VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Konvoluční neuronové sítě
Závěrečná zpráva

Členové týmu:

Michal Pyšík {xpysik00} (vedoucí týmu)

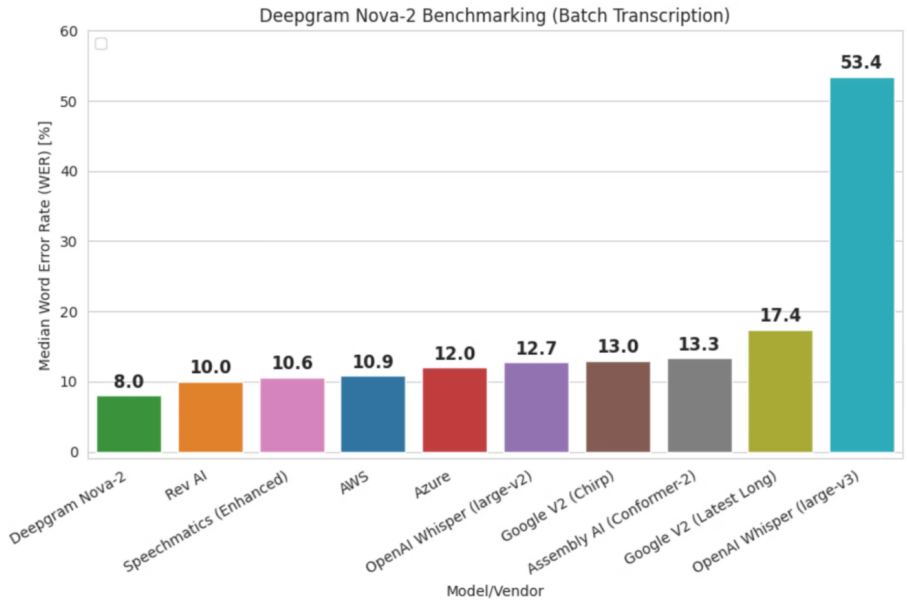
Maxim Plička {xplick04}

Michal Bartošák {xbarto0d}

Github repozitář: https://github.com/MichalPysik/KNN_project

1 Popis problému

Cílem našeho projektu je prozkoumat možnosti zlepšení modelu pro rozpoznávání řeči Whisper. Ačkoli je Whisper v současné době jedním z nejpoužívanějších nástrojů pro přepisování zvukových nahrávek, stále má svá úskalí. Jedním z aktuálně největších problémů aktuální verze Whisperu je problém halucinací.



Obrázek 1: Srovnání jednotlivých modelů pro rozpoznávání řeči z hlediska WER [1]

Vzhledem k tomu, že vědecká oblast nedospěla k jednotné ustálené definici tohoto pojmu, tak se v této práci budeme opírat o následující formulaci: Halucinace se jeví jako plynulé části výstupu, které na první pohled mohou působit věrohodně, ale ve skutečnosti nemají žádnou spojitost s původním nahrávkou [2]. Příklad takové halucinace může vypadat následovně:

- **Ground truth:** Someone had to run and call the fire department to rescue both the father and the cat.
- **Predikce:** Someone had to run and call the fire department to rescue both the father and the cat. **All he had was a smelly old ol' head on top of a socked, blood-soaked stroller.**

Takovéto výstupy pak mohou obsahovat nepřesné či zavádějící informace, které mohou zmást či dokonce oklamat koncového uživatele a z dlouhodobého hlediska mohou poškodit i samotné společnosti. Proto v tomto projektu budeme zkoumat vliv různých metod pro snížení výskytů halucinací v přepisovaných datech. K tomu bude potřeba vytvořit datovou sadu, na které bude zvolený model halucinovat. Tato datová sada bude nutná k tomu, abychom model na těchto datech mohli dotrénovat, případně upravit a zjistit, zda se vyhodnocování modelu na problematických datech zlepšilo. Na závěr se pokusíme navrhnout metodu pro detekci halucinací a analyzujeme různé přístupy, které použijeme k optimalizaci Whisperu tak, aby snížil svou náchylnost k halucinacím.

