AUTOMATICKÉ GENEROVÁNÍ INTERPUNKCE V SYSTÉMECH ROZPOZNÁVÁNÍ ŘEČI

Martin Poláček, Petr Červa, Jindřich Žďánský, Lenka Weingartová
<martin.polacek@tul.cz>

Tato práce prezentuje modul pro automatické generování interpunkce (APR) v systémech pro rozpoznávání řeči. Modul využívá jazykový model ELECTRA-Small doplněný o klasifikační hlavu a dokáže generovat tečky, čárky a otazníky s nízkým inferenčním časem a zpožděním pouhá 3 slova. V práci jsou nejdříve porovnány různé architektury jazykových modelů a klasifikačních hlav na datech, která jsou složena z ručních přepisů televizních zpráv. Následně je zkoumán vliv počtu slov jako budoucího kontextu a závěrem je modul porovnán s již existujícím modulem, který využívá jak textové, tak i prosodické příznaky.

Klíčová slova: automatické generování interpunkce, automatické rozpoznávání řeči, ELECTRA model, transformery

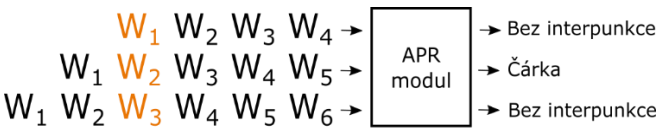
ÚVOD

V současné době patří systémy automatického rozpoznávání řeči (ASR) k neodmyslitelným nástrojům každodenního života. Jejich uplatnění nacházíme nejen v přepisech audio nahrávek, ale i u živých přenosů, jako jsou rozhlasové či televizní vysílání.

ASR systémy obvykle generují textové přepisy bez interpunkčních znamének, což je dáno jejich absencí v trénovacích datech. Tento nedostatek výrazně komplikuje čitelnost a srozumitelnost výstupu. Kromě toho absence interpunkce může negativně ovlivnit úlohy, které následují (například automatický překlad nebo analýzu textu, kde jsou jasně definované hranice vět klíčové).

Pro řešení tohoto problému je potřeba použít doplňkový modul pro automatické generování interpunkce (APR), který do výstupu z ASR systému doplní interpunkční znaménka na základě kontextu (minulého a budoucího), který je mu poskytnut.

Tato práce je zaměřena na generování interpunkce do streamovaného datového toku. S tím však přicházejí výzvy spojené s výpočetní náročností a zpožděním. Pro efektivní zpracování streamovaných dat musí být zpoždění omezeno na několik slov, což významně limituje kontext, který může APR modul využít pro správné doplnění interpunkčních znamének.



Obrázek 1: Generování interpunkce APR modulem v online režimu

METODIKA

Všechna data pro trénování a validaci byla předem zpracována do jednotlivých vět a ponechány byly jen tečky, čárky a otazníky. Poté byl vytvořen unigramový Sentencepiece model pro tokenizaci vět.

Jádrem APR modulu je předtrénovaný transformer ELECTRA-Small [1], který generuje pro každý vstupní token příznakový vektor o velikosti 256. Transformer je následně doplněn o klasifikační hlavu, která se skládá ze dvou feed-forward vrstev (512 a 4 neurony) s aktivační funkcí SELU.

Transformer doplněný o klasifikační hlavu byl dotrénován s použitím malého kroku učení (LR) pro předtrénovaný model a velkým LR pro nově přidanou klasifikační hlavu.

V každém dopředném průchodu modul generuje pro každý vstupní token pravděpodobnosti pro všechny interpunkční třídy (žádná interpunkce, otazník, čárka, a tečka). V reálném čase, tedy v online zpracování, je nutné provést dopředný průchod pro každé slovo vstupního textu.

Pro vyhodnocení výsledků byly využity metriky precision, recall a F1 skóre. Nejdříve byly vyhodnocovány výsledky pro jednotlivé třídy zvlášť. Následně v „one-class“ scénaři byly třídy pro tečku, čárku a otazník sloučeny do jedné. Jednalo se tak pouze o binární klasifikaci, zda za tokenem následuje interpunkční znaménko nebo nikoliv.

VÝSLEDKY A DISKUZE

Při porovnání architektur byl použit transformer BERT-Base [2], GPT-3.5 (v editačním režimu) [3], ELECTRA-Small [1] a ELECTRA-Base [1]. Oba ELECTRA modely byly navíc trénované od počátku.

