Úloha 3:

Diakritika: Automatická obnova diakritiky v slovenskom texte

Príbeh

Ráno. Autobus. Mobil v ruke. Kamarát ti píše:

> *“Ahoj, mozes mi pozicat poznamky z matiky? Zabudol som si zosit doma a ucitelka sa bude pytat. Dakujem!”*

Rozumieš mu, samozrejme. Ale čo keby takúto správu dostal počítač? Alebo čo keby si chcel tento text publikovať v školskom časopise?

Písanie bez diakritiky je na Slovensku bežné – najmä pri rýchlom písaní na mobile, v chatoch, alebo keď používame klávesnicu bez slovenských znakov. Problém nastáva, keď takýto text potrebujeme spracovať automaticky, preložiť, alebo jednoducho urobiť čitateľnejším.

Veta “mama ma rada” môže znamenať:

- “mama má rada” (mama niečo rada robí)
- “mama ma rada” (mama so mnou rada čosi robí, povedzme “oblieka”)
- “mamá mä rada” (no, toto akurát nie je vôbec správne…)

Tvoja úloha bude teda relatívne priamočiara. Vytvoriť inteligentný systém, ktorý dokáže **automaticky obnoviť diakritiku** v slovenskom texte. Využiješ pritom silu moderných jazykových modelov – konkrétne **SlovakBERT**, prvý (nie zas až tak) veľký jazykový model trénovaný špeciálne pre slovenčinu.

Prehľad úlohy

Úloha sa skladá z **troch častí**, ktoré na seba nadväzujú:

Časť	Názov	Body	Popis
1	Príprava dát	30	Extrahovanie a spracovanie trénovacích dát zo slovenskej Wikipédie
2	Zero-shot prístup	20	Obnova diakritiky bez finetunovania, len s využitím predtrénovaného SlovakBERT
3	Finetunovaný model	50	Natrénovanie vlastného modelu na dátach z časti 1

Spoločný vstup: Text v slovenčine s **odstránenou diakritikou**.

Spoločný výstup: Ten istý text so **správne doplnenou diakritikou**.

Príklad

Vstup:
```
Slovensko je krajina v strednej Europe. Hlavne mesto je Bratislava.
```

Očakávaný výstup:
```
Slovensko je krajina v strednej Európe. Hlavné mesto je Bratislava.
```

Časť 1: Príprava dát (30 bodov)

Úloha

Tvojou prvou úlohou je pripraviť **trénovacie dáta** pre model na obnovu diakritiky. Ako zdroj použiješ **slovenské Wikipedia dumpy**, ktoré sú voľne dostupné na:

```
https://dumps.wikimedia.org/skwiki/20251120/
```

Konkrétne potrebuješ súbor `skwiki-latest-pages-articles.xml.bz2`.

Požiadavky

1. **Stiahni a spracuj** slovenský Wikipedia dump
2. **Extrahuj čistý text** z článkov (odstráň wiki markup, šablóny, referencie, infoboxe)
3. **Rozdel' text na vety** (sentence segmentation)
4. **Vytvor páry** (text_bez_diakritiky, text_s_diakritikou):
   - Originálny text = text s diakritikou
   - Vstup = ten istý text so systematicky odstránenou diakritikou
5. **Priprav train/val split** (odporúčame 90/10)

Minimálne požiadavky

- Minimálne **10 000 párov viet**
- Vety by mali mať rozumnú dĺžku (odporúčame 50-2000 znakov)
- Dáta musia byť **reprodukovatel'né** (určite zdokumentuj svoj postup)

Odstránenie diakritiky

Pre vytvorenie vstupných dát odstráň všetku diakritiku podľa tejto mapy:

```
á → a    Á → A
ä → a    Ä → A
č → c    Č → C
ď → d    Ď → D
é → e    É → E
í → i    Í → I
Í → l    Ĺ → L
ľ → l    Ľ → L
ň → n    Ň → N
ó → o    Ó → O
ô → o    Ô → O
ŕ → r    Ŕ → R
š → s    Š → S
ť → t    Ť → T
ú → u    Ú → U
ý → y    Ý → Y
ž → z    Ž → Z
```

Čo odovzdať

1. **Kód** na spracovanie dát (`.py` alebo `.ipynb`)
2. **Štatistiky** vo formáte JSON:

```json
{
  "total_sentence_pairs": 125000,
  "train_pairs": 112500,
  "val_pairs": 12500,
  "avg_sentence_length_chars": 87.3,
  "total_characters": 10937500,
  "diacritic_characters_count": 892341,
  "unique_words": 234567
}
```

1. **Krátka dokumentácia** (README alebo komentáre v kóde) vysvetľujúca:
   - Aké kroky si použil na čistenie dát
   - Ako si riešil problematické prípady (špeciálne znaky, cudzie slová, atď.)