2 Související práce

Na problém halucinací jsme poprvé narazili v článku, který se zabýval automatickým přepisem zvukových nahrávek u osob trpících vadou řeči. Tento článek pro své vyhodnocení využíval jazykový model Whisper (zřejmě verzi 2), který u těchto nahrávek produkoval halucinace [4]. Po důkladné rešerši v rámci tohoto tématu jsme se zjistili, že na verzi modelu Whisper záleží. Článek *„Whisper-v3 Hallucinations on Real World Data“* [1] došel k závěru, že jeho poslední verze 3 (vydaná v listopadu roku 2023) produkuje data, která mají až 4x vyšší Word Error Rate než ostatní modely. Proto jsme se rozhodli používat právě tuto verzi.

Nejprve jsme se zaměřili na přípravu datasetu. Pro náš výzkum jsme chtěli využít dataset, který byl použit v článku s osobami trpícími vadou řeči, ale přístup k tomuto datasetu nám doposud nebyl poskytnut. Z tohoto důvodu jsme rozhodli prozkoumat další možnosti. Nakonec jsme zvolili tvorbu datasetu v rámci metody augmentace dat z článku [2], která bude blíže specifikována v sekci 3.

Dále jsme se zabývali vytvořením metody pro detekci halucinací, kterou jsme původně převzali z článku *„Hallucinations in Neural Automatic Speech Recognition: Identifying Errors and Hallucinatory Models“* [2]. Metoda využívala kombinací 3 metrik, jenž měly za úkol zjistit chybovost, sémantickou správnost a plynulost přepisu. Tento přístup se však neosvědčil, proto jsme rozhodli použít vlastní metody popsané v sekci 4.

3 Příprava datasetu a vyvolání halucinací

Nalezení datasetu, na kterém by model často halucinoval, bylo obtížnější, než jsme očekávali. Problém jsme nakonec překonali pomocí vlastní augmentace dat, která dokázala halucinace vyvolat. Rozhodli jsme se praocovat s korpusem zvukových nahrávek LibriSpeech[1] [5] (test set, „other“ speech), který obsahuje kratší, jazykově náročnější (přeci jen pracujeme s ”nejsilnější” vezí Whisperu) nahrávky v anglickém jazyce (celkem 2939 nahrávek, jehož přepis nám zabral na grafické kartě Nvidia RTX 4070 Super vždy přibližně 2 hodiny, tudíž větší dataset by byl nežádoucí). Samotný výběr datasetu však díky augmentační metodě není příliš podstatný.

Původně (při odevzdání checkpointu) jsme augmentovali každou zvukovou nahrávku vložením souvislého ticha o náhodné délce z intervalu 3 až 30 sekund do náhodného místa v dané nahrávce, přičemž tento mezivýsledek jsme vždy ještě proložili náhodným šumem. Zřetelné halucinace modelu se nám však dařilo vyvolat pouze s přibližně 2% šancí na výskyt. Tuto původní augmentační metodu lze najít v souboru `data_augmentation.py` pod názvem `augment_audio()`. Po mnoha dalších experimentech jsme však zjistili, že delší pauzy, nejlépe na začátku či konci nahrávek, byly hlavním podnětem proč model halucinoval, a že vložený šum frekvenci halucinací viditelně příliš nezvyšoval. Navíc se nám vůči konečnému vyhodnocení nelíbilo to, že je tato metoda příliš nedeterministická (náhodná délka a náhodné místo vložení ticha). Naše aktualizovaná metoda (`augment_audio_v2()`) tedy vkládá přesně 20 sekund ticha před i po originální nahrávku, přičemž také nabízí možnost přidat sinusovku o frekvenci 25 kHz do výsledné nahrávky (experimentovali jsme totiž také s přidáním frekvencí neslyšitelnými člověkem, tuto funkcionalitu však nakonec nepoužíváme). Při experimentováním s touto metodou se nám podařilo konzisteně vyvolat zřetelně viditelné halucinace s frekvencí výskytu přesahující 50 %.

