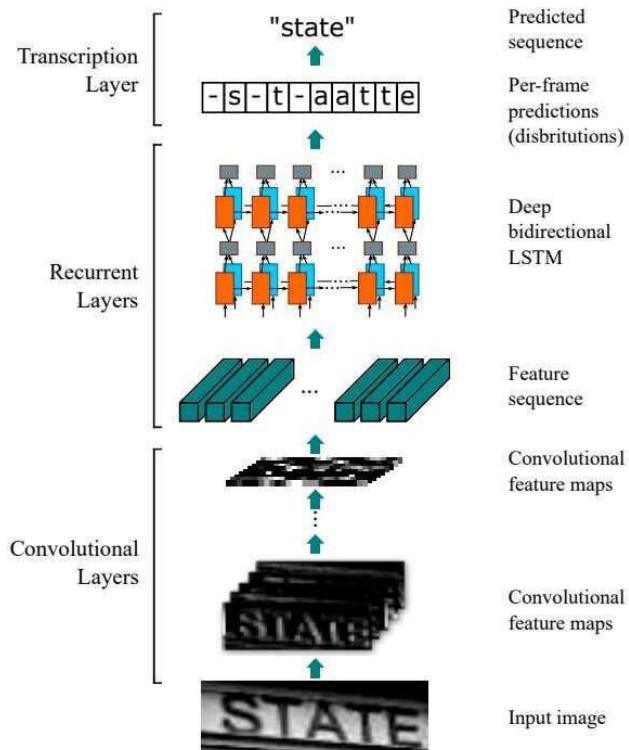


KNN – projektová dokumentace

Radomír Bábek (xbabek02), Petr Volf (xvolfp00), Přemek Janda (xjanda28)
Brno University of Technology

1 Úvod

OCR (Optical character recognition) je metoda, která rozpoznává tištěný či ručně psaný text a vrací jeho digitální přepis. Pro tuto techniku se již celkem standardně využívají neuronové sítě. Starší architektury bývají často postaveny na kombinaci konvolučních a rekurentních neuronových sítí. Konvoluční neuronové sítě obvykle zastávají roli extraktoru rysů. Díky jejich správné utilizaci je možné detektovat text na obrázcích s různým osvětlením a na odlišných pozadích. Rekurentní neuronové sítě, které jsou často rozšířeny do varianty LSTM, se dále starají o samotný překlad rysů na znaky. V kontextu neuronových sítí lze mluvit o standardní architektuře encoder-decoder, kdy je encoder tvořen konvolučními vrstvami a dekodér vrstvami rekurentními.



Obrázek 1: Tesseract OCR rozčleněn na jednotlivé funkci vrstvy, převzato z [5]

Encoder-decoder architektura je napříč novými řešeními stále populární, v poslední době však bylo experimentováno s novými moderními architekturami. Konvoluční neuronové sítě jsou nahrazovány vision transformery postavenými na attention mechanismu. Experimentuje se také s úplným vyloučením encoderu, výsledné neuronové sítě tak bývají postaveny na principu decoder-only. Jako dekodery se využívají generativní předtrénované transformery (GPT), které vstupní obrázky kódují na tokeny a následně pomocí embedding informace iterativně generují znaky až do rozpoznání konce textové sekvence. Generativní transformery se ukázaly jako vysoce efektivní napříč různými typy úloh, aktuálně jsou nejvíce používané v rámci velkých jazykových modelů (LLM).



Obrázek 2: Dataset LAM je předzpracován pro snadné použití při učení. Řádky jsou označené a přepsané do digitální podoby. **Text na obrázku:** *troppo grasso di porco. È necessario ancora*

Dataset

V práci chceme dosáhnout co nejlepších výsledků na některém z ručně psaných datasetů. Mezi zajímavé volby mohou patřít středověké texty z platformy **Zenodo** nebo například German Handwriting¹.

V rámci naší práce jsme zvolili dataset LAM, který obsahuje zápis Italského autora Ludovico Antonio Muratori, pořízené během 60 let jeho života. Vzhledem ke změnám ve formě je vhodná pro HTR (Hand Text Recognition). [1] Dataset je rozdělen na stránky a dále segmentován do řádků, kterých je celkově 25 823. Dataset je předem rozdělený na trénovací, validační a testovací sady.

