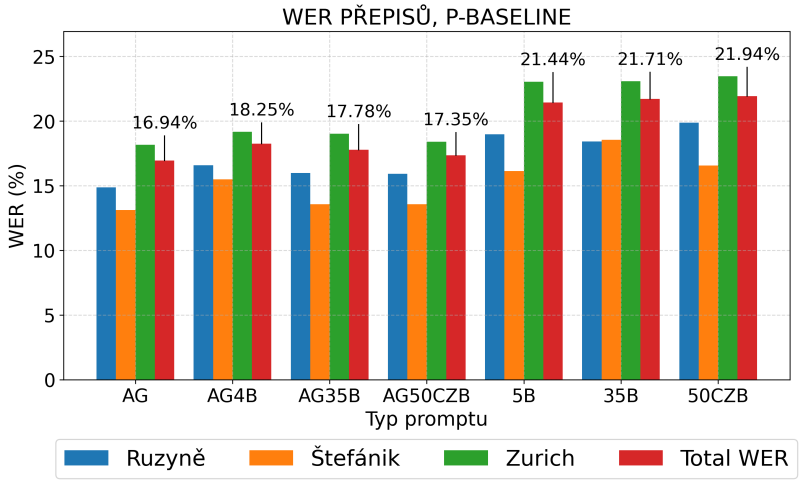
Z dat grafů vidíme očekávaný strmý pokles hodnot pro modely patřící do rodiny P-BASELINE, které žádná data ATCO2-EN ještě neviděla. Druhé dvě skupiny modelů částečně trpí přetrénováním. U modelu AG-ALLDSATCOEN ale vidíme, že WER hodnoty klesají. To může souviset s tím, že informační hodnota promptu AG je ze všech největší. Model AG-ALLDS je však zjevně přetrénován, a tak nemůžeme říct závěr, že prompt AG při trénování modelům jednoznačně pomáhá.

V tabulce 6.2, korespondující s grafy na obrázku 6.1 můžeme detailně číst nejnižší celkové WER hodnoty i s epochami, ve kterých jich bylo dosaženo.

Rodina modelů	**AG**		**AG35B**		**50CZB**	
	WER %	ep	WER %	ep	WER %	ep
P-ALLDS	17.85	14	18.76	27	20.41	10
P-BASELINE	16.94	17	17.78	19	21.94	22
P-ALLDSATCOEN	17.69	13	19.24	7	20.28	3

Tabulka 6.2: Nejnižší dosažené WER hodnoty (%) pro plné přepisy promluv a epochy, ve kterých jich bylo dosaženo, pro modely trénované na promptech AG, AG35B a 50CZB. V levém sloupci se nacházejí označení rodin modelů a každý další sloupec pak označuje model z každé rodiny natrénovaný na daný prompt.

Nejlepších výsledků obecně při použití promptu dosahuje model P-BASELINE (Vanilla Whisper Medium). Pro názornější porovnání jeho chování při použití jednotlivých druhů promptu je k dispozici graf 6.2.



Obrázek 6.2: Graf ukazující celkovou úspěšnost přepisu dat pro každý typ promptu rodinou modelů P-BASELINE, přičemž pro každý prompt se liší epocha modelu, ve které dosahovala metrika WER nejnižších hodnot.

Vcelku se model chová předvídatelně. Vezmeme-li v potaz, že model ATCOEN (základní model trénovaný bez promptu) dosahuje celkové WER 22.48 % je jisté, že zavedením trénování s promptem nedojde ke zhoršení kvality rozpoznávání promluv. Zajímavé výsledky jsou zde dva:

- S promptem natrénovaný model **AG50CZB-BASELINE** vykazuje při testování druhý nejlepší výsledek ze všech promptů. Pravděpodobně se model naučil v sérii českých slov najít volací znak. To ovšem může být hypoteticky usnadněno jiným jazykem volacího znaku, kdy okolní slova jsou v češtině, zatímco volací znak v angličtině. Tuto hypotézu podporuje i WER hodnota modelu 50CZB-BASELINE, která je nejhorší. Druhou možností může být též delší sekvence slov mezi čárkami, na kterou by si mohl model zvyknout, a tak automaticky by došlo k výběru slov volacího znaku (volací znak obsahuje obvykle tři a více slov, zatímco česká slova jsou po jednom oddělena čárkami, ukázáno níže 6.3).

  ahoj, můj, nový, traktor, one zero alpha bravo, hezký, potravina

- Prompt **50CZB** stále ovlivňuje model pozitivním způsobem. S jistotou lze prohlásit, že žádné slovo v promptu 50CZB se nevyskytuje v dané promluvě. Přesto by důvodem zlepšení mohlo být správné rozlišení některých českých slov, které se zřídka v nahrávkách vyskytují díky tomu, že náhodná slova jsou stále v českém jazyce.

6.3.1 Testování modelů na datech s promptem

V této části se nacházejí výsledky testování každého modelu ze všech tří rodin modelů (P-ALLDS, P-BASELINE, P-ALLDSATCOEN). Testování probíhá za použití promptu. Pro názornost jsou zde slovně popsány dva příklady testů:

- model, natrénovaný na prompt AG4B (například AG4B-BASELINE) je testován na datech ATCO2-EN za použití promptu typu AG4B;
- model natrénovaný na prompt 50CZB (například 50CZB-ALLDS) je testován na datech ATCO2-EN za použití promptu typu 50CZB.

Tabulka 6.3 nám ukazuje WER hodnoty pro jednotlivé modely trénované na daných promptech. U modelů z rodin P-BASELINE a P-ALLDSATCOEN navíc můžeme pozorovat, o kolik se získané výsledky liší oproti základním modelům, které byly trénované bez promptu na stejných datasetech.

WER hodnoty celých přepisů

Rodina modelů	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
P-ALLDS	17.85	18.70	18.76	18.40	19.95	20.10	20.41
	– Rozdíl proti ALLDSATCOEN bez promptu (WER = 20.27%)						
	(-2.42)	(-1.57)	(-1.51)	(-1.87)	(-0.32)	(-0.17)	(+0.14)
	– Rozdíl proti ALLDSATCOEN testovanému s promptem (WER viz. 5.5)						
	(-0.06)	(-2.06)	(-8.46)	(-3.40)	(-0.79)	(-5.19)	(-1.69)
P-BASELINE	16.94	18.25	17.78	17.35	21.44	21.71	21.94
	– Rozdíl proti ATCOEN bez promptu (WER = 22.48%)						
	(-5.54)	(-4.23)	(-4.70)	(-5.13)	(-1.04)	(-0.77)	(-0.54)
	– Rozdíl proti ATCOEN testovanému s promptem (WER viz. 5.5)						
	(-1.87)	(-1.17)	(-1.85)	(-2.38)	(-0.81)	(+0.02)	(-1.03)
P-ALLDSATCOEN	17.65	18.95	19.20	19.00	19.81	19.91	20.22

Tabulka 6.3: WER hodnoty (%) plných přepisů ATCO2-EN datasetu za použití promptů na modelech adaptovaných na ně. V levém sloupci je název každé rodiny a v každém dalším sloupci jsou ukázány WER hodnoty adaptovaného modelu na konkrétní typ promptu. Druhý číselný řádek pro každou rodinu modelů obsahuje rozdíl oproti referenční hodnotě modelu trénovaného bez promptu – zeleně zlepšení, červeně zhoršení. Pro rodiny P-ALLDS je referenční hodnotou WER 20.27% získaná modelem ALLDSATCOEN, pro P-BASELINE potom WER 22.48% získaná modelem ATCOEN. Oranžově je označen sloupec pro modely trénované a testované na promptu AG, modře pak sloupec s modely týkající se promptu AG35B. Třetí řádek pro každou rodinu modelů srovnává základní modely ALLDSATCOEN a ATCOEN, testovanými s promptem proti jednotlivým adaptacím modelů na prompt. K modelům P-ALLDSATCOEN neexistuje ekvivalentně trénovaný model (model trénovaný na všech dostupných datasetech a poté dvakrát na ATCO2-EN není dostupný), a tudíž pro něj nejsou zobrazena srovnání.

Z dat lze poznat, že po natrénování modelů má pozitivnější vliv použití promptu AG. To ukazuje, že pokud model ví, že se obsah promptu bude v celé své formě vyskytovat v promluvě, potom jej dokáže zasadit správně do promluvy. Dále si všimněme, že nejlepších hodnot obecně pro prompty, kde se vyskytoval správný volací znak (AG_), je dosaženo rodinou modelů P-BASELINE.

Tabulka 6.4 nám názorně srovnává dva modely, které prošly adaptací na stejných datech, ale jeden (P-BASELINE) s adaptací na prompt AG a druhý (ATCOEN) bez adaptace na prompt.

Model	WER plných přepisů (%)	WER vol. zn. (%)	Počet správně přepsaných vol. zn.
AG-BASELINE	16.94	2.64	719
ATCOEN, adaptovaný bez promptu	22.48	19.37	397
rozdíl	5.54	16.73	322

Tabulka 6.4: Srovnání WER hodnot (%) přepisu a volacích znaků v nich pro modely ATCOEN a AG-BASELINE.

