Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Semestrální práce z KIV/ANLP

Morfologické značkování s využitím strukturovaných značek

Adam Mištera

A19N0038P

amistera@students.zcu.cz

Plzeň, 20. 1. 2021

Obsah

1 Úvod

Cílem semestrální práce z předmětu KIV/ANLP bylo dosáhnout hlubšího porozumění v oblasti rozpoznávání přirozeného jazyka prostřednictvím jedné z jeho základních úloh, konkrétně morfologického značkování. Při tvorbě práce byl současně využit předtrénovaný model BERT, který je v současné době považován za „state-of-the-art“ v oblasti NLP a bude blíže představen v kapitole 4.

2 Zadání

Úkolem bude vytvořit systém pro morfologické značkování s využitím strukturovaných značek. Strukturovaná značka popisuje morfologii slov 15-ti různými hodnotami (některé pozice nejsou využívány).

V rámci řešení vytvoříte tři architektury:

- Rozpoznávání atomických značek (celá značka bude brána jako nedělitelný prvek).
- Nezávislé rozpoznávání dekomponovaných značek (značka bude rozdělena na 15 částí a každá část bude klasifikována zvlášť).[1]
- Rozpoznávání dekomponovaných značek s jejich následnou syntézou (architektura bude typu end2end).

V následují kapitole bude představena kompletní hardwarová i softwarová konfigurace, která byla nezbytná pro natrénování i otestování daných modelů.

3 Parametry testovacího prostředí

Vzhledem k poměrně vysoké hardwarové náročnosti modelu BERT pro natrénování je v této kapitole uvedena kompletní konfigurace prostředí použitého pro trénování všech tří modelů zmíněných v zadání. Trénování a všechny dále uvedené testy byly provedeny v následujícím prostředí metacentra:

- **OS** - Debian GNU/Linux 10
- **CPU** - 1 × Intel® Xeon Gold 5218 2.30GHz (1 jádro)
- **RAM** - 24 GB
- **SSD** - 20 GB
- **GPU** - 1 × NVIDIA Tesla T4 16GB[2]

[1] Struktura značky je dostupná na adrese `http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/html/ch02s02s01.html`.

[2] Zejména pro druhý model by bylo lepší využít novější GPU NVIDIA A100 40GB, která však správně nefungovala s použitým softwarem.

Semestrální práce byla kompletně vypracována s použitím programovacího jazyka **Python 3.7** a balíčku **Keras** sloužícího jako rozhraní pro knihovnu **TensorFlow**. Během tvorby semestrální práce však vzniklo několik problémů kvůli nekompatibilitě verzí některých balíčků. Pro jednodušší správu prostředí a jednotlivých balíčků byla proto posléze využita aplikace *Conda*.[3].

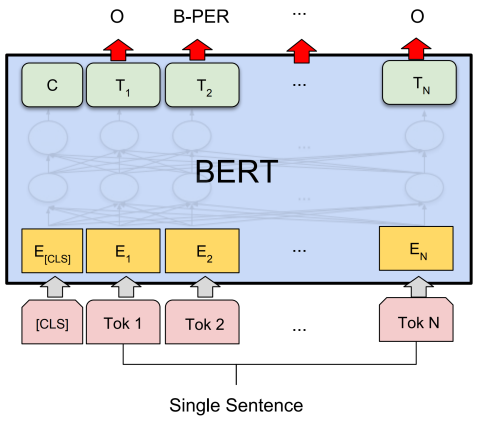
Dále jsou uvedeny tři nejdůležitější balíčky pro trénink modelu BERT společně s označením verzí, které jsou kompatibilní:

- **Keras** - 2.4.3,
- **Tensorflow** - 2.2.0,
- **Transformers** - 4.0.1.