Výsledky porovnání jednotlivých architektur jsou uvedeny v tabulce 1. ELECTRA-Small dosahuje nejlepších výsledků v porovnání se všemi ostatními transformery. Zároveň dosahuje i nejnižšího inferenčního času a proto je také použitelný v online režimu.

architektura	F1 [%]	F1 (one class) [%]	inf. čas [ms]
BERT-Base	75.2	88.9	61
ELECTRA-Small	**76.0**	**90.7**	**11**
ELECTRA-Base	75.4	90.7	60
GPT-3 (v editačním módu)	65.1	73.7	-

Tabulka 1: Porovnání výsledků různých architektur v blokovém režimu

Druhý experiment měl za cíl zjištění pravého (budoucího) kontextu. Byl nastaven limit pro maximální levý kontext na 100 slov a byl vyšetřován počet budoucích slov, kdy modul bude dosahovat podobných výsledků jako v blokovém režimu (interpunkce vygenerována pro celý blok textu při jednom dopředném průchodu) a zároveň co nejmenšího zpoždění. Z výsledků v tabulce 2 vyplývá, že nejlepší pravý kontext jsou tři slova.

max. levý kont.	max. pravý kont.	P [%]	R [%]	F1 [%]
100	1	73.3	67.4	70.2
100	2	75.0	72.5	73.7
100	3	75.3	74.2	74.7
100	4	75.2	75.1	75.1
100	5	75.5	75.5	75.5
100	10	**76.0**	76.0	**76.0**
100	100	75.6	73.7	74.6
blokové zpracování bez překryvu		75.3	**76.8**	**76.0**

Tabulka 2: Výsledky [%] APR modulu ve streamovacím režimu

V posledním experimentu byl navržený APR modul porovnán s již existujícím modulem [4]. Tento modul je založen na rekurentních neuronových sítích (RNN) a využívá jak textové příznaky, tak i prosodické příznaky a informace o tichu získané z audio nahrávek.

Pro experiment byla vytvořena speciální datová sada složená z automaticky přepsaných debat a čtených pořadů, kde nebyly opraveny chyby a pouze doplněna interpunkce.

Typ promluvy	Navržený APR F1 [%]	Navržený APR F1 (one class) [%]	RNN APR F1 [%]	RNN APR F1 (one class) [%]
Čtená	**71.2**	**84.3**	62.1	73.4
Spontánní	69.4	**89.0**	**71.6**	73.3

Tabulka 3: Porovnání navrženého APR modulu v online režimu s existujícím RNN APR modulem

Z výsledků v tabulce 3 vyplývá, že navržený APR modul dosahoval téměř srovnatelných nebo lepších výsledků při použití pouze textových příznaků. Výrazně vyšších výsledků dosáhl hlavně při one-class scénáři u spontánní řeči. Z výsledků vyplývá, že dochází k časté záměně teček a čárek. Tato chyba však ve většině případů nemá vliv na výsledný kontext daného přepisu.

ZÁVĚR

Navržený APR modul využívá transformer ELECTRA-Small doplněný o klasifikační hlavu. Funguje v online režimu s latencí pouhých 3 slov a na vstupu přijímá pouze textové příznaky. Zároveň má modul velmi nízký inferenční čas, kdy na procesoru i7 9700K jeden dopředný průchod trvá 11 ms.

Přesnost modulu je téměř srovnatelná s přesností v blokovém režimu. Nejčastější chyby vznikají záměnou teček a čárek, což ve většině případů neovlivní výsledný kontext.

Pro trénování potřebuje navržený modul pouze textová data, což výrazně ulehčuje proces přípravy dat v porovnání s moduly, které využívají i prosodické příznaky získané z audio nahrávek.

PODĚKOVÁNÍ

Tato práce byla podpořena z projektu Studentské grantové soutěže (SGS) na Technické univerzitě v Liberci v roce 2023.

REFERENCE

[1] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: pre-training text encoders as discriminators rather than generators” in ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pretraining of deep bidirectional transformers for language understanding,” in NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019. ACL, 2019, pp. 4171–4186.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in NIPS 2020, Virtual event, December 6-12, 2020

[4] P. Hlubik, M. Spanel, M. Bohac, and L. Weingartova, “Inserting punctuation to ASR output in a real-time production environment,” in TSD 2020, Brno, Czechia, September 8-11, 2020, ser. Lecture Notes in Computer Science, vol. 12284. Springer, 2020, pp. 418–425.