Cíl práce

V naší práci se zaměřujeme na OCR realizované pomocí transformerů s cílem natrénovat model, který bude kompetentní na datasetu LAM. Výsledný model se zaměřuje na přepis italského ručně psaného textu, který je zpravidla špatně čitelný at' už z důvodu poškození psacího podkladu či kvůli nejednoznačnosti ručně psaného písma.

V první části popisujeme problematiku finetuningu modelu TrOCR[3]. Ve druhé části se zaměřujeme na model DTrOCR a to předtrénování, finetuning a implementační úpravy. V evaluační sekci zhodnocujeme výsledky učení nad datasetem LAM.

2 TrOCR

Model TrOCR je postaven na architektuře typu encoder–decoder, kde vizuální vstup (obraz textu) zpracovává vizuální transformer a výstupní text generuje jazykový transformer. V případě encoderu využívá TrOCR předtrénovaný model **DeiT (Data-efficient Image Transformer)**, který nahrazuje tradiční CNN backbone. Model pracuje na základě rozdělení obrazu do patchů, které zpracovává jako sekvenci.

Na Hugging Face jsou pro ručně psané texty k dispozici tři modely: `small`, `base` a `large`. Pro jednoduchost a rychlosť učení jsme si vybrali variantu `small`.

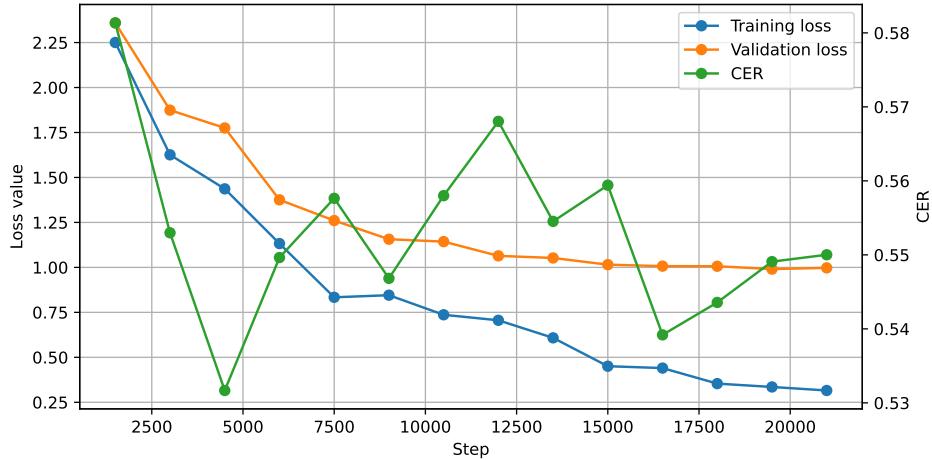
Při vyhodnocení baseline mělo naše použití modelu velmi špatně výsledky. Metoda téměř nefungovala, hodnotící metrika CER značila že model správně určil jen jeden znak z deseti (CER okolo 0.9). Chyba byla v konfiguraci vstupních dat při fine-tuningu, kde se hodnoty v obrázku normalizovaly dvakrát.

Po opravě této chyby a novém finetuningu už model produkoval lepší výsledky, poznačené v grafu 3. Chyba byla však přesto příliš vysoká pro jakékoli využití. Po manuální kontrole výsledků jsme zjistili že model i po fine-tuningu rozeznával pouze polovinu znaků, druhá polovina ve výstupu chyběla. Ukázka tohoto chování je v tabulce 1. Domněnka byla taková, že za to může interní škálování vstupního obrázku na rozlišení 384 × 384. To mohlo kurzívu na obrázcích s velkým poměrem stran v datasetu LAM téměř smazat. I z tohoto důvodu jsme zvolili jiný přístup.

Ground truth	TrOCR finetuned
troppo grasso di porco. È necessario ancora	trop gra dicens anra
lo star allegro il più che si può, e non faticar	lo all il che pu, noncar
alle volte più di quello che portano le forze	all volt cheano forze
naturali: il troppo poco ancor egli genera	natural. trop fu e general

Tabulka 1: Tabulka s ukázkou ořezání slov při inferenci TrOCR.

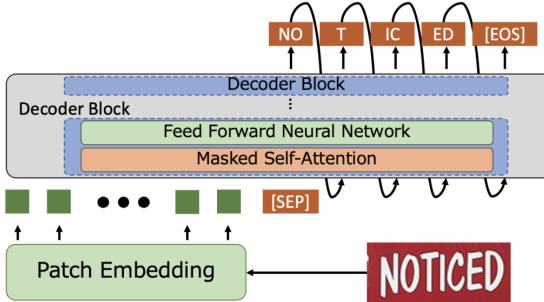
¹https://huggingface.co/datasets/fhswf/german_handwriting dataset.