4 Metody detekce halucinací

Vzhledem k tomu, že samotné halucinace nejsou formalizované, je jejich automatická detekci dosti složitým problémem. Původně jsme se snažili vycházet z metody z článku [2] (viz funkce `detect_hallucinations_article()`), tento přístup se nám ovšem neosvědčil, žřejmě kvůli své komplexitě a nutnosti vždy přizpůsobovat metodu danému datasetu či doméně. Studovali jsme tedy halucinace které se nám podařilo vyvolat, a snažili se v nich identifikovat určité vzory. Na základě empirických zjištění jsme implementovali 2 metody, které se vzájemně dopňují, a přestože tyto metody nejsou samy o sobě až tak přesné, jsou navrženy tak, že snížení počtu nahrávek označené jimi jako halucinační výstupy při samotném vyhodnocení považujeme za velice přesvědčivý důkaz toho, že se nám frekvenci výskytu halucinací

[1] `https://www.openslr.org/12`

podařilo snížit. Všechny zmíněné metody se nachází v souboru `hallucination_detection.py`, přičemž obě následující používané metody jsou implementovány metodou `detect_hallucinations_simple()`.

První používaná metoda vychází z pozorování, že většina halucinací je charakterizována vložením extra slov do samotného zbytku přepisu, který je často správně, tudíž se kontroluje zda-li je délka výstupu modelu delší než referenční přepis. Dále zde kontrolujeme, že Word Error Rate je alespoň 5 % (to již nemá významný vliv, ale pomáhá odfiltrovat určité velice drobné chyby). Tato metoda má samozřejmě tendence označovat některé chyby přepisu za halucinace, avšak jen velmi málo halucinací zůstane touto metodou nedetekováno. Pokud bychom tuto metodu vnímali jako binární klasifikátor (kde třída 1 označuje halucinaci), prohlásili bychom že má skvělý recall.

Druhá používaná metoda vychází z pozorování, že určité konkrétní halucinace se vyskytují ve výstupu velice často (především související s očividným přetrénováním modelu na Youtube obsahu). Vytvořili jsme tedy určitý slovník podřetězců ("end", "thank", "you"[2], "watching", ...), přičemž nachází-li se alespoň jeden z těchto podřetězců ve výstupu modelu, aniž by se nacházel v referenčním přepisu, je výstup považován za běžnou halucinaci. Tato metoda samozřejmě nezachytí určité halucinace, které nejsou až tak časté, avšak přibližně 99 % výstupů označených jako halucinace halucinační opravdu jsou, dá se tedy říct že má tato metoda skvělou přesnost (precision).

5 Metody potlačení halucinací

Vzhledem k tomu, že potlačení halucinací je velmi specifickým tématem a existuje jen málo článků, které se jím zabývají, inspirovali jsme se při řešení tohoto problému různými přístupy používanými v obecných velkých jazykových modelech [3]. Na základě tohoto článku jsme se rozhodli prozkoumat metody zaměřené na preprocessing vstupu a postprocessing výstupu. Hlavním důvodem, proč jsme si vybrali tyto metody, je fakt, že jsme přišli na způsob, jak deterministicky vyvolávat halucinace a tím pádem máme dobrou představu o tom, jak většinu těchto halucinací eliminovat. Obě metody pro potlačení halucinací jsou implementovány jako celek třídou `WhisperLargeV3Wrapped` v souboru `solution.py`.