Ostatní potřebné balíčky, kterých je celkem přes 100, si již dokáže aplikace *Conda* doinstalovat sama. Pro natrénování modelů bez rizika chyb je však nutné využít výše uvedenou kombinaci. Využitý repositář *conda-forge* pro získání jednotlivých balíčků totiž v současné době obsahuje knihovnu *Tensorflow* v nejvyšší verzi pouze 2.2.0, se kterou však nejsou kompatibilní novější verze klíčového balíčku *Transformers*.

4 Analýza

Bidirectional Encoder Representations from Transformers nebo také zkráceně **BERT** označuje rozsáhlý předtrénovaný model založený na technice strojového učení známé jako *Transformer*, který v současné době dosahuje nejlepších výsledků v oblasti zpracování přirozeného jazyka (NLP). Jeho strukturu můžeme vidět na obrázku 1. Základní model **BERT-base** obsahuje enkodér,



Obrázek 1: Model BERT pro úkol NER (Devlin et al. 2018)

[3]V prostředí metacentra se jedná o balíček `conda-modules-py37`.

který se skládá z 12 bloků *Transformerů*, 12 takzvaných *self-attention heads* a skryté vrstvy obsahující 768 neuronů. BERT na vstupu přijímá sekvenci o maximální délce 512 tokenů.[1] Jeho výstupem je následně reprezentace dané sekvence. Jedná se tedy o velmi komplexní model.

Značnou výhodou tohoto modelu je fakt, že je již předtrénován. S jeho pomocí je tedy možné řešit více různých úkolů s NLP. Použitý model je nutné na zvolený úkol pouze doladit. Výhody tohoto modelu jsou však částečně vykoupeny jeho značnou velikostí, která výrazně znesnadňuje trénování modelu na běžně dostupném hardwaru.

5 Implementace

V této části si detailněji představíme implementaci všech tří modelů. BERT model pro dotrénování byl u prvních dvou modelů zvolen *bert-base-multilingual-cased*.[4] Tento model byl doporučen pro případ, kdy vstupní data obsahují texty v jiném než anglickém jazyce a lze jej velmi jednoduše stáhnout a načíst.

5.1 Rozpoznávání atomické značky

Pro implementaci prvního modelu, který rozpoznává celou značku jako jeden nedělitelný prvek, byl použit objekt `TFBertForTokenClassification`, který je pro tento případ přizpůsoben. Počet rozpoznávaných značek lze díky použití tohoto objektu nastavit pouze předáním parametru do konstruktoru. Pro správnou funkčnost modelu bylo nutné dále nastavit aktivační funkci pro výstupní vrstvu. V tomto případě byla zvolena aktivační funkce *softmax*. Model obsahuje dva vstupy, konkrétně se jedná o tokenizovaná slova jednotlivých vět a takzvanou *attention mask*, která udává, na které části vstupu má model při tréninku reagovat.[5].

Jednotlivé hyperparametry trénovaného modelu jsou již poměrně standardní, vyjma parametru pro rychlost učení, kterou je nutné zvolit nízkou, jelikož je model BERT již předtrénován. Hodnoty pro rychlost učení se ve většině případů pohybují v rozmezí od 1e−5 do 5e−5. V rámci semestrální práce byla jako optimální hodnota zvolena 3e−5 pro první model a 2e−5 pro druhý model, který byl na tento parametr citlivější. Pro natrénování tohoto modelu byly využity celkem tři epochy.

5.2 Rozpoznávání dekomponované značky

Druhý model byl oproti prvnímu modelu značně komplikovanější, jelikož rozpoznával značku po jednotlivých částech, kterých bylo celkem 15. V tomto případě byl tedy použit obecnější objekt `TFBertModel`, který lze lépe přizpůsobit

[4] Existují také další modely, které se více specializují na slovanské jazyky, například `https://github.com/deepmipt/Slavic-BERT-NER`, avšak nejsou v současnosti zcela kompatibilní s použitými balíčky.