Nyní pojďme ověřit, nakolik užití promptu AG ovlivňuje i jiné části přepisu, než jen volací znaky v promluvě. Výpočet vztáhneme k modelu AG-BASELINE:

- Obsah volacích znaků v testovací sadě ATCO2-EN ..... 23.42 %
- Rozdíl WER hodnot vol. zn. mezi AG-BASELINE a ATCOEN ..... 16.73 %

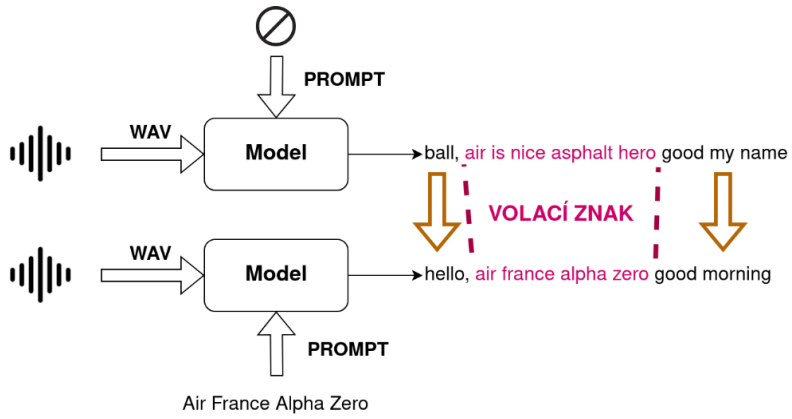
Potom,

$$\frac{23.42}{100} \times 16.73 = 3.92 \tag{6.1}$$

je výpočet, který lze interpretovat slovy: *„pokud se 23.42% přepisu (což jsou volací znaky) zlepší o 16.94%, potom pokud by se toto zlepšení týkalo pouze volacích znaků v promluvách, tak by rozdíl WER hodnot přepisů (100%) měl být 3.92%“*. Vidíme však, že rozdíl mezi AG-BASELINE a ATCOEN je 5.54%, což ukazuje, že dotrénování na promptu, který obsahuje jen volací znaky, ovlivňuje také jiné části promluvy. To je možné ze dvou důvodů – první, přepis obsahuje slova, která jsou uvedena v samotném volacím znaku. Jednalo by se o případ:

> PROMPT: Alpha Bravo **Zero** X-Ray **One**
> PROMLUVA: Hey, here is Alpha Bravo Zero X-Ray One, altitude Three **One Zero Zero**.

Tučně jsou ukázána slova, která jsou předaná jako součást volacího znaku, ale vyskytují se i jinde v přepisu. Druhým důvodem by mohla být adaptace na řečníka. Tu ilustruje následující obrázek 6.3. V této hypotéze by platilo, že model by zjistil, jaký přepis je správný pro danou část promluvy (v tomto případě částí promluvy rozumíme vyslovený volací znak) a následně by podle toho upravil své porozumění dalším částem promluvy. Tato hypotéza však není prokazatelná.



Obrázek 6.3: Ilustrace ovlivnění přepisu promptem jako nástrojem na adaptaci modelu na řečníka.

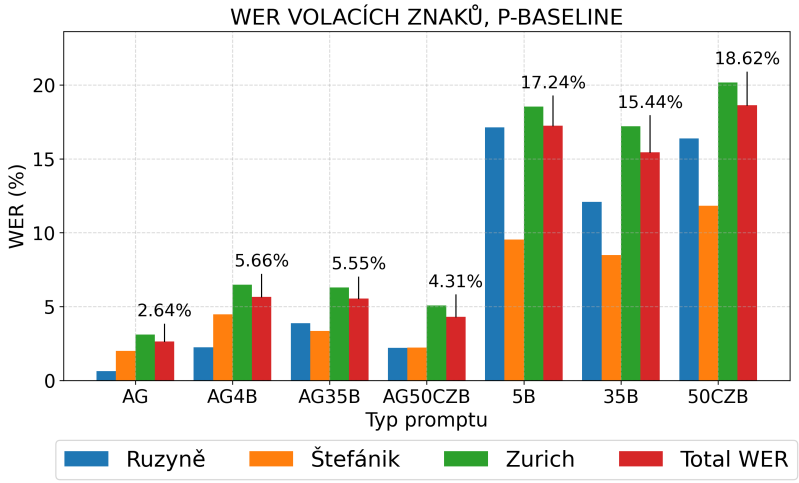
WER hodnoty volacích znaků

V tabulce 6.5 čteme výsledné WER hodnoty pro volací znaky při provedení testování modelů s promptem.

Rodina modelů	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
P-ALLDS	7.19	9.04	10.40	9.26	13.98	14.59	15.21
	↓ Rozdíl proti ALLDSATCOEN test. bez promptu (WER 14.78 %) ↓						
	(-7.59)	(-5.74)	(-4.38)	(-5.52)	(-0.80)	(-0.19)	(+0.43)
	↓ Rozdíl proti ALLDSATCOEN test. s promptem (WER viz. 5.6) ↓						
	(+1.21)	(-1.97)	(-8.89)	(-3.21)	(-0.78)	(-5.39)	(-1.48)
P-BASELINE	2.64	5.66	5.55	4.31	17.24	15.44	18.62
	↓ Rozdíl proti ATCOEN test. bez promptu (WER 19.37 %) ↓						
	(-16.73)	(-13.71)	(-13.82)	(-15.06)	(-2.13)	(-3.93)	(-0.75)
	↓ Rozdíl proti ALLDSATCOEN test. s promptem (WER viz. 5.6) ↓						
	(-3.65)	(-2.06)	(-4.43)	(-5.50)	(-0.54)	(-0.64)	(-1.29)
P-ALLDSATCOEN	7.05	10.52	12.38	10.15	14.28	14.09	15.17

Tabulka 6.5: WER hodnoty (%) transkripce volacích znaků za použití promptů. V levém sloupci je název každé rodiny a v každém dalším sloupci jsou ukázány WER hodnoty adaptovaného modelu na konkrétní typ promptu. Druhý řádek pro každou rodinu modelů ukazuje rozdíl oproti referenční hodnotě bez promptu (ATCOEN pro rodinu P-BASELINE, ALLDSATCOEN pro rodiny P-ALLDS, tabulka 5.4) – zeleně zlepšení, červeně zhoršení. Třetí řádek je rozdíl proti testům základních modelů na datové sadě s promptem (viz tabulka 5.5).

Téměř všude došlo ke zlepšení. Výjimku tvoří prompt AG, kde vidíme, že využití tohoto promptu u modelu základního modelu ALLDSATCOEN (který na prompt adaptován není) je účinnější, než je tomu u AG-ALLDS, který na tento typ adaptován byl. Níže na obrázku 6.4 pak můžeme pozorovat rozložení WER hodnot volacích znaků pro jednotlivé subsety ATCO2-EN pro rodinu modelů P-BASELINE.



Obrázek 6.4: WER hodnoty přepisu volacích znaků pro rodinu modelů P-BASELINE.

Zde si můžeme všimnout, že přepis promluv viděného letiště Štefánik je horší, nebo téměř stejný jako pro letiště Ruzyně, pokud se v promptu vyskytuje správný volací znak (tzn. prompty AG_). Naopak pokud je celý prompt bez správného volacího znaku, Štefánik začne mít výrazně lepší výsledky než Ruzyně. Tento výsledek je zvláštní, protože subset Ruzyně sice obsahuje více volacích znaků než Štefánik, ale pouze o 1.6% (o devět volacích znaků více), což neodpovídá tomu, o kolik se zhorší přepis volacích znaků letiště Ruzyně při *špatných* promptech oproti letišti Štefánik. Subset Zurich má vždy horší výsledky, což odpovídá tomu, že se jedná o neviděné letiště.

6.3.2 Experiment I. – testování modelů na datech bez promptu

V následujícím experimentu je cílem zjistit, jak velká je míra závislosti natrénovaných modelů na prompt. Tedy rodiny P-BASELINE, P-ALLDSATCOEN a P-ATCOEN obsahující modely adaptované na jednotlivé prompty jsou zde otestovány na datasetu ATCO2-EN **bez použití promptu**. Výsledky zobrazuje tabulka 6.6.

Rodina modelů	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
P-ALLDS	20.26	20.81	23.71	20.41	20.17	20.28	20.47
	↓ *Rozdíl proti P-ALLDS adapt. na dané typy promptů (6.3)*↓						
	(+2.41)	(+2.11)	(+4.95)	(+2.01)	(+0.22)	(+0.18)	(+0.06)
P-BASELINE	24.92	24.82	23.16	27.84	22.86	23.38	22.18
	↓ *Rozdíl proti P-BASELINE adapt. na dané typy promptů (6.3)* ↓						
	(+7.98)	(+6.57)	(+5.38)	(+10.49)	(+1.42)	(+1.67)	(+0.24)
P-ALLDSATCOEN	20.62	21.10	20.14	21.08	19.99	20.27	20.30
	↓ *Rozdíl proti P-ALLDSATCOEN adapt. na dané typy promptů (6.3)* ↓						
	(+2.97)	(+2.15)	(+0.94)	(+2.08)	(+0.18)	(+0.36)	(+0.08)

Tabulka 6.6: WER hodnoty (%) **testování bez promptu** modelů adaptovaných na jednotlivé typy promptu. V levém sloupci je název každé rodiny a v každém dalším sloupci jsou ukázány WER hodnoty adaptovaného modelu na konkrétní typ promptu. Druhé číselné řádky ukazují rozdíly oproti testům rodin P-ALLDS, P-BASELINE a P-ALLDSATCOEN s promptem (tabulka 6.3).