Hodnotenie časti 1

Kritérium	Body
Funkčný extraction pipeline	10
Splnenie min. 10k párov viet	10
Kvalita dát a dokumentácia	5
Reprodukovateľnosť kódu	5

Tipy

- Knižnica `mwparserfromhell` alebo nástroj `wikiextractor` môžu pomôcť s parsovaním wiki markup
- Dávaj pozor na:
  - Cudzojazyčné citáty a názvy
  - Matematické vzorce
  - Tabuľky a zoznamy
  - Neúplné vety

Licencia dát

Dáta zo slovenskej Wikipédie sú dostupné pod licenciou **CC-BY-SA 3.0**. Pri použití je potrebné uviesť zdroj.

Časť 2: Zero-shot prístup (20 bodov)

Úloha

V tejto časti máš obnoviť diakritiku **bez akéhokoľvek finetunovania** modelu. Použiješ predtrénovaný **SlovakBERT** a jeho schopnosť masked language modeling (MLM).

Obmedzenia

- **Musíš použiť** model `gerulata/slovakbert` z HuggingFace
- **Nesmieš model finetunovať** – žiadne úpravy váh, žiadne ďalšie trénovanie
- Môžeš použiť len **inferenčné** operácie
- Môžeš implementovať ľubovoľné **heuristiky a post-processing**

Možné prístupy (inšpirácia)

SlovakBERT je **masked language model** – dokáže predpovedať maskované tokeny v kontexte. Môžeš to využiť napríklad takto:

1. **Iteratívna predikcia:** Postupne maskuj pozície, kde by mohla byť diakritika, a nechaj model predikovať najlepší token
2. **Difúzny prístup:** Začni s textom bez diakritiky, iteratívne vylepšuj predikcie
3. **Kandidátne hodnotenie:** Pre každé slovo vygeneruj kandidátov s rôznou diakritikou a vyber najpravdepodobnejšieho

Toto sú len návrhy – kreatívne riešenia sú vítané!

Pozor na tokenizáciu

SlovakBERT používa **subword tokenizáciu**. Slovo "krásny" môže byť rozdelené na viacero tokenov. Pri práci s diakritikou musíš toto zohľadniť.

```
from transformers import AutoTokenizer, AutoModelForMaskedLM

tokenizer = AutoTokenizer.from_pretrained("gerulata/slovakbert")
model = AutoModelForMaskedLM.from_pretrained("gerulata/slovakbert")

# Príklad tokenizácie
tokens = tokenizer.tokenize("krásny")
# Možný výstup: ['krá', 'sny']
```

Čo odovzdať

1. **Kód** riešenia (`.py` alebo `.ipynb`)
2. **Predikcie** na testovacej sade vo formáte:

```
<id>\t<predikovaný_text>
1 Slovensko je krajina v strednej Európe.
2 Hlavné mesto je Bratislava.
...
```

Hodnotenie

Hodnotí sa **presnosť na znakoch s možnou diakritikou** (viď sekciu Evaluácia).

Výsledky budú zobrazené na **samostatnom leaderboarde**.

Časť 3: Finetunovaný model (50 bodov)

Úloha: Teraz môžeš využiť plnú silu strojového učenia. Tvoja úloha je **natrénovať model** na obnovu diakritiky s využitím dát, ktoré si pripravil v časti 1.

Obmedzenia

- **Musíš použiť** model `gerulata/slovakbert` ako základ
- **Môžeš použiť len dáta** z časti 1 (žiadne externé korpusy okrem skwiki)
- Výpočtové prostriedky: **Google Colab free tier** (T4 GPU, ~12GB RAM, ~12h session)

Možné formulácie problému

1. **Token classification:** Pre každý token predpovedaj, či má diakritiku a akú
2. **Sequence-to-sequence:** Vstup je text bez diakritiky, výstup je text s diakritikou (vyžaduje encoder-decoder)
3. **Character-level:** Pracuj na úrovni znakov, nie tokenov

Príklad: Token classification

```
from transformers import AutoModelForTokenClassification

# Definuj labely pre každý typ diakritiky
# Napríklad: O (žiadna zmena), A_ACUTE (a→á), A_UMLAUT (a→ä), ...

model = AutoModelForTokenClassification.from_pretrained(
    "gerulata/slovakbert",
    num_labels=NUM_LABELS
)
```

Odporúčania

- Začni s jednoduchým baseline a postupne vylepšuj
- Sleduj výkon na validačnej sade
- Experimentuj s:
  - Learning rate (odporúčame 1e-5 až 5e-5)
  - Batch size (podľa dostupnej pamäte)
  - Počet epoch (typicky 3-10)
  - Rôzne formulácie problému

Čo odovzdať

1. **Trénovací kód** (`.py` alebo `.ipynb`)
2. **Predikcie** na testovacej sade (rovnaký formát ako v časti 2)
3. **Krátky report** (voliteľné, ale odporúčané):
   - Aký prístup si zvolil a prečo
   - Aké experimenty si skúšal
   - Výsledky na validačnej sade

Hodnotenie

Hodnotí sa **presnosť na znakoch s možnou diakritikou** (viď sekciu Evaluácia).