[5] Délka všech vstupů je fixní, avšak většina vět je kratších a tokenizovanou část po konci věty je tedy nutné pro model označit.

pro daný problém. Tento objekt posloužil jako skrytá vrstva, na kterou byly následně napojeny výstupní vrstvy. Pro každou část značky byla přidána jedna výstupní vrstva s počtem neuronů odpovídajícím počtu kategorií dané části značky.

Tento model byl hardwarově nejnáročnější, jelikož obsahoval mnoho výstupních vrstev. Z tohoto důvodu bylo také nezbytné omezit velikost dávek trénovacích dat. Pro natrénování modelu byly jako v předchozím případě využity tři epochy.

5.3 Rozpoznávání dekomponovaných značek s jejich následnou syntézou

Třetí a poslední model byl navržen jako nejjednodušší, jelikož oproti ostatním nevyužíval model BERT. Jako vstup pro trénink tento model využíval výstupní vektory z předchozích modelů obsahující predikované části dekomponované značky. Samotný model se skládá z několika plně propojených vrstev, které využívají aktivační funkci ReLU. Výstupem modelu byla predikce výsledné složené značky, došlo tedy k její následné syntéze.

6 Evaluace

V této kapitole si představíme výsledky jednotlivých modelů. Maximální délka sekvence na vstupu modelu byla po provedení několika experimentů nastavena na 64 tokenů. Zvolená hodnota se ukázala jako dostatečná, jelikož většina vět ze vstupní množiny obsahovala méně slov. Pokud by byla zvolena větší délka sekvence, například 128, bylo by patrně možné dosáhnout ještě lepších výsledků pro přesnost, avšak druhý model by se posléze nevešel do paměti použité grafické karty.

Trénovací množina obsahovala celkem 1171 různých značek. Pro natrénování všech modelů bylo využito 18 782 vět, které dohromady obsahovaly více než 400 tisíc slov. Pro otestování výsledné přesnosti modelů bylo využito 4 696 vět. Přesnost pro jednotlivé modely lze vidět v tabulce 1. Na první pohled

Tabulka 1: Přesnost natrénovaných modelů

	M1	M2	M3
Přesnost	0.939	0.741	0.826

je tedy patrné, že si nejlépe vedl první model, který dosáhl přesnosti **93.9** %. Druhý model dosáhl přesnosti pouze 74.2 %. Jedná se průměr přesností predikce jednotlivých částí značky, které můžeme vidět v tabulce 2. Z té je patrné, že jednodušší části značky dokázal model identifikovat velmi dobře. Ovšem pro části značky, které mají vyšší počet variant model dosáhl přesnosti pouze 50 až 60 procent. Třetí model dosáhl přesnosti téměř 83 procent. Jeho výsledek je

tedy zřejmě omezen kvalitou druhého modelu. Jako nejlepší přístup se tedy jeví první model, který rozpoznává danou značku jako celek.

Tabulka 2: Přesnost pro dekomponované části značky

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Přesnost	0.704	0.548	0.626	0.801	0.511	0.997	0.997	0.521	0.921	0.605	0.582	0.795	0.830	0.692	0.991

7 Závěr

V rámci semestrální práce z předmětu KIV/ANLP se podařilo vytvořit všechny tři zadané modely neuronových sítí a posléze je úspěšně natrénovat na zadanou úlohu, přičemž první a zároveň nejlepší model, který rozpoznával značku jako celek, dosáhl velmi vysoké přesnosti téměř 94 %. Možným vylepšením semestrální práce by bylo natrénování modelu BERT přímo na český jazyk, případně vyzkoušet jiný druh modelu, čímž by pravděpodobně bylo možné dosáhnout dalšího zlepšení přesnosti predikce dané značky.

Seznam tabulek

Reference

[1] SUN, Chi, et al. How to Fine-Tune BERT for Text Classification?. *China National Conference on Chinese Computational Linguistics*. Springer, Cham, 2019. s. 194-206.