Obrázek 3: Průběh opraveného fine-tuningu TrOCR

3 DTrOCR

Architektura DTrOCR je produktem práce Masata Fujitakeho [2]. Jedná se o decoder-only architekturu využívající GPT-² poskytnutého od OpenAI. Parametry modelu, který článek představující model evaluuje, nejsou veřejně dostupné. Model bylo tedy v rámci práce nutné kompletně natrénovat. Oproti TrOCR tréninku v počátcích práce však přistupujeme k problému metodičtěji a pokoušíme se o rozsáhlé předtrénování na latinských textech před samotným finetuningem.



Obrázek 4: Architektura DTrOCR, (převzato z [2]). Vstupní obrázky jsou rozděleny na patches o určitém počtu pixelů. Pro každý patch je vytvořen patch embedding. Za sérii patches je vložen token [BOS] – beginning of sequence. Následně je sekvence tokenů vložena na vstup generativního transformera, který vygeneruje další token. Původní vstup s novým přidaným tokenem je opět vložen na vstup transformera, procedura se opakuje do vygenerování tokenu [EOS] – end of sequence.

Pro využití DTrOCR v rámci našeho projektu jsme se rozhodli na základě neúspěchu s TrOCR a pozitivních výstupů prezentovaných v rámci vědeckých publikacích, které nasvědčují, že decoder-only byť poměrně nová architektura prokazuje lepší výsledky než většina dosavadních architektur. Architekturu DTrOCR jsme upravili a při tréninku využíváme CTC loss nad zadánou abecedou. Pro trénování experimentujeme s SAM (Sharpness aware minimization) optimizátorem, který se snaží o lepší generalizační vlastnosti výsledného modelu a penalizuje ostrá minima při trénování. Nápad využít CTC loss a SAM optimizátor přebíráme z [4], kde autoři popisují dobré výsledky nad námi cíleným datasetem LAM.

Předtrénování a finetuning

Architektury využívající transformery jsou schopné dosáhnout velice dobrých predikčních výsledků, je však nutné poskytnout velkou datovou sadu v rámci fáze předtrénování. Pouze poté se model přizpůsobí chtěné činnosti, kterou může být například porozumění jazyku či obrázkům. Pro předtrénování jsme vygenerovali velký syntetický dataset

²<https://huggingface.co/openai-community/gpt2>



Obrázek 5: Příklady řádků syntetického datasetu. Horní obrázek představuje příklad z datasetu 1, který více či méně zhoršuje kvalitu původního textu z latinského korpusu a snaží se přiblížit vizuální podobě LAM datasetu v jednom z využitých fontů. Spodní 2 příklady představují příklady z datasetu 2, který má za cíl zvýšit generalizaci modelu nad ručně psaným textem a alespoň přibližně pomoci modelu rozpoznat tištěný text. Důležitá složka datasetu je také barevná náhodnost, kdy je provedena snaha naučit model vyhledávat text ne dle barvy, ale pomocí kontrastu pozadí a textu.

obsahující především latinské texty z korpusu **oscar**³. Jako generativní program byl využit nástroj Synthtiger⁴. Primární cíl práce byl dosáhnout dobrých výsledků nad LAM datasetem. Syntetický dataset je tedy generován pomocí fontů simulující ruční psaní volně dostupných na Google Fonts. Generovány byly celkem 2 datasety, oba použité v rámci předtrénování. První datová sada byla vytvořena s účelem co největší podobnosti LAM datasetu, podklad pro generování tvoří obrázky starých papírů, texty jsou pouze latinsky. Druhá datová sada obsahuje i klasické fonty jako je Ubuntu a Roboto, může obsahovat i anglické a zvláštní texty (například url adresy). Písma a pozadí jsou různě barevné, účelem druhé datové sady je zvýšit generalizační schopnosti modelu. Model byl předtrénován na 1.2 milionech vstupů v počtu jedné epochy.

Předtrénovaný model byl následně dotrénován nad LAM datasetem. Bylo provedeno 35 epoch nad trénovací datovou sadou, trénování bylo zastaveno, aby se předešlo neefektivnímu overfittingu. Je nutné podotknout, že velikost datové sady pro předtrénování stále nebyla dostatečná. Z důvodu nedostatků zkušeností trénování probíhalo nejprve na osobních počítačích členů týmu, později však byl využit nemalý počet výpočetních hodin na českém Metacentru. Lepších výsledků po stejném počtu epoch by šlo získat rozšířením datové sady pro předtrénování.