První metoda (postprocessing, `transcribe_sample_explicit_silence()`) spočívá v tom, že samotný vstup je přepsán také menším ASR modelem, který (příliš) nehalucinuje, a porovnáním obou výstupů se snažíme odstranit části přepisu velkého modelu, které malý model vůbec nepřepsal. Při experimentování s různými verzemi modelu Whisperu jsme si všimli, že nejen že menší modely (tiny, small) halucinují velice málo, ale navíc explicitně vypisují na výstup "silence" nebo "blankaudio", když je na vstupu nějaké delší ticho—této příležitosti jsme tedy využili. Metoda spočívá v současném průchodu obou řetězců, přičemž vždy, když se v přepisu malého modelu narazí na explicitní ticho, identifikují se slova nacházející se těsně před a za tímto slovem (začátky a konce vět jsou také ošetřeny), a část přepisu velkého modelu nacházející se mezi těmito slovy se z výstupu odebere. Jednou z více nevýhod této metody však je, že spoléhá na to že tato ohraničující slova přepíšou oba modely stejně. Funkce je psána konzervativně tak, aby v případě, že tato podmínka splněna není, řetezec od daného bodu už nijak nemodifikovala (ani následující "ticha"). Metoda tedy nehalucinační výstupy nezhorší, ale kvůli podmínkám potřebných pro její správné fungování se nám s ní většinou dařilo odstranit pouhou třetinu až polovinu halucinací.

Druhá metoda (preprocessing, `transcribe_sample_remove_silence()`) se zabývá předzpracováním vstupní zvukové nahrávky a vychází z toho, že prakticky všechny halucinace modelu jsou reakcí na (především delší) úseky vstupu, které neobsahují řeč. Pomocí modelu pro detekci hlasu [6] se ze vstupu vyřežou části, které neobsahují žádný hlas. Námi vybraný model Silero VAD [6] je hodně robustní a zvládá detekovat hlas v téměř libovolném prostředí. Výstupní časové úseky obsahující hlas jsou poté konkatenovány a poslány jako vstup do Whisperu. Ten poté přepíše tuto upravenou nahrávku a vypíše ji na výstup. Model jsme testovali i na zvukových nahrávkách lidí s vadou řeči, která byla dle článku [4] velmi problémová a produkovala halucinace. Pomocí tohoto přístupu jsme byli schopni u testovaných nahrávek odstranit veškeré halucinace.