Výsledky budú zobrazené na **samostatnom leaderboarde** (oddelene od časti 2).

Evaluácia

Metrika

Hlavná metrika je **presnosť na pozíciách s možnou diakritikou**:

```
accuracy = správne_predikované_znaky /
celkový_počet_pozícií_s_možnou_diakritikou
```

Ktoré pozície sa počítajú?

Počítajú sa **len znaky, ktoré v slovenčine môžu mať diakritiku**:

```
a, A  (môže byť á, ä)
c, C  (môže byť č)
d, D  (môže byť ď)
e, E  (môže byť é)
i, I  (môže byť í)
l, L  (môže byť Í, ľ)
n, N  (môže byť ň)
o, O  (môže byť ó, ô)
r, R  (môže byť ŕ)
s, S  (môže byť š)
t, T  (môže byť ť)
u, U  (môže byť ú)
y, Y  (môže byť ý)
z, Z  (môže byť ž)
```

Príklad výpočtu

Referencia: `Môj pes má rád kosti.` **Predikcia:** `Môj pes ma rád kosti.`

- `o` na pozícii 2 → ref: `ô` , pred: `ô` ✓
- `a` na pozícii 8 → ref: `a` , pred: `a` ✓
- `a` na pozícii 10 → ref: `á` , pred: `a` ✗
- `a` na pozícii 13 → ref: `á` , pred: `á` ✓
- `o` na pozícii 16 → ref: `o` , pred: `o` ✓
- `i` na pozícii 19 → ref: `i` , pred: `i` ✓

Accuracy = 5/6 = 83.3%

Pozície s možnou diakritikou (lowercase pre jednoduchosť):

Formát odovzdania

Súbor s predikciami vo formáte TSV (tab-separated):

```
id text
1 Môj pes má rád kosti.
2 Bratislava je hlavné mesto Slovenska.
```

Technické požiadavky

Prostredie

- **Python 3.10+**
- **Google Colab** (free tier) ako referenčné prostredie

Povolené knižnice

```
torch
transformers
datasets
numpy
pandas
scikit-learn
tqdm
mwparserfromhell (pre časť 1)
```

Ďalšie štandardné knižnice Pythonu sú povolené. Ak potrebuješ niečo špeciálne, spýtaj sa organizátorov.

Model

Musíš použiť `gerulata/slovakbert` :

```python
from transformers import AutoTokenizer, AutoModel

tokenizer = AutoTokenizer.from_pretrained("gerulata/slovakbert")
model = AutoModel.from_pretrained("gerulata/slovakbert")
```

Dokumentácia: https://huggingface.co/gerulata/slovakbert

Odovzdanie

Čo odovzdať

Časť	Súbory
Časť 1	`data_preparation.py` (alebo `.ipynb`), `stats.json` , dokumentácia
Časť 2	`zeroshot_solution.py` , `predictions_zeroshot.tsv`
Časť 3	`finetuning_solution.py` , `predictions_finetuned.tsv` , (voliteľne) report

Kam odovzdať

Edupage, k príslušnému číslu zadania úlohy
Zo všetkých odovzdávaných súborov vytvoriť jeden .zip súbor (nie väčší než 70MB) a ten odovzdať.

Užitočné odkazy

- **SlovakBERT:** https://huggingface.co/gerulata/slovakbert
- **Slovak Wikipedia dumps:** https://dumps.wikimedia.org/skwiki/latest/
- **HuggingFace Transformers dokumentácia:** https://huggingface.co/docs/transformers
- **Wikiextractor:** https://github.com/attardi/wikiextractor

Často kladené otázky (FAQ)

Q: Môžem použiť iný model ako SlovakBERT? A: Nie, musíš použiť `gerulata/slovakbert`.

Q: Môžem v časti 3 použiť dáta z iných zdrojov ako Wikipedia? A: Nie, môžeš použiť len dáta pripravené v časti 1 zo slovenskej Wikipédie.

Q: Čo ak moje riešenie potrebuje viac času/pamäte ako ponúka Colab free? A: Optimalizuj svoje riešenie. Súťaž je navrhnutá tak, aby bola riešiteľná na free tier.

Q: Ako sa počíta skóre pri zhodných výsledkoch? A: Pri rovnakom skóre rozhoduje čas odovzdania.

Bonusové nápady (nehodnotené)

Pre tých, ktorí chcú ísť ďalej:

- **Spoločná oprava diakritiky a preklepov** – čo ak vstup obsahuje aj preklepy?
- **Viacjazyčná verzia** – rozšírenie na češtinu alebo iné jazyky s diakritikou
- **Analýza chýb** – ktoré typy slov/kontextov sú pre tvoj model najťažšie?
- **Porovnanie s ByT5** – ako by sa líšil character-level prístup?

Pošlite priamo organizátorom – isto vaše riešenia radi ďalej rozdebatujú!

Veľa zdaru!