Srovnání modelů adaptovaných na prompty **AG_** vykazuje jistou anomálii, totiž že tyto modely v rodině P-BASELINE se při nepoužití promptů zhoršily mnohem více než ostatní modely. Zjevně zde tedy existuje korelace mezi použitím promptu a množstvím dat, na která je model adaptován.

Důležitým ukazatelem jsou malé rozdílové hodnoty ve sloupci 50CZB (modře označený). Ty dokazují, že se všechny modely správně adaptovaly na neužitečnost tohoto promptu a relativně dobře začaly ignorovat jeho dopad.

6.3.3 Experiment II. – všechny špatné vs. všechny dobré volací znaky

Druhý experiment ukazuje v tabulce 6.7, co se bude dít s modely, které byly adaptovány s promptem, který obsahoval buď samé v promluvě neexistující volací znaky (35B), nebo jenom existující (AG), pokud budou testovány na komplementárním promptu.

Rodina modelů	AG	35B
P-ALLDS	17.85	20.10
	↓ *Test na promptu 35B* ↓	↓ *Test na promptu AG* ↓
	20.44	18.45
	(+2.59)	(-1.65)
P-BASELINE	16.94	21.71
	↓ *Test na promptu 35B* ↓	↓ *Test na promptu AG* ↓
	25.78	19.83
	(+8.84)	(-1.88)
P-ALLDSATCOEN	17.65	19.91
	↓ *Test na promptu 35B* ↓	↓ *Test na promptu AG* ↓
	20.92	19.20
	(+3.27)	(-0.71)

Tabulka 6.7: WER hodnoty (%) testování modelů, které byly adaptované na prompt AG, s promptem 35B a obráceně – výsledné hodnoty testů na komplementárním promptu jsou označeny fialově. V prvním řádku pro každou rodinu modelů nalezneme hodnoty, kterých bylo dosaženo při trénování i testování na stejném promptu.

Očekávaně si pohoršily modely adaptované na prompt AG. Naopak modely trénované na 35B zvládly využít předanou informaci z promptu AG. To svědčí o tom, že prompt 35B obsahuje slova, která se v promluvě vyskytují, a tudíž se model naučil prompt nějak využít. Pokud ovšem porovnáme výsledky adaptovaných modelů na 35B při testech bez využití promptu (tabulka 6.6), můžeme pozorovat, že i tam modely reagují jen nepatrným zhoršením WER hodnot, které je sice větší než pro prompt 50CZB, nikoli však výrazně. To by nasvědčovalo hypotéze, že model se nikdy zjevně zcela neodnaučí prompt ignorovat a při předání smysluplné informace v něm ji využije.

6.3.4 Experiment III. – stabilní vs. nestabilní pozice volacího znaku

Pro trojici rodin modelů P-ALLDS, P-BASELINE a P-ALLDSATCOEN byla provedena experimentální adaptace na prompt AG35B. Tento prompt měl vždy následující formu:

[správné volací znaky],[špatné volací znaky]

– tedy správné volací znaky se vyskytovaly **vždy** na prvním místě v tomto typu promptu. Cílem bylo otestovat, zda se model bude fixovat na pozici správných volacích znaků a jak bude reagovat v případě změny této pozice. To je reflektováno v tabulce 6.8 s WER hodnotami přepisů a v tabulce 6.9 s WER hodnotami volacích znaků.

Modely adapt. na AG35B se stabilní pozicí správných volacích znaků	**Test na AG35B, stabilní pozice správných volacích znaků**	**Test na AG35B, randomizovaná pozice správných volacích znaků**
AG35B-ALLDS	18.11	20.14
AG35B-BASELINE	16.89	23.33
AG35B-ALLDSATCOEN	18.83	19.48

Tabulka 6.8: WER hodnoty (%) přepisů. Testované modely byly adaptovány na prompt AG35B (vznikly modely AG35B-ALLDS, AG35B-BASELINE a AG35B-ALLDSATCOEN), ve kterém měly správné callsigny stabilní pozici (nacházely se na začátku promptu). WER hodnoty v tabulce se pak týkají testů na promptech se stabilní a randomizovanou pozicí správných volacích znaků.

Modely adapt. na AG35B se stabilní pozicí správných volacích znaků	**Test AG35B, stabilní pozice správných volacích znaků**	**Test AG35B, *randomizovaná pozice správných volacích znaků***
AG35B-ALLDS	6.63	11.81
AG35B-BASELINE	3.01	18.09
AG35B-ALLDSATCOEN	10.19	12.76

Tabulka 6.9: WER hodnoty (%) volacích znaků. Testované modely byly adaptovány na prompt AG35B, ve kterém měly správné callsigny stabilní pozici (nacházely se na začátku promptu). WER hodnoty (%) v tabulce se pak týkají testů na promptech se stabilní a randomizovanou pozicí správných volacích znaků.

Pozorovat opět můžeme velké rozdíly pro model AG35B-BASELINE, který se nejvíce přizpůsobil stabilní pozici správných volacích znaků. Proti tomu model AG35B-ALLDSATCOEN vykazuje největší schopnost vzít si z promptu potřebné informace, i když byl naučen na fixní pozici skutečně se vyskytujících volacích znaků v promluvě.

6.3.5 Experiment IV. – jiné prompty

Protože jednotlivé prompty, vyjma AG, obsahují náhodně vygenerované části, je vhodné otestovat případ změny těchto promptů. Proto byl pro celou testovací sadu ATCO2-EN vygenerován nový set promptů. Nezměněným zůstává prompt AG, protože se skládá ze skutečně se vyskytujících volacích znaků v promluvách, které se nemění. V tabulce 6.10 sledujeme WER hodnoty přepisů pro dva sety promptů pro rodinu modelů P-BASELINE.

	Rodina modelů P-BASELINE						
	AG	**AG4B**	**AG35B**	**AG50CZB**	**5B**	**35B**	**50CZB**
PROMPT DS I.							
Štefánik	13.13	15.49	13.58	13.58	16.12	18.55	16.57
Ruzyně	14.87	16.59	15.98	15.93	18.98	18.42	19.87
Zurich	18.16	19.16	19.02	18.41	23.05	23.07	23.46
Celkem WER	16.94	18.25	17.78	17.35	21.44	21.71	21.94
PROMPT DS II.							
Štefánik	13.13	14.72	13.70	14.53	16.06	17.91	16.51
Ruzyně	14.87	16.20	15.70	16.70	18.92	19.03	20.03
Zurich	18.16	19.75	20.01	18.42	25.87	23.02	23.57
Celkem WER	16.94	18.49	18.46	17.61	23.41	21.68	22.03
Rozdíl **II.** – **I.** celkové WER přepisů	0.00	(+0.24)	(+0.68)	(+0.26)	(+1.97)	(-0.03)	(+0.09)

Tabulka 6.10: WER hodnoty (%) přepisů pro testování rodiny P-BASELINE na dvou různých setech promptů. PROMPT DS I. obsahuje prompty používané na všech ostatních místech v této práci, PROMPT DS II. je použitý pouze zde pro srovnání. Oranžově je označen řádek s rozdílem celkových WER hodnot pro oba sety promptů.

Zcela očekávaně došlo k drobným změnám ve WER hodnotách – ty dokazují, že přidané informace v promptu jsou pro model užitečné a do značné míry na nich závisí celková přesnost přepisu. Při podrobnějším prohlížení tabulky vidíme, že někdy dojde ke zlepšení pro jednotlivé subsety sady ATCO2-EN, zatímco ostatní se zhorší. Nula pro rozdíl týkající se testování promptu AG je v tabulce pouze kontrolní.

Prompt AG50CZB neočekávaným způsobem negativně ovlivňuje přepis týkající se letišť Štefánik a Ruzyně, zatímco Zurich zůstává prakticky nedotčený. Z toho vyplývá odhad, že přestože jsou data označená jako anglická pro všechna tři letiště, Štefánik a Ruzyně obsahují více útržků libovolných českých promluv (*„ahoj“*, *„díky“*, atp.) a díky tomu jsou jejich přepisy ovlivněny i náhodnými slovy, která jsou v češtině.

Tabulka 6.11 zaznamenává výsledné **WER hodnoty volacích znaků** pro testování rodiny modelů P-BASELINE na dvou setech promptů. Pozorovat můžeme také celkový počet zcela správně přepsaných volacích znaků.