4 Vyhodnocení

Vyhodnocovací metriky

Pro OCR úlohy se běžně používají dvě základní metriky: chybovost na úrovni písmen (*Character Error Rate*, CER) a chybovost na úrovni slov (*Word Error Rate*, WER). Výstupní (predikovaný) text modelu je porovnáván s manuálně označeným (referenčním/ground truth) textem.

CER měří rozdíl mezi jednotlivými znaky a je vhodnější pro modely zaměřené na rozpoznávání bez znalosti jazykového kontextu, typicky tam, kde modely klasifikují každé písmeno zvlášť. Oproti tomu WER porovnává celá slova a je vhodnější tam, kde model využívá jazykové modelování (např. jedna chyba v písmenu vede k celkové chybě ve slově, což WER oproti CER penalizuje více).

Měříme celkem čtyři metriky:

$$\text{CER} = \frac{S + D + I}{N}$$

$$\text{WER} = \frac{S + D + I}{W}$$

$$\text{Accuracy} = \frac{\text{Počet správně rozpoznaných slov}}{\text{Celkový počet slov}}$$

³Dostupné z <https://huggingface.co/datasets/oscar-corpus/oscar>

⁴Dostupné z <https://github.com/clovaai/synthtiger>

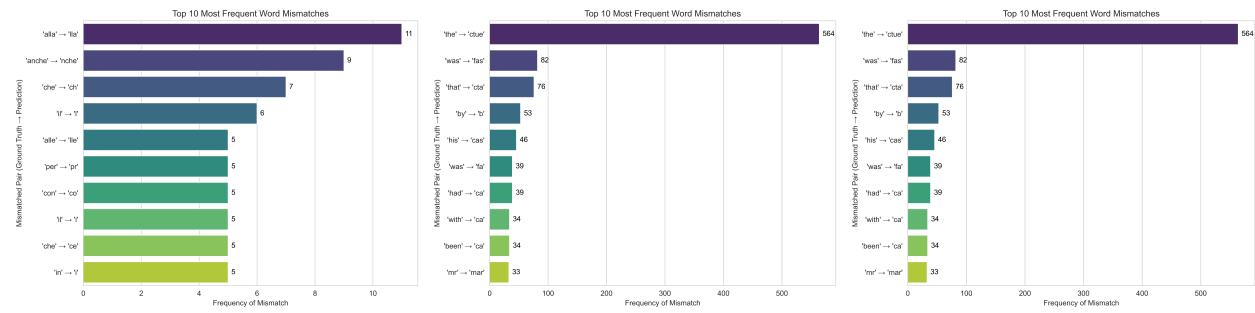
$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

kde:

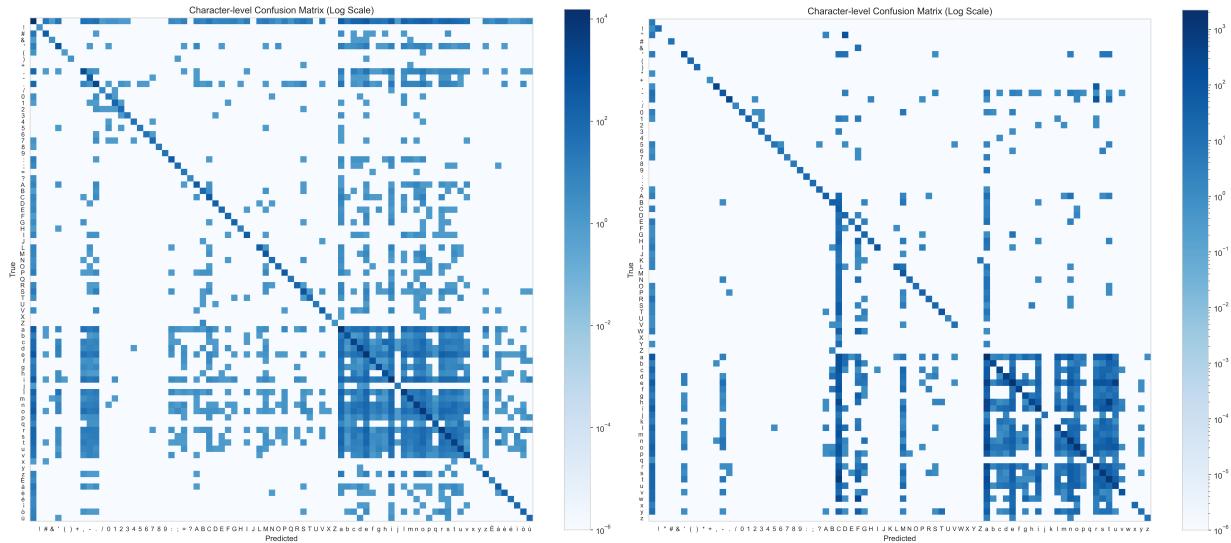
- S je počet substitucí (špatně rozpoznaných znaků nebo slov),
- D je počet delecí (chybějících znaků nebo slov),
- I je počet insercí (nadbytečných znaků nebo slov),
- N je počet znaků v referenčním textu (pro CER),
- W je počet slov v referenčním textu (pro WER).