6 Kvantitativní vyhodnocení výsledků

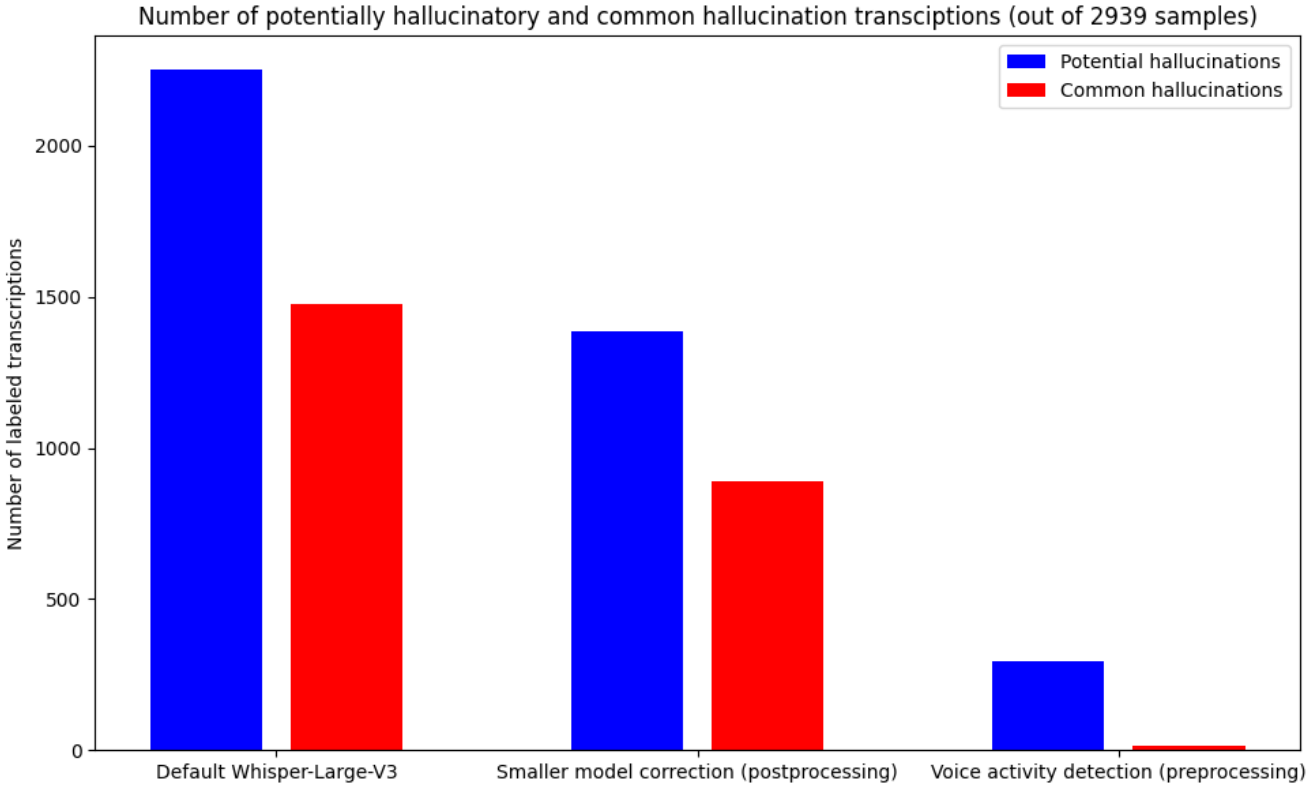
Pro účely vyhodnocení našeho řešení jsme celý dataset (2939 vzorků) přepsali referenčním modelem Whisper-Large-V3, a poté také s pomocí obou našich metod (samozřejmě zvlášť). Jelikož vybraný dataset obsahuje referenční přepisy, které jsou celé velkými písmeny, a obsahují jen určitou diaktritiku, rozhodli jsme se nakonec referenční přepisy, i výstupy modelů převést na malé písmena a odstranili všechnu diakritiku (což snižuje podíl přepisů nepravdivě označených jako potenciální halucinace a zřejmě také pomůže metodě, která zarovnává výstup s výstupem menšího modelu). Výsledky lze vidět v tabulce 1 a také ve formě grafu na obrázku 2. Je vhodné zdůraznit, že až na výjimky by se daly bežné halucinace považovat za podmnožinu potenciálních halucinací.

Pomocí naší první metody (postprocessing, zarovnání s Tiny modelem) se nám podařilo potlačit přibližně třetinu halucinací, a to dle obou našich metrik. Po bližší manuální inspekci samotných přepisů (primárně těch označených za halucinace) jsme dospěli k závěru, že zde není žádný zajímavý vzor v tom, které halucinace tato metoda dokáže

[2]Řetězec "you" je zřejmě tak běžný, že může zapříčinit častá false positiva. Rozhodli jsme se ho však i tak použít, jelikož přidání tohoto řetězce na konec přepisu je jednou z úplně nejběžnějších halucinací použitého modelu.

Tabulka 1: Kvantitativní vyhodnocení výsledků ve formě tabulky.

Metrika / Metoda	Původní Whisper-Large-V3	Zarovnání s Tiny modelem	Detekce řeči
Potenciální halucinace	2251/2939 (76,59 %)	1385/2939 (47,12 %)	294/2939 (10 %)
Běžné halucinace	1476/2939 (50,22 %)	888/2939 (30,21 %)	15/2939 (0,51 %)



Obrázek 2: Kvantitativní vyhodnocení výsledků formou grafu.

odstranit, ale jedná se opravdu o časté nesplnění předpokladů, které jsou na odstranění halucinací touto metodou zapotřebí, jak již bylo vysvětleno v sekci 5.