	Adaptace modelu P-BASELINE						
	AG	**AG4B**	**AG35B**	**AG50CZB**	**5B**	**35B**	**50CZB**
PROMPT DS I.							
Štefánik	2.01	4.47	3.35	2.22	9.54	8.49	11.82
Ruzyně	0.63	2.25	3.88	2.20	17.12	12.09	16.38
Zurich	3.11	6.48	6.29	5.08	18.54	17.21	20.16
Celková WER vol. zn.	2.64	5.66	5.55	4.31	17.24	15.44	18.62
Celkem správně přepsaných vol. zn.	719	656	659	674	422	458	415
PROMPT DS II.							
Štefánik	2.01	3.32	4.83	3.75	9.60	8.63	11.29
Ruzyně	0.63	2.86	3.33	2.49	17.36	12.44	16.57
Zurich	3.11	7.39	6.18	5.07	20.00	16.85	20.23
Celková WER vol. zn.	2.64	6.28	5.63	4.56	18.36	15.24	18.63
Celkem správně přepsaných vol. zn.	719	657	652	672	414	461	419
Rozdíl **II.** – **I.** celkové WER vol. zn.	0.00	(+0.62)	(+0.08)	(+0.25)	(+1.12)	(-0.20)	(+0.01)
Rozdíl **II.** – **I.** počtu správně přepsaných vol. zn.	0	+1	-7	-2	-8	+3	+4

Tabulka 6.11: WER hodnoty (%) volacích znaků pro testování rodiny P-BASELINE na dvou různých setech promptů. PROMPT DS I. obsahuje prompty používané na všech ostatních místech v této práci, PROMPT DS II. je použitý pouze zde. Oranžově je označen řádek s rozdílem celkových WER hodnot volacích znaků pro oba sety promptů, modře pak řádek s rozdílem správně přepsaných volacích znaků.

Opět pozorujeme drobné změny ve WER hodnotách, které ovšem nekorespondují zcela se změnou počtu správně přepsaných volacích znaků (například vidíme, že pro AG35B je sice rozdíl WER hodnot pouze 0.08, ale pro DS II. je o sedm méně bezchybně přepsaných promptů). Prompt AG35B má ze všech promptů AG_ celkem nejvíc negativní vliv na metriku zcela správně přepsaných promptů – příčinu bychom mohli hledat v tom, že AG35B může obsahovat volací znaky, které se liší například pouze v jednom slově od toho, který se skutečně v promluvě vyskytuje. To by potom mohlo zapříčinit zmatení modelu při přepisu volacích znaků.

6.4 Závěr

V této kapitole bylo provedeno trénování trojice výchozích modelů ALLDS, ALLDSATCOEN a BASELINE (Vanilla Whisper Medium) na všechny dostupné typy promptu. Vznikly tak tři rodiny modelů P-ALLDS, P-ALLDSATCOEN a P-BASELINE.

Testování pak probíhalo na sadě ATCO2-EN. Každý model byl testován s použitím toho samého typu promptu, na který byl trénován.

Vůbec nejlepší dosažená WER hodnota přepisu byla 16.94 %, získaná modelem AG-BASELINE. Proti modelu ATCOEN (trénovaný na stejných datech jako BASELINE, ale bez promptu) tak došlo ke zlepšení o 5.54 %.

Při trénování na promptu AG35B byla nejnižší WER hodnota 17.78 % získaná opět modelem AG-BASELINE, s rozdílem proti ATCOEN o 4.7 %. Tento výsledek je důležitý, protože prompt AG35B je v této práci považován za využitelný v praxi.

První experiment ukázal, že použití promptu AG neovlivňuje jen volací znaky, ale i další části promluvy.

Další z experimentů se týkal trénování modelů na promptu AG35B se stabilní pozicí správných volacích znaků v něm. Natrénované modely byly vzápětí testovány jak s promptem s randomizovanou pozicí správného volacího znaku, tak s pozicí stabilní. Při užití randomizovaného promptu AG35B se ukázalo, že zatímco model P-ALLDSATCOEN se zhoršil jenom o 0.65 % WER, model P-BASELINE se zhoršil o 6.44 % WER. Z toho lze vyvodit závěr, že množství dat, na kterých je model trénován ovlivňuje jeho stabilitu proti změnám, které nastanou ve fázi testování proti trénování. Tedy i když je na začátku kapitoly z grafů ukazujících evaluace jednotlivých epoch zřejmé, že se modely z rodiny P-ALLDSATCOEN obecně výrazně nezlepšují, zvyšuje jejich natrénovanost na velkém množství dat jejich robustnost.

Při experimentování na dvou různých setech promptů vykázaly modely natrénované na AG50CZB výrazněji vyšší WER hodnoty na subsetech Štefánik a Ruzyně. To vedlo k hypotéze, že Štefánik a Ruzyně mohou obsahovat více útržků českých promluv, než je tomu u letiště Zurich, a proto jsou českými slovy v promptech více ovlivněny. Když došlo k jiné volbě slov v promptu AG50CZB, mohly být výsledky výrazněji ovlivněné těmito slovy.

Kapitola 7

Finetuning III. – Adaptace Whisperu na zkrácené přepisy bez a s použitím promptu

Formát zde použitých dat v této kapitole je zkrácený, tzn. všechny volací značky, slova představující leteckou abecedu či číslovky, jsou přepsána na zkrácenou formu.

7.1 Vyhodnocení základních modelů natrénovaných bez promptu

Adaptace základních modelů, tj. bez použití promptu, proběhla na stejných typech dat a stejných dvou skupinách hyperparametrů, jako tomu bylo u plných přepisů. Rovněž konvence zápisu zůstává pro zkrácené přepisy identická a budeme zde hovořit opět o modelech BASELINE, ALLDS, ATCOEN a ALLDSATCOEN.

Testovací i trénovací sadou zůstává dataset ATCO2-EN a některé výsledky obsažené v této sekci ukazují hodnoty WER metriky zvlášť pro subsety této sady (Ruzyně, Štefánik a Zurich).

7.1.1 Testování modelů na datech bez promptu

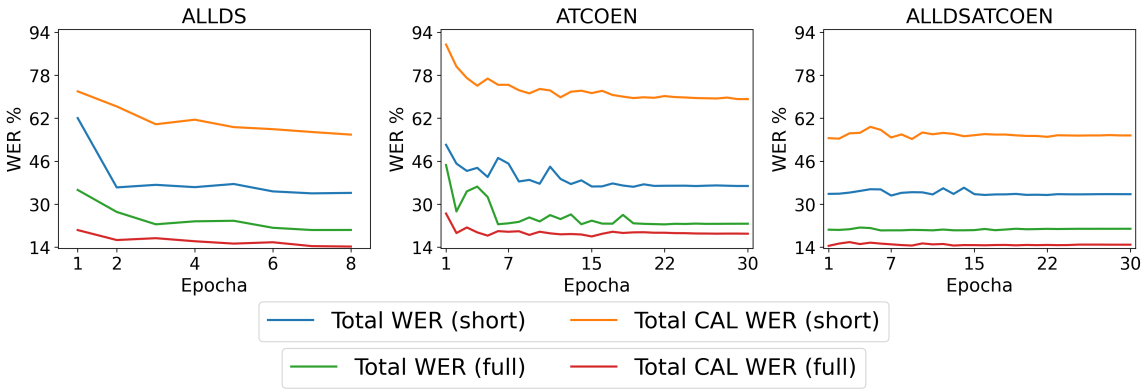
Tabulka 7.1 ukazuje epochy s nejnižší WER hodnotou, kterých dané modely dosáhly pro zvolené skupiny trénovacích hyperparametrů. Model ATCOEN dosáhl nejnižší hodnoty WER o tři epochy dříve než u plných přepisů. Model ALLDS se choval z tohoto pohledu identicky pro plné i zkrácené přepisy a ALLDSATCOEN dosáhl nejnižší hodnoty WER o epochu později, než tomu bylo pro plné přepisy.

Model	I. sk. par.		II. sk. par.		Celkem trénovacích epoch
	WER %	ep	WER %	ep	
ALLDS	34.02	7	35.51	6	8
ATCOEN	36.48	19	39.31	14	30
ALLDSATCOEN	34.02	1	33.22	7	30

Tabulka 7.1: Nejlepší získané WER hodnoty modelů pro danou skupinu hyperparametrů s uvedenou epochou, během které bylo hodnot dosaženo. Zvýrazněny jsou nejnižší hodnoty. **Modely jsou do trénování s promptem převzaté právě v těchto epochách.** Při dalším zmiňování modelů ALLDS, ATCOEN a ALLDSATCOEN již budeme mít na mysli ty verze, které dosáhly nejnižších hodnot WER tak, jak jsou vyznačeny v této tabulce.