5 Experiments

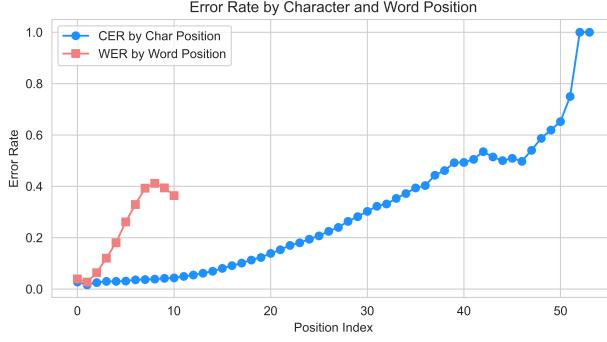
Následující grafy, primárně, ukazují vyhodnocení finetuned modelu nad LAM datasetem.



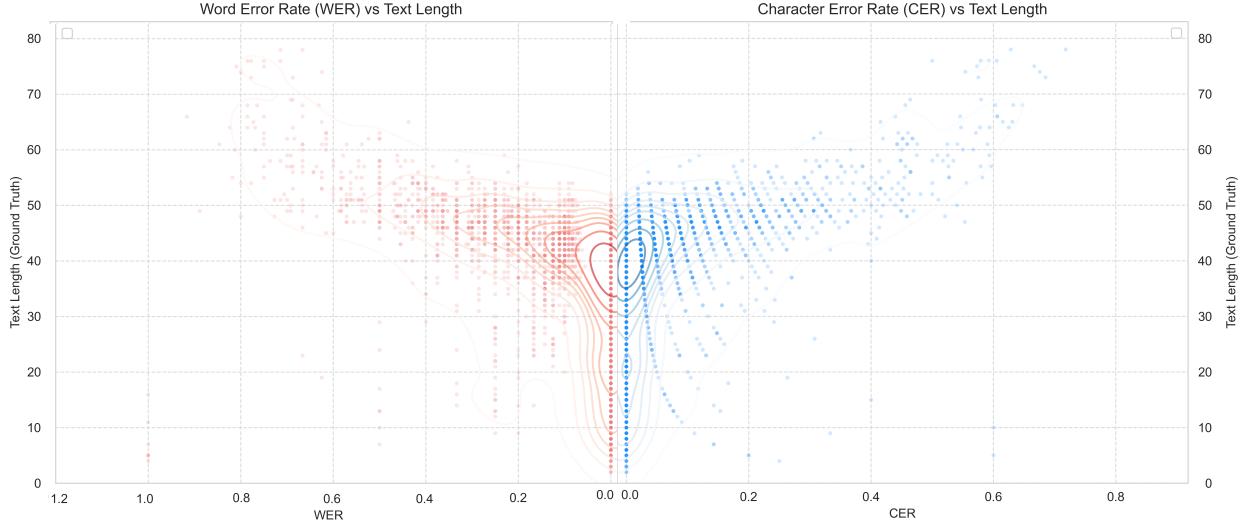
Obrázek 6: Následující 3 grafy ukazují nejčastější chyb při rozpoznávání slov u dotrénovaného modelu nad zleva LAM, IAM a Syntetic datasetem. Lze obecně pozorovat trend, kdy model vynechává počáteční samohlásky (např. 'a', 'i') nebo koncové hlásky u krátkých, běžně užívaných italských slov, pak je také vysoká četnost chyb u neznámých anglických slov např. všudypřítomného the v IAM datasetu.