Druhá metoda se jednoznačně pyšní skvělou úspěšností. Při manuální inspekci všech 294 potenciálních a 15 běžných "halucinací" jsme dokonce zjistili to, že se o halucinace nejedná. Příčinou označení přepisu jako potenciální halucinace v tomto případě byla vždy buď fonetická chyba ("Hermon" → "her mom"), která způsobila že je výstup delší než reference, dále výběr ekvivalentního přepisu, který je delší než alternativní možnost ("Im" → "I am"), případně jiný druh chyby přepisu (nenašli jsme ani jednu jednoznačnou halucinaci). Ještě zajímavější je fakt, že to stejné platí pro všech 15 přepisů označených jako běžná halucinace. Například přepis "but scuse me didnt yo figger on" → "but excuse me didnt you figure on" zapříčinil výskyt podřetezce "you", který se nenachází v referenci. Za předpokladu, že jsme při manuální kontrole výsledků nic nepřehlédli, si dovolíme tvrdit, že se nám touto metodou podařilo potlačit všechny halucinace modelu (a tedy také ověřit, že halucinace jsou zpravidla vyvolané neaktivitou řečníka ve vstupní nahrávce.)

7 Závěr

V této práci jsme se hlouběji zaměřili na problém halucinací modelu pro automatický přepis řeči Whisper (konkrétně verzi Large-V3) od OpenAI. Dokázali jsme deterministicky vyvolat halucinace, díky vkládání úseků bez řeči (v našem případě ticha) před i za nahrávku. Tato augmentace dat dokázala vyvolat (potenciální) halucinace až u 76,59% případů při použití na zvolené datové sadě [5], detekovaných pomocí dvou námi navržených metrik.

Pro potlačení halucinací jsme navrhli dvě různé metody. První je zaměřena na odstranění halucinací z výstupu Whisper-large-V3 použitím post-korekce na základě výstupu jeho nejmenší verze Whisper-Tiny. Tato metoda dokázala odstranit přibližně třetinu halucinací. Druhá metoda je zaměřena na odstranění úseků vstupní nahrávky, které neobsahující řeč pomocí modelu pro detekci řeči [6]. Tato metoda na první pohled dokázala potlačit výskyt halucinací až

strokrát. Nicméně po podrobné inspekci detekovaných halucinací jsme dospěli k závěru, že ve všech takto označených výstupech se dokonce vlastně nejednalo o halucinace. Navíc jsme také tento přístup otestovali na videiích, kde mluví lidé s vadou řeči (afázií), což má podle článku [4] tendenci způsobovat halucinace. U těchto vstupů původní Whisper halucinoval, zatím co Whisper-large-V3 s námi implementovaným preprocessingem ne.

Naším závěrem tedy je, že halucinace tohoto modelu jsou zapříčeněné zpravidla jeho "snahou" o přepis částí vstupní nahrávky, ve které řečník nemluví. Naším doporučením pro uživatele potýkající se s tímto problémem je tedy zajistit to, že vstupní nahrávky neobsahují žádné (delší) pauzy, a ideálně doporučujeme automatizaci tohoto procesu tím, že se všechny vstupy modelu Whisperu nejprve zpracují na základě časových razítek vygenerovaných libovolným spolehlivým modelem na detekci aktivity řečníka (dokonce existují modifikace Whisperu, které mu umožňují časová razítka s lepší přesností generovat).

Reference

[1] Francisco, J. N.: Whisper-v3 Hallucinations on Real World Data. https://deepgram.com/learn/whisper-v3-results, 2024, [Accessed 30-03-2024].

[2] Frieske, R.; Shi, B. E.: Hallucinations in Neural Automatic Speech Recognition: Identifying Errors and Hallucinatory Models. 2024, arXiv:2401.01572.

[3] Ji, Z.; Lee, N.; Frieske, R.; aj.: Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, ročník 55, č. 12, mar 2023, ISSN 0360-0300, doi:10.1145/3571730. Dostupné z: https://doi.org/10.1145/3571730

[4] Koenecke, A.; Choi, A. S. G.; Mei, K.; aj.: Careless Whisper: Speech-to-Text Hallucination Harms. 2024, arXiv:2402.08021.

[5] Panayotov, V.; Chen, G.; Povey, D.; aj.: Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, s. 5206–5210, doi:10.1109/ICASSP.2015.7178964.

[6] Team, S.: Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. https://github.com/snakers4/silero-vad, 2021.