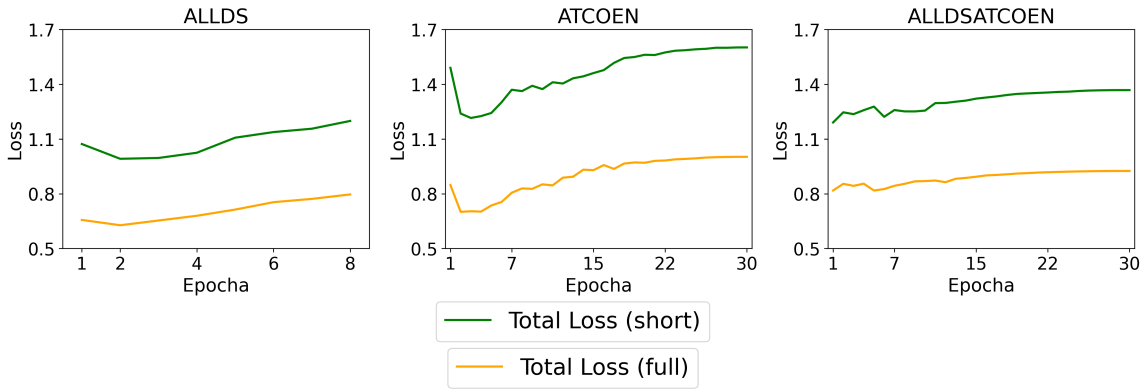
Obdobně jako pro plné přepisy i zde první skupina hyperparametrů dosahuje nejlepších výsledků pro modely ALLDS a ATCOEN. Druhá skupina hyperparametrů s nižší hodnotou hyperparametru *learning rate* se osvědčila pro model ALLDSATCOEN. To koresponduje s faktem, že model ALLDSATCOEN již vyžaduje menší kroky v učení vzhledem k použitým datům a počtu epoch.

Níže jsou ukázány srovnávací grafy průběhu validačních WER hodnot přepisů, WER hodnot volacích znaků a loss funkcí pro celý dataset ATCO2-EN pro plné i zkrácené přepisy.



Obrázek 7.1: Srovnání celkového průběhu WER hodnot přepisů a volacích znaků pro modely ALLDS, ATCOEN a ALLDSATCOEN pro plné a zkrácené přepisy.

Z grafů 7.1 můžeme pozorovat, že v zásadě se průběhy všech funkcí chovají pro plné i zkrácené přepisy podobně. Pro model ALLDS pozorujeme rychlejší pokles WER hodnot zkrácených přepisů, a její zlom, kdy funkce začíná růst, nastává o epochu dříve, než je tomu pro plné přepisy. Přitom pro ten samý model trvá naopak delší dobu (opět o jednu epochu) klesání WER hodnoty přepisu volacích znaků, než dojde ke zlomu a mírnému růstu. Průběh funkcí pro modely ATCOEN a ALLDSATCOEN je pak téměř identický při srovnání plných a zkrácených přepisů.



Obrázek 7.2: Srovnání celkového průběhu loss funkcí pro modely ALLDS, ATCOEN a ALLDSATCOEN pro plné a zkrácené přepisy.

Na grafech 7.2 vidíme opět velmi podobný průběh loss funkcí pro plné a zkrácené přepisy, který naznačuje, že trénování modelů má obdobný průběh pro obě formy přepisů.

V tabulce 7.2 nalezneme pro modely ALLDS, ATCOEN a ALLDSATCOEN čísla epoch, ve kterých dosáhly tyto modely nejnižších WER hodnot během trénování na viděných letištích Ruzyně a Štefánik a na neviděném letišti Zurich. Pro srovnání je přidán i sloupec s celkově nejlepší WER hodnotou pro celou testovací sadu.

Model	**Štef.+Ruzyně**		**Zurich**		**AVG rozdíl**	**Celkem**	
	WER %	ep	WER %	ep	WER %	WER %	ep
ALLDS	30.40	7	35.29	7	7.37	34.02	7
ATCOEN	32.62	19	37.63	19	5.30	36.48	19
ALLDSATCOEN	29.60	14	34.41	7	4.86	33.22	7

Tabulka 7.2: Nejnižší dosažené WER hodnoty a epochy, ve kterých jich bylo dosaženo, pro jednotlivé subsety testovací datové sady. Ve třetím sloupci je ukázán průměrný rozdíl WER hodnot mezi sadami Štefánik a Ruzyně (viděná letiště) a Zurich (neviděné letiště). Poslední sloupec opět pro přehlednost ukazuje celkově nejlepší WER hodnoty s epochami.

Následuje tabulka 7.3 uvádějící referenční hodnoty, od kterých budou v dalších krocích a experimentech počítány rozdíly.

Model	WER přepisů(%)	WER volacích zn. (%)	Správně přepsané vol. zn.
BASELINE	84.74	97.01	10 / 763 (1.31%)
ALLDS	34.02	56.86	304 / 763 (39.84%)
ATCOEN	36.48	69.50	210 / 763 (27.52%)
ALLDSATCOEN	33.22	54.83	320 / 763 (41.94%)

Tabulka 7.3: Porovnání úspěšnosti natrénovaných základních modelů na volacích znacích. V tabulce pozorujeme i vyhodnocení modelu **BASELINE**, tedy Vanilla Whisper Medium. Správně přepsaným volacím znakem je myšlena naprostá shoda mezi jeho formou v referenčním a predikovaném přepisu.

7.1.2 Testování modelů na datech s promptem

Tabulka 7.4 ukazuje výsledky testování základních modelů na datech ATCO2-EN s využitím každého druhu promptu.

Model	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
BASELINE	72.37	73.42	75.15	79.87	85.69	84.26	86.14
	(-12.37)	(-11.32)	(-9.59)	(-4.87)	(+0.95)	(-0.48)	(+1.40)
ALLDS	28.91	30.59	31.68	32.02	34.26	33.91	35.77
	(-5.11)	(-3.43)	(-2.34)	(-2.00)	(+0.24)	(-0.11)	(+1.75)
ATCOEN	30.67	31.88	33.63	33.25	35.94	36.09	36.52
	(-5.81)	(-4.60)	(-2.85)	(-3.23)	(-0.54)	(-0.39)	(+0.04)
ALLDSATCOEN	28.89	29.80	31.78	31.12	33.66	33.89	34.80
	(-4.33)	(-3.42)	(-1.44)	(-2.10)	(+0.44)	(+0.67)	(+1.58)

Tabulka 7.4: WER hodnoty (%) přepisů pro základní modely, které byly testované na všech typech promptů. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě: BASELINE (84.74 %), ALLDS (34.02 %), ATCOEN (36.48 %), ALLDSATCOEN (33.22 %) – zeleně zlepšení (nižší WER), červeně zhoršení.

Zajímavé jsou rozdíly proti této tabulce u plných přepisů 5.5. Model BASELINE dosahuje na zkrácených přepisech v průměru o 22 % horších výsledků než na plných přepisech, s výjimkou promptu 50CZB, kde je rozdíl pouze 5.18 %. Tento výsledek vypovídá o neužitečnosti promptu 50CZB pro model BASELINE, tedy netrénovaný Vanilla Whisper Medium.

Na rozdíl od plných přepisů si model BASELINE pohoršil při využití promptu 5B, a celkově mu prompt nepřinesl výrazné zlepšení ve srovnání s plnými přepisy.

Model ALLDS dosahuje na promptu AG35B o 2.96 % **lepšího** výsledku na zkrácených přepisech. Na dalších promptech je však horší na zkrácených přepisech a celkově horší výsledky pro zkrácené přepisy platí i pro modely ATCOEN a ALLDSATCOEN.

Natrénované modely ALLDS a ALLDSATCOEN měly celkově pozitivnější dopad více typů promptů na zkrácené přepisy než u plných přepisů. Největší rozdíl se projevil u modelu ALLDS, kterému pro plné přepisy pomohl pouze prompt AG, zatímco na zkrácených přepisech mu zlepšily výsledky téměř všechny prompty kromě 5B a 50CZB. Model ALLDSATCOEN si vedl obdobně.

Zlepšení modelu ATCOEN se ukázalo při použití stejných typů promptů pro zkrácené i plné přepisy.

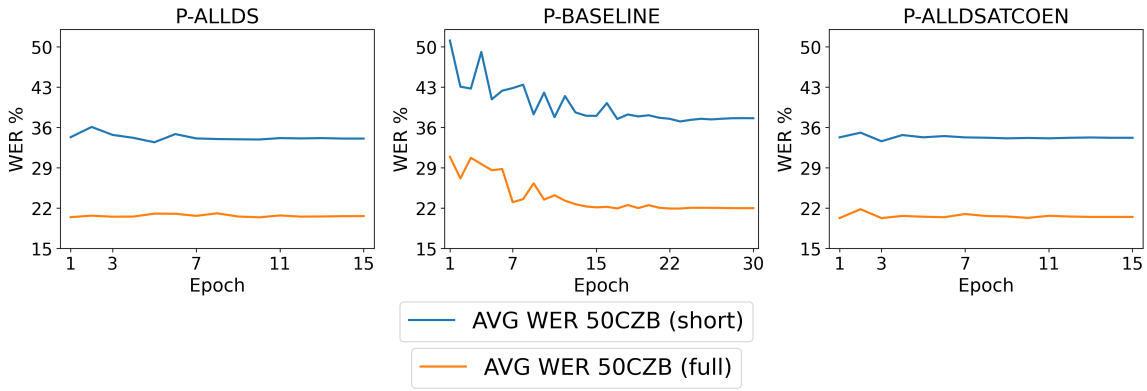
7.2 Vyhodnocení modelů natrénovaných s promptem

Poslední sekce se týká dotrénování a testování základních modelů na prompt, získaných v předchozí sekci. Budeme používat stejnou konvenci pojmenování modelů, jaká byla zavedena pro plné přepisy (tedy například P-BASELINE označuje rodinu modelů vycházející z modelů BASELINE a natrénovanou na jednotlivé typy promptu).