Obrázek 7: Oba modely (fine-LAM vlevo, fine-IAM cpravo) ukazují obecně dobrou schopnost rozpoznávat znaky, jak naznačuje silná diagonála. Běžné problémy jako záměna velikosti písmen a záměna vizuálně podobných znaků/interpunkce jsou přítomny u obou. Hlavní rozdíl spočívá ve zpracování znaků specifických pro daný jazyk/dataset. U LAM datasetu je největším problém systematické vynechávání nebo chybné rozpoznávání diakritiky (typické pro italštinu), což u IAM datasetu není takovým problémem. Grafy používají logaritmickou škálu, což znamená, že i méně časté záměny jsou lépe viditelné.



Obrázek 8: Model má tendenci dělat více chyb (jak na úrovni znaků, tak slov) čím dále postupuje v generování nebo rozpoznávání sekvence. Začátky sekvencí jsou rozpoznávány s vyšší přesností než koncovyšší přesností než jejich konce. Chyby se akumulují a model má potíže s udržením kontextu na delší vzdálenosti (error propagation).



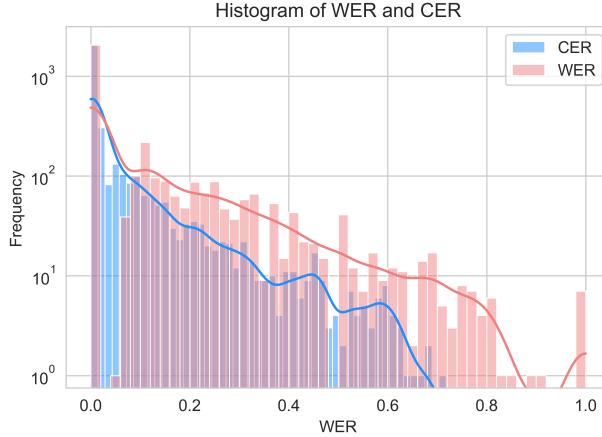
Obrázek 9: Tento dvojitý graf znázorňuje vztah mezi délkou textu (Text Length - Ground Truth) na svislé ose a chybovostí slov (WER) v levém panelu, respektive chybovostí znaků (CER) v pravém panelu na vodorovné ose. Každý bod představuje jeden textový vzorek, přičemž konturové čáry znázorňují hustotu rozložení bodů. Celkově graf naznačuje, že existuje optimální rozsah délky textu, pro který model dosahuje nejnižší chybovosti. Pro velmi krátké a velmi dlouhé texty má model tendenci vykazovat vyšší chybovost a větší variabilitu ve výkonu.

5.1 Experiment

Jak jsme již výše zmiňovali, hlavním cílem naší práce bylo předtrénovat model na syntetickém datasetu (v tabulce je model symbolizován prefixem *pre-*) a poté ho specializovat na rozpoznávání textu na datasetu LAM (prefix *fine-*). Vzhledem k výsledkům summarizovaným v tabulce 2 se nám to povedlo (především porovnání prvního a čtvrtého sloupce). Nicméně, dosažené výsledky nejsou uspokojující, vzhledem k tomu, že se výsledný model zhoršil na dalších datasetech. Nedá se tedy tvrdit, že by model bych schopen uspokojivé generalizace napříč datasety. Předtrénování prokázalo nejvyšší výkon a nízkou chybovost na obou hlavních metrikách CER i WER. Následné doladění na reálné datové sadě LAM však vedlo ke zhoršení těchto metrik, přičemž celková CER vzrostla na 0.0781, celková WER na 0.1381 a přesnost klesla na 0.5819.

5.2 Experiment

Varianta s největším množstvím unikátních dat (Model 1a) selhala kvůli nestabilitě. Je třeba poznamenat, že menší batch size (8) může vést k dynamitčtějším gradientům. To může být na jednu stranu prospěšné pro exploraci prostoru parametrů, což ale na druhou stranu nejspíše způsobilo nestabilitu učení. Při volbě vyššího batch size (32) se gradienty



Obrázek 10: Graf ukazuje, že pro obě metriky (CER, WER) jsou zastoupeny nejvyšší frekvence u nízkých hodnot chybovosti, vzhledem k logaritmickému měřítku osy y.