Každý z modelů byl natrénován na všech sedm zmíněných typů promptů (jedná se o dataset ATCO2-EN). Pro trénování rodiny P-BASELINE (Vanilla Whisper Medium) bylo využito trénovacích hyperparametrů I. skupiny s nastavením na třicet epoch. Modely rodiny P-ALLDS, tedy Vanilla Whisper Medium natrénovaný na všech datasetech, byl nyní dotrénován s parametry II. skupiny po dobu patnácti epoch. Modely rodiny P-ALLDSATCOEN, což je ALLDS dotrénovaný čistě na ATCO2-EN datech, byly trénovány s parametry II. skupiny po dobu patnácti epoch.

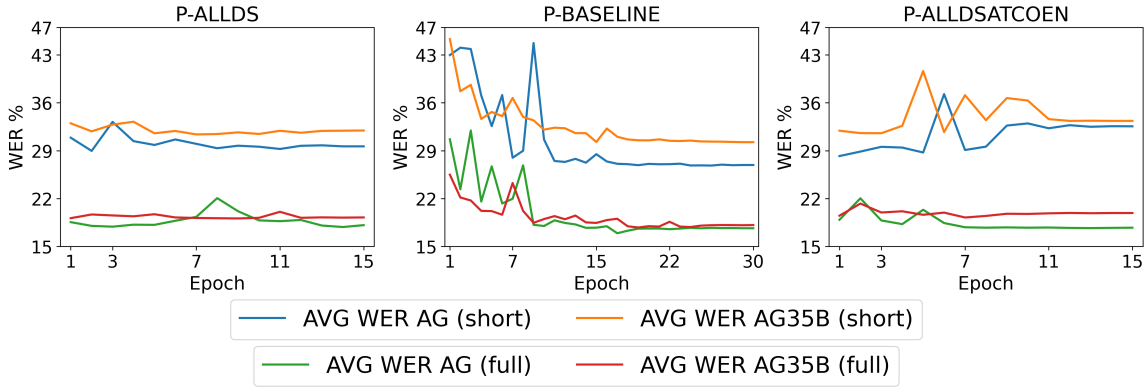
7.2.1 Testování modelů na datech s promptem

Na grafech 7.3 můžeme vidět srovnání průběhu WER hodnot pro modely natrénované na 50CZB pro plné i zkrácené přepisy. Průběhy WER hodnot jsou srovnatelné pro plné i zkrácené přepisy, což je očekávané, protože nemají pro modely téměř žádnou hodnotnou informaci. Přesto lze vidět mírně větší kolísání WER hodnot pro zkrácené přepisy. To může být způsobeno skladbou promptu 50CZB, který obsahuje pouze slova oddělená čárkami, a tedy může docházet k matení modelu při zkracovaní volacích znaků (to u plných přepisů nenastává).



Obrázek 7.3: Průběh WER hodnot zkrácených přepisů pro modely natrénované na prompt 50CZB společně s hodnotami pro obdobné modely, trénované na plné přepisy.

Grafy 7.4 ukazuje srovnání průběhu WER hodnot modelů trénovaných na promptech AG a AG35B. Modely na zkrácených přepisech mají horší průběhy těchto hodnot, zejména se zhoršují WER hodnoty v průběhy epoch pro rodiny modelů P-ALLDSATCOEN. Zde se již určitě může hovořit o přetrénování (WER hodnoty v grafu ukazující rodinu P-ALLDSATCOEN narostly o cca 5 %). Je tedy možné, že zkrácené formy přepisů vedou modely, které jsou natrénované na větším množství dat, rychleji k bodu přetrénování, než tomu bylo pro plné přepisy.



Obrázek 7.4: Průběh WER hodnot zkrácených přepisů pro modely natrénované na promptech AG a AG35B, společně s hodnotami pro obdobné modely, trénované na plné přepisy.

Dále je ukázána tabulka 7.5 WER hodnot všech natrénovaných modelů na různé typy promptu na zkrácené formě přepisů.

Rodina modelů	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
P-ALLDS	28.97	29.90	31.38	30.52	33.08	33.82	33.44
	– Rozdíl proti ALLDSATCOEN bez promptu (WER 33.22 %)						
	(-4.25)	(-3.32)	(-1.84)	(-2.70)	(-0.14)	(+0.60)	(+0.22)
	– Rozdíl proti ALLDSATCOEN testovanému s promptem (WER viz. 7.4)						
	(+0.08)	(+0.10)	(-0.40)	(-0.60)	(-0.58)	(-0.07)	(-1.36)
P-BASELINE	26.83	28.75	30.25	28.18	36.63	36.19	37.05
	– Rozdíl proti ATCOEN bez promptu (WER 36.48 %)						
	(-9.65)	(-7.73)	(-6.23)	(-8.30)	(+0.15)	(-0.29)	(+0.57)
	– Rozdíl proti ATCOEN testovanému s promptem (WER viz. 7.4)						
	(-3.84)	(-3.13)	(-3.38)	(-5.07)	(+0.69)	(+0.10)	(+0.53)
P-ALLDSATCOEN	28.22	30.40	31.56	30.59	33.49	33.11	33.61

Tabulka 7.5: WER hodnoty (%) modelů trénovaných s prompty na zkrácených přepisech ATCO2-EN datasetu, testovaných s použitím promptů. Každý model je testován na tom promptu, na jakém byl trénován. Druhý řádek pro každý model obsahuje rozdíl oproti referenční hodnotě modelu trénovaného bez promptu – zeleně zlepšení, červeně zhoršení. Pro modely P-ALLDS je referenční hodnotou WER 33.22 % získaná modelem ALLDSATCOEN, pro P-BASELINE potom WER 36.48% získaná modelem ATCOEN. Oranžově je označen sloupec pro modely trénované a testované na promptu AG, modře pak sloupec s modely týkající se promptu AG35B. Třetí řádky srovnávají výsledky testů základních modelů ALLDSATCOEN a ATCOEN na datech s promptem (viz tabulka 7.4), proti výsledkům rodin modelů P-ALLDS a P-BASELINE. K modelům P-ALLDSATCOEN neexistuje ekvivalentně trénovaný referenční model (model trénovaný na všech dostupných datasetech a poté dvakrát na ATCO2-EN není dostupný) a tudíž pro něj nejsou zobrazena srovnání.

Ve výsledcích je zvláštní větší počet kladných rozdílových hodnot, tedy těch, které ukazují zhoršení a kterých je více než u modelů pro plné přepisy. Zároveň je zvláštní, že modely 50CZB-ALLDS a 50CZB-BASELINE se zhoršily proti svým srovnatelným modelům trénovaným bez promptu. Namístě je hypotéza, že modely na zkrácených přepisech dokázaly při trénování čerpat z tohoto typu promptu a jeho nevhodná skladba (tedy náhodný výběr slov), která může nastat v datech pro testování, negativně ovlivní přepis.

Další zajímavostí je fakt, že modely AG-ALLDS a AG4B-ALLDS dosáhly horších výsledků, než pokud byly prompty AG a AG4B použity na model ALLDS bez předchozího natrénování na ně.

7.2.2 Testování modelů na datech bez promptu

V této sekci je proveden test natrénovaných modelů s promptem na datech bez promptu. Důvodem je opět snaha zjistit, jak moc jsou modely závislé na existenci promptu. Výsledky jsou uvedeny v tabulce 7.6.

Rodina modelů	AG	AG4B	AG35B	AG50CZB	5B	35B	50CZB
P-ALLDS	34.23	34.72	33.87	33.66	33.86	34.27	34.23
	↓ *Rozdíl proti P-ALLDS adapt. na dané typy promptů (7.5)* ↓						
	(+5.26)	(+4.82)	(+2.49)	(+3.14)	(+0.78)	(+0.45)	(+0.79)
P-BASELINE	43.33	42.61	40.93	40.39	37.40	40.44	37.91
	↓ *Rozdíl proti P-BASELINE adapt. na dané typy promptů (7.5)* ↓						
	(+16.50)	(+13.86)	(+10.68)	(+12.21)	(+0.77)	(+4.25)	(+0.86)
P-ALLDSATCOEN	33.78	34.09	39.53	34.21	33.76	33.36	33.82
	↓ *Rozdíl proti P-ALLDSATCOEN adapt. na dané typy promptů (7.5)* ↓						
	(+5.56)	(+3.69)	(+7.97)	(+3.62)	(+0.27)	(+0.25)	(+0.21)

Tabulka 7.6: WER hodnoty (%) získané z testování bez promptu modelů adaptovaných na jednotlivé typy promptu. Druhé číselné řádky ukazují rozdíly oproti testům stejných modelů s promptem.

Pozorujeme stejný trend zhoršování WER hodnot jako pro plné přepisy, přičemž nejvíce signifikantní rozdíly zaznamenala rodina modelů P-BASELINE. U většiny promptů této rodiny jsou navíc ukázané rozdílové hodnoty dvojnásobné proti rozdílovým hodnotám na plných přepisech. Tedy modely P-BASELINE jsou mnohem více závislé na jednotlivých promptech, než je tomu u plných přepisů.

Pro modely trénované na promptu 50CZB se projevilo do velké míry jejich ignorování tohoto promptu a WER hodnoty se příliš nezměnily.

7.3 Závěr

V této kapitole bylo provedeno trénování i testování modelů na zkrácených přepisech datasetu ATCO2-EN s promptem i bez něj (při trénování modelů bez promptu byly využity i ostatní datasety, konkrétně pro modely ALLDS a ALLDSATCOEN).

Celkově dosahovaly všechny trénované modely horších výsledků, než jejich protějšky trénované na plných přepisech. To bylo očekávané, protože model Whisper je ve své základní variantě trénován na plné přepisy.

Ukázalo se, že v mnoha případech jsou trendy a rozdíly výsledků na zkrácených přepisech obdobné jako u plných přepisů.

Byly zaznamenány velké rozdíly WER hodnot pro rodinu modelů P-BASELINE při testování bez promptu, což ukazovalo velkou závislost této rodiny na použití promptu. Rozdíl WER hodnot při testování rodiny P-BASELINE s promptem a bez něj byl několikrát téměř dvojnásobný, než jaké byly hodnoty takto získaných rozdílů při plných přepisech.

U modelů AG35B-ALLDSATCOEN a AG-ALLDSATCOEN se projevilo přetrénování, které se ukázalo růstem WER hodnot v průběhu trénovacích epoch, což u plných přepisů takovým výrazným způsobem nenastalo.

Kapitola 8

Závěr

V této práci byla stručně uvedena problematika a princip fungování neuronových sítí, zejména transformerů. Popsán byl také samotný model Whisper a jeho práce s promptem. V této práci figuroval pouze Whisper verze Medium, na kterém probíhalo veškeré dotrénování – *finetuning*. Pozornost byla též věnována problematice letecké komunikace, zpracování datasetů a tvorbě promptů.

Ústředním bodem této práce bylo vyzkoušet dopad použití promptu na zlepšení přepisu volacích znaků vyskytujících se v promluvách. Vedle vyhodnocování přepisu volacích znaků se však zkoumalo i přispění promptu k celkové kvalitě přepisu. Cíleně byl vytvořen prompt typu AG35B (složený z volacích znaků, které se v promluvě vyskytují a z třiceti pěti volacích znaků, které v promluvě jistě obsaženy nejsou), který měl reprezentovat prompt hypoteticky využitelný ve skutečném provozu.

Model trénovaný bez promptu v prvotní adaptaci Whisperu poukázal na užitečnost trénování na všech datasetech zpracovaných v této práci. Zvlášť jeho výsledky na datasetu ATCO2 byly lepší, než kterých dosáhl model trénovaný pouze na sadě ATCO2. Proto byly pro trénování s promptem zahrnuty modely předtrénované na všech datasetech.

Aby byla zajištěna konzistence dat při trénování a vyhodnocování, byly všechny datasety převedeny do formy *lower-case*.

Kapitola Finetuning I. se věnovala tréninku modelů bez promptu na plné formě přepisů dat. Natrénovaná byla trojice modelů, každý z nich na dvou sadách hyperparametrů. Nejnižší dosažené hodnoty WER jednotlivými modely na datech ATCO2-EN byly považovány za referenční pro trénování modelů s promptem. Nejnižší WER hodnoty, konkrétně 20.24 %, bylo dosaženo modelem ALLDSATCOEN, který byl trénován nejprve na všech dostupných datasetech a následně pouze na ATCO2-EN. Při provedeném testování s promptem na modelech, které na něj adaptovány nebyly, se projevilo očekávané zlepšení WER hodnot přepisů pro model BASELINE (Vanilla Whisper Medium). Dále se ukázalo, že jediný prompt, který pomohl bez předchozí adaptace všem modelům, byl prompt typu AG, který obsahoval všechny v promluvě se vyskytující volací znaky. Ostatní typy promptu měly různý vliv na jednotlivé modely. Proběhla též testování s použitím promptu na dvou různých epochách modelu ALLDSATCOEN a ukázalo se, že ačkoliv zvolené epochy dosahovaly WER s rozdílem pouze 0.03 %, rozdíly na některých typech promptu tvořily přes 8 % WER.

V kapitole Finetuning II. byla pozornost zaměřená na trénování modelů s použitím různých typů promptu. Byly zvoleny tři výchozí modely – ALLDS a ALLDSATCOEN, které byly natrénované bez promptu v kapitole Finetuning I., a BASELINE (Vanilla Whisper Medium). Jeden z důležitých závěrů testování byl získán v prvním provedeném experimentu a týkal se rodiny modelů P-ALLDSATCOEN. Šlo o to, že ačkoliv se modely při dotrénování

na prompt nijak dramaticky mezi epochami nelepšily, došlo ve WER hodnotách pouze k malým změnám, když byly modely testovány bez promptu. Proti tomu rodina modelů P-BASELINE vykázala významně klesající WER hodnoty při evaluaci epoch a dosáhla s některými prompty velmi dobrých výsledků: 16.94 % WER na promptu AG (což byla vůbec nejlepší WER hodnota dosažená v této práci na ATCO2-EN datech) a 17.78 % WER na promptu AG35B. Když však byla rodina modelů P-BASELINE testována bez promptu, došlo zde k markantnímu propadu WER hodnot. Ukázalo se tedy, že množství a různorodost použitých dat při trénování s promptem mají vliv na stabilitu přepisu, když prompt modelu dodán není.

Ve druhém experimentu byly natrénované modely 35B-ALLDS (tzn. výchozí model ALLDS dotrénovaný s promptem 35B), 35B-BASELINE a 35B-ALLDSATCOEN otestovány s promptem AG. Ukázalo se, že modely nedokázaly promptu využít v takové míře jako modely, které na prompt AG byly trénovány. Z toho lze usuzovat, že během trénovacího procesu se model v případě promptu 35B naučil, že prompt obsahuje informace, jejichž výskyt v přepisech není zaručen, a proto jim nemá věnovat takovou pozornost. To však vedlo k neschopnosti využít plně prompt AG. Na druhé straně došlo při testování na prompt 35B u modelu AG-BASELINE k zhoršení přepisů o 8.84 %, což bylo mnohem víc, než u modelů AG-ALLDS a AG-ALLDSATCOEN. Opět se tak projevila silná navázanost modelů P-BASELINE na typ promptu použitý při trénování.

Ve třetím experimentu byl použit pro trénování prompt AG35B se stabilní pozicí správných volacích znaků. Při testování s randomizovaným promptem AG35B modelů, které byly natrénovány na AG35B se stabilní pozicí správných volacích znaků, vykázal model AG35B-ALLDSATCOEN minimální zhoršení WER hodnot přepisů i volacích znaků. V kontrastu k němu se velmi zhoršil model AG35B-BASELINE, jehož WER hodnoty přepisů vzrostly o 6.5 % a WER hodnoty volacích znaků o 15 %. Projevila se tak odolnost modelů rodiny P-ALLDSATCOEN vůči adaptování se na konkrétní formát promptu použitého při trénování.

V posledním experimentu byla pro testování zvolena jiná sada promptů. Protože při vytváření některých typů promptů hrála roli náhoda, bylo důležité ukázat, že získané výsledky nejsou jen dílem dobře vygenerovaných hodnot. Na druhé testovací sadě dosáhly téměř všechny modely v průměru o 0.64 % horších WER hodnot. Jediného zlepšení bylo dosaženo při testování na promptu 5B, konkrétně o 0.03 % proti promptu 5B z původní testovací sady. Zmíněná zhoršení nebyla považována za významná a výsledky na první testovací sadě promptů by tedy neměly být dílem náhodného složení promptů.

Zkráceným přepisům nebyla věnována taková pozornost, většinově byly pouze výsledky modelů srovnávány s výsledky modelů trénovaných na plných přepisech. Projevila se opět vysoká závislost natrénované rodiny modelů P-BASELINE na existenci promptu – při testování bez jeho použití došlo k propadu WER hodnot až o 16 %. Modely rodiny P-ALLDSATCOEN tak velký propad nezaznamenaly, což ukazuje větší robustnost těchto modelů.

Z dosažených výsledků v této práci lze konstatovat, že obecně modelům použití promptu pomáhá pro kvalitnější přepis letecké komunikace. Efektivita promptu se však výrazně liší v závislosti na jeho typu, konzistenci a způsobu využití během trénování.

Zároveň se ukázalo, že větší množství různorodých dat během trénování přispívá k vyšší odolnosti modelu vůči absenci promptu při přepisu. Výsledky rovněž ukazují, že schopnost modelu využít prompt efektivně závisí na tom, jaký typ promptu byl použit při trénování, a do jaké míry jsou informace v promptu spolehlivé a relevantní k dané promluvě.

V budoucí práci by bylo vhodné zaměřit se na rozsáhlejší experimenty se strukturou a generováním promptů. Přínosem by mohlo být zkoumání dalších variant modelu Whisper a způsobů použití promptu. Dále by mohlo být užitečné vyhodnocovat také jiné aspekty kvality přepisu a využít jiné metriky.