	fine-LAM	fine-IAM	fine-Synth	pre-LAM	pre-IAM	pre-Synth
↓ Total CER	0.0781	0.5251	0.1674	0.2865	0.1775	0.0309
↓ Total WER	0.1381	0.6509	0.3193	0.4224	0.2353	0.0678
↑ Accuracy	0.5819	0.3801	0.4605	0.1655	0.7631	0.8369
↑ F1	0.3926	0.1337	0.2830	0.0880	0.4624	0.7062
↓ Average CER	0.0582	0.4919	0.1796	0.2479	0.1175	0.0222
↓ Median CER	0.0000	0.5000	0.0400	0.2500	0.0000	0.0000
↓ Std Dev CER	0.1120	0.5542	0.3142	0.1799	0.2562	0.0700
↓ Average WER	0.1098	0.6595	0.3514	0.3638	0.2353	0.0595
↓ Median WER	0.0000	1.0000	0.1667	0.3750	0.0000	0.0000
↓ Std Dev WER	0.1736	0.5740	0.4972	0.2401	0.4303	0.1737

Tabulka 2: Šipky ↓, ↑ označují vyšší respektive nižší hodnotu jako lepší.

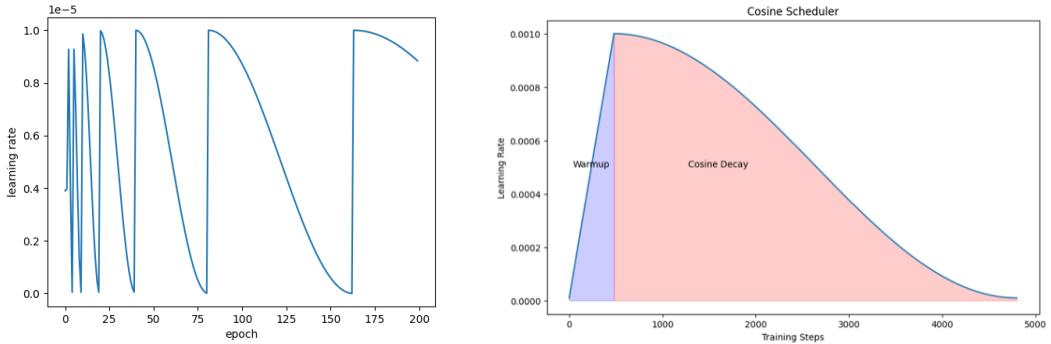
stávají výrazně hladšími, což vede k rychlejší konvergenci při učení. Experimenty také prokázaly, že vyšší variabilita datové sady vede k lepšímu výsledku než zvýšení počtu epoch, zatímco celkové množství vstupů zůstává zachováno.

	Epochs	Batch size	Samples	Dataset size
Model 1a	1	8	50 k	30 %
Model 1b	2	8	50 k	16 %
Model 1c	5	8	50 k	8.3 %
Model 1d	10	8	50 k	4.2 %
Model 2a	1	32	200 k	16 %
Model 2b	2	32	100 k	8.3 %
Model 2c	20	32	10 k	0.8 %

Tabulka 3: Porovnání učících křivek dle počtu dat a trénovacích epoch.

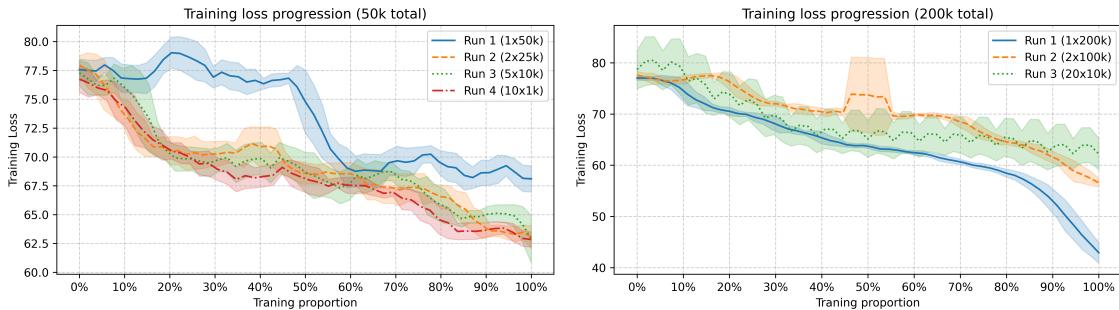
5.3 Experiment

Cílem tohoto experimentu bylo prověřit, jakým způsobem ovlivňují některé faktory učení. Prvně jsem se zaměřili na to, jakým způsobem jsou schopny modely generalizovat podle toho, na jak rozmanitých datech se učily. Tabulka 4