Literatura

[1] Brownlee, J. *What is Teacher Forcing for Recurrent Neural Networks?* [online]. 8. Apr 2021 [cit. 15. 5. 2025]. Dostupné z: https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/.

[2] Cochard, D. *Prompt Engineering in Whisper* [online]. 28. Apr 2025 [cit. 28. 4. 2025]. Dostupné z: https://medium.com/axinc-ai/prompt-engineering-in-whisper-6bb18003562d.

[3] Gandhi, S. *Fine-tune whisper for multilingual ASR with huggingface transformers* [online]. Nov 2022 [cit. 18. 11. 2024]. Dostupné z: https://huggingface.co/blog/fine-tune-whisper.

[4] Grieco, J.; Benarousse, L.; Geoffrois, E.; Series, R.; Steeneken, H. et al. *N4: NATO Native and Non-Native Speech* [online]. 2006 [cit. 18. 5. 2025]. Dostupné z: https://catalog.ldc.upenn.edu/LDC2006S13.

[5] Jurafsky, D. a Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* [online]. 3rd. 2024. Dostupné z: https://web.stanford.edu/~jurafsky/slp3/.

[6] Kocour, M.; Veselý, K.; Blatt, A.; Gomez, J. Z.; Szöke, I. et al. Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition. In: *Proc. Interspeech 2021*. 2021, s. 3301–3305.

[7] Nevařilová, V. *Automatický přepis řeči letecké komunikace do textu* [online]. 2024. Dostupné z: https://theses.cz/id/zz4oa0/.

[8] OpenAI. *Introducing Whisper* [online]. 2022 [cit. 10. 5. 2025]. Dostupné z: https://openai.com/index/whisper/.

[9] Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C. et al. *Robust Speech Recognition via Large-Scale Weak Supervision* [online]. 2022. Dostupné z: https://arxiv.org/abs/2212.04356.

[10] Roberts, L. *Understanding the Mel Spectrogram* [online]. 2020 [cit. 20. 12. 2024]. Dostupné z: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

[11] SKYbrary Aviation Safety. *Waypoint* [online]. 2021 [cit. 16. 5. 2025]. Dostupné z: https://skybrary.aero/articles/waypoint.

[12] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L. et al. *Attention Is All You Need* [online]. 2023. Dostupné z: https://arxiv.org/abs/1706.03762.

[13] Wikipedia contributors. *List of airline codes* [online]. 2025 [cit. 27. 3. 2025]. Dostupné z: https://en.wikipedia.org/wiki/List_of_airline_codes.

[14] Yang, C.-K.; Huang, K.-P. a Lee, H.-y. Do Prompts Really Prompt? Exploring the Prompt Understanding Capability of Whisper. *ArXiv preprint arXiv:2406.05806* [online], jun 2024. Dostupné z: https://arxiv.org/abs/2406.05806.

[15] Šmídl, L. *Air Traffic Control Communication* [online]. 2011 [cit. 16. 4. 2025]. Dostupné z: https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-CCA1-0.

Příloha A

Příklad plného a zkráceného přepisu společně s vytvořenými prompty k těmto přepisům

Přepisy

Plný přepis:
`vista jet four three seven` `turn left via taxiway alpha echo six hold short runway one six`

Zkrácený přepis:
`vjt437` `turn left via taxiway ae6 hold short runway 16`

Vytvořené prompty

AG

Prompt (plný): `vista jet four three seven`
Prompt (zkrácený): `vjt437`

AG4B

Prompt (plný):

> `hotel bravo three three one two, flycom romeo india five one seven, vista jet four three seven, galaxy alpha one three, hotel bravo two five three four`

Prompt (zkrácený): `flori517, hblww, eugen17, vjt437, sia345`

AG35B

Prompt (plný):

> `swiss one nine six, tango echo two four, contract four seven three, flycom romeo india five one seven, habicht sierra whiskey, papa one six, legend one four one two, arcus air four two kilo mike, hotel bravo three three one`

two, affaires tchad five five, heliblue victor charlie, hotel bravo kilo mike x-ray, delta six nine zero seven, swiss one four echo, swiss two one eight lima, galaxy alpha one seven, euroguinea echo november, air portugal nine three one charlie, vista four three seven, swiss ambulance six three, euroguinea echo november nine, jet management six one three, air france one eight quebec victor, fraction one three three foxtrot, swiss air force five two one, air portugal nine three two, qatari zero two mike, hansa nine one victor, oscar echo india tango hotel, flycom romeo india alpha november four, galaxy alpha one three, vista jet four three seven, hotel kilo yankee one four, swiss one six one, uniform lima charlie seven seven, klm nine four foxtrot

Prompt (zkrácený):

florian4, rga01, dbold, swr218l, sejpk, lxa2f, aza571, hb2534, baw713, yupbb, d6907, tap931c, ane87uh, te22, hbymd, sia345, tcnka, diahg, eugen17, nje133f, hbkma, aho261x, axy1412, hblvc, thy6329, saz63, kondor11, om1010, ulc77, n842ts, ftc55, afr18qv, vjt437, uae87, kondor9, hbtsw

AG50CZB

Prompt (plný):

jasně, děje, rukama, toleranci, když, citlivá, štěstí, tvorbou, hudby, myšlenka, vědeckofantastického, zde, myslím, mají, právě, někde, vyjadřovací, jedná, převážně, činnost, znalosti, velkou, divadlu, porovnání, asteroidech, minulého, pedagogy, vzpomínám, snímáním, uvědomit, praktické, vyjadřování, lze, předvádění, ne, laserovým, každého, podobné, sond, a, zákonitosti, komentováním, žákem, vyhýbat, jednat, zdát, vista jet four three seven, dějinám, učitele, poťomkin, rozhovor

Prompt (zkrácený):

jasné, zdrojů, zprávy, meteor, konec, mladého, spolupráce, čeští, pořady, toto, celkem, celou, dialogu, kreslení, posluchačů, zájmem, tematikou, vyplynulo, proč, vyzkoušejí, odvrácené, nezničíme, společné, srovnání, bez, atd, využití, dopad, atomovou, kde, poradce, vjt437, širokého, cestování, zvuku, měsíce, g, atomy, předurčen, jehož, vědci, který, čase, podle, jsou, filmových, m, sama, spojené, projekci, rozhovor

50CZB

Prompt (plný):

sekvenci, čeští, zde, atomy, způsob, především, alfa, žánr, oživuje, pohybem, vyučující, montáži, ústavu, dokumentem, poslouží, asteroidů, překročit, prostě, filmových, během, je, zdroje, mnozí, nic, cestování, krůse, zpracování, měření, jediný, reflexi, opomíjený, také, reálném, laserového, dokázal, laboratoři, televizních, obrovský, dobrý, vyslat, neboť, vyhýbat, konec, tematikou, někde, úspěšné, plíživé, televizí, krůčky, rychlé

Prompt (zkrácený):

neustálým, rámeček, centru, lásky, nebo, děti, konkrétní, au, fyziky, jeden, gramotnosti, vyjadřovat, zajímavé, vyprchá, pohybové, čem, mladého, jiné, často, reálný, filmů, diskutuje, ukázky, dokončen, nějak, všechny, velikostí, řeči, podmínky, celých, posluchačů, být, techniky, krůčky, součinnost, sami, oko, důležité, osobě, kreslili, motorem, našem, své, musí, mezihvězdných, následovně, jedním, těch, případu, zdrojů

5B

Prompt (plný): `euroguinea echo november two zero, kilo oscar november delta oscar romeo one one, euroguinea echo november nine, euroguinea echo november one seven, lima delta x-ray eight`
Prompt (zkrácený): `dlh91v, gala17, qaj1553, uae88, baw713`

35B

Prompt (plný):

> `euroguinea echo november one three, galaxy alpha one three, hotel bravo two five three four, delta bravo oscar lima delta, hotel bravo yankee lima lima, royalair maroc three five two, a b g four nine two, affaires tchad five five, air france one eight quebec victor, alitalia four four mike, air hamburg two six one x-ray, air france six six two, bird seven one two, emirates eight eight, kilo oscar november delta oscar romeo nine, singapore three four five, sunturk three one hotel, arcus air four two kilo mike, fraction one three three foxtrot, swiss seven nine whiskey, contract four seven three, swiss one four echo, fraction one two two mike, heliblue whiskey whiskey, spicejet papa kilo, hotel bravo yankee mike delta, flycom romeo india alpha november four, jet management six one three, speedwing tango hotel, blue sky november alpha one, guja tango india two, galaxy alpha one seven, hotel bravo kilo mike alpha, etihad three nine bravo, flycom romeo india five one seven`

Prompt (zkrácený):

> `diana1, hbcqm, aab492, yupbb, saz63, aho261x, tap931c, swr218l, atl780, tap932, hbymd, hbypl, rga01, swr39, thy6329, nje122m, swr196, hb2534, nje133f, oeith, afr662, abr473, tcnka, d6907, te22, aze42km, hky14, sejpk, fhjre, axy1412, florian4, hbtsw, eugen, gala13, uae88`