Obrázek 11: Průběh cos learning rate, Vlevo reset po dosažení minimální hodnoty, vpravo jedna fáze wam-pu a cosice decay. Grafy převzaty ze zdrojů (zde a zde).

zaznamenává 2 různé konfigurace modelů, kde model 1 je trénován celkem na 50 000 rozdelených do batch o velikosti 8 obrázcích. Model 2 je rozdělen po 32 a celkem byl trénován na 200 000 příkladech. Na porovnání se používá cosine-warmup (viz) na při každé epoše, což bylo druhotním předmětem zkoumání. Tato charakteristika je nejlépe zřetelná u křivky s 20 epochami, kde lze pozorovat nábeh ztráty při učení při každé epoše. Dalším pozorovaným faktem bylo, že zvolením větší velikosti batch byl průběh učení hladší a stabilnější. Nicméně s velikostí batch byla limitací na většině použitých strojů bylo možno použít nejvýše velikosti 32-64.



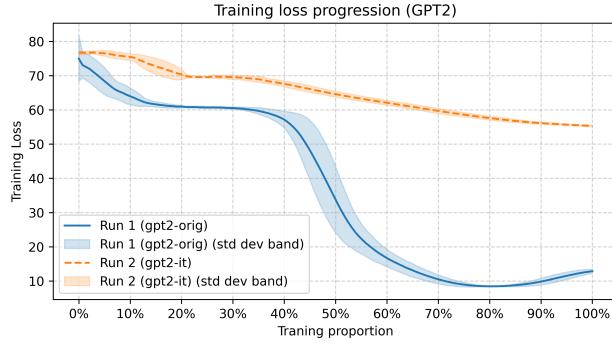
Obrázek 12: Vývoj chybové chyby na modelech trénovaných na 50 000 vstupech vlevo a 100 000 vstupech vpravo.

5.4 Experiment

Experiment záměny předtrénovaného backbone z originálního GPT-2 z platformy *hugging face*. *openai-community/gpt2* (počet tokenů: 50257) na *LorenzoDeMattei/GePpeTto* (počet tokenů: 30000). Naší hypotézou bylo, že použití přetrénovaného modelu s italským kontextem povede ke zlepšení. Nicméně z výsledků experimentů se ukázal opak. Příčinu chování se nám nepodařilo odhalit. Pro trénování bylo použito 50 000 vstupů trénováno na 100 epochách, 32 je originální model GPT-2 (*gpt2-orig*) výrazně vhodnější než italsky předtrénovaný model (*gpt2-it*)

	Epochs	Batch size	Samples	Dataset size	CER	WER	Final Loss
GPT2 original	100	32	160 k	12.7 %	0.1877	0.4339	21.366
GPT2 it	100	32	160 k	12.7 %	0.5823	0.5891	37.130

Tabulka 4: Porovnání modelů dekodérů GPT-2 z platformy *hugging face*.



Obrázek 13: Porovnání ztrátové chyby modelů dekodérů GPT-2 z platformy hugging face.

6 Závěr

Výstupy z trénování, grafy a vygenerované data lze najít ve složce output, kde jsou jednotlivé výsledky experimentů rozděleny do složek podle toho, zda šlo o testování (adresář TEST) nebo trénování (TRAIN). V souboru README.md jsou také příkazy, pro reprodukci výsledků.

Reference

- [1] Silvia Cascianelli, Vittorio Pippi, Maarand Martin, Marcella Cornia, Lorenzo Baraldi, Kermorvant Christopher, and Rita Cucchiara. The lam dataset: A novel benchmark for line-level handwritten text recognition. In *International Conference on Pattern Recognition*, 2022.
- [2] Masato Fujitake. Dtrocr: Decoder-only transformer for optical character recognition, 2023.
- [3] Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models, 2021.
- [4] Yuting Li, Dexiong Chen, Tinglong Tang, and Xi Shen. Htr-vt: Handwritten text recognition with vision transformer. *Pattern Recognition*, 158:110967, 2025.
- [5] Ritika Rai, Srushti Shitole, Pratiksha Sutar, Swapnali Kaldhone, and Jayashree Jadhav. Automatic license plate recognition using yolov4 and tesseract ocr. *International Journal of Innovative Research in Computer and Communication Engineering*, 10:1656, 01 2008.