Detekcia prispievateľov do online diskusie typu troll

Samuel Gecík

Sémantický a sociálny web, 2022

Detekcia trollov v diskusiách je proces identifikácie a filtrovania urážlivých alebo rušivých komentárov na online platformách. Môže zahŕňať použitie algoritmov a techník strojového učenia na nahlásenie komentárov, ktoré obsahujú nepravdivé informácie, nenávistné prejavy alebo osobné útoky. Okrem automatizovaných softvérových riešení sa využívajú aj ľudskí moderátori, ktorí môžu tiež kontrolovať komentáre, aby zabezpečili bezpečné prostredie pre používateľov. Pre online platformy je dôležité, aby mali zavedený robustný systém na detekciu trollov, a tým udržiavali zdravú komunitu a vytvorili bezpečný priestor pre všetkých ľudí, ktorí na nich vytvárajú alebo konzumujú obsah.

Cieľ

Vytvorenie programu, ktorý dokáže odhaliť účty trollov v online diskusiách, môže slúžiť pozitívnemu účelu prostrdníctvom podpory zdravej a konštruktívnej online komunikácie. Trollovia sú jednotlivci, ktorí uverejňujú poburujúce, od témy odbočujúce alebo zavádzajúce príspevky v online komunitách s úmyslom narušiť a vyprovokovať ostatných. Týmto konaním vytvárajú toxické prostredie pre ostatných a môžu sťažiť uskutočnenie zmysluplných rozhovorov. Vytvorením programu, ktorý dokáže odhaliť a označiť tieto typy účtov, môže pomôcť zmierniť negatívne účinky trollingu a pomôcť vytvoriť pozitívnejšiu online skúsenosť pre každého. Okrem toho detekcia a nahlásenie takýchto účtov môže tiež pomôcť moderáciám webových stránok prijať potrebné opatrenia. Cieľom tohto zadania bolo vytvorenie aplikácie zabezpečujúcej detekciu používateľov typu troll v komentároch pod videami na platforme YouTube.

Riešenie

Dosiahnuť svoj cieľ sa zadanie snaží prostredníctvom filtrovania najaktívnejších prispievateľov a následnou anlýzou sentimentu v príspevkoch podozrivých užívateľov. Vychádza pritom z predpokladu, že trollie účty odosielajú veľké množstvo komentárov s negatívnym sentimentom.

Analýza sentimentu

Model BERT

Na analýzu sentimentu sme využili model hlbokej neurónové siete typu transformer s názvom BERT, presnejšie jeho predtrénovanú verziu, ktorá bola špecificky trénovaná pre potreby klasifikácie sekvencií. Model bolo ešte potrebné

dotrénovať špeciálne na úlohu klasifikácie sentimentu. Na to sme využili verejne dostupný dataset vhodný práve na našu úlohu.

Dataset

Stanford Semantic Treebank (SST) je databáza syntakticky analyzovaných anglických viet, ktoré boli sémanticky označené výskumníkmi z Univerzity Stanford. Obsahuje viac ako 10 000 vet a 100 000 slov a je navrhnutá na použitie pre tréning a hodnotenie modelov prirodzeného jazyka. Vety v databáze SST sú analyzované pomocou schémy Penn Treebank a obsahujú informácie o predikátoch, argumentoch, koreferencii a kvantifikácii. Táto databáza je cenná pre výskumníkov v oblasti NLP pretože poskytuje veľký a rôznorodý súbor sémanticky označených viet, ktoré môžu byť použité na trénovanie a hodnotenie modelov.

Dataset, ktorý sme použili vychádza zo známeho SST (Stanford Semantic Treebank), konkrétnejšie z jeho verzie pre binárnu klasifikáciu sentimentu. To znamená, že po natrénovaní, by náš model mal byť schopný prijať ľubovoľnú vetu ako vstup a na výstupe klasifikovať či daná veta obsahuje pozitívny alebo negatívny sentiment.

Dataset ako aj predtrénovanú verziu modelu BERT sme získali pomocou platformy Huggingface.

Výsledky trénovania modelu

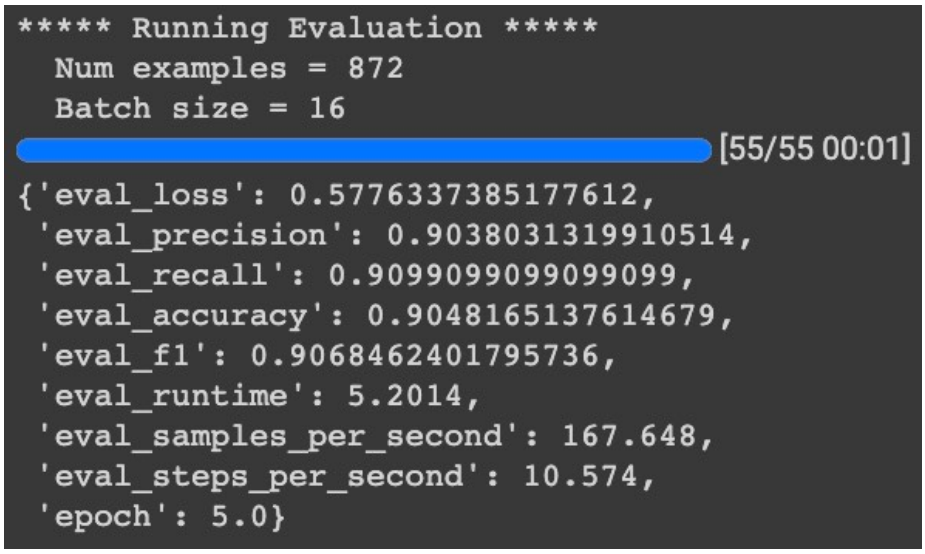
Ladenie modelu na zvolenom datasete prinieslo podľa očakávania dobré výsledky. Na meranie správnosti výsledkov sme použili nasledujúce metriky: presnosť, návratnosť, úspešnosť a F1 skóre.

Výsledky trénovania:

```
***** Running training *****
  Num examples = 67349
  Num Epochs = 5
  Instantaneous batch size per device = 16
  Total train batch size (w. parallel, distributed & accumulation) = 16
  Gradient Accumulation steps = 1
  Total optimization steps = 21050
  Number of trainable parameters = 66955010
[21050/21050 26:50, Epoch 5/5]
```

Epoch	Training Loss	Validation Loss	Precision	Recall	Accuracy	F1
1	0.023700	0.663915	0.868476	0.936937	0.895642	0.901408
2	0.024700	0.577634	0.903803	0.909910	0.904817	0.906846
3	0.030400	0.653282	0.883871	0.925676	0.900229	0.904290
4	0.028100	0.665413	0.887689	0.925676	0.902523	0.906284
5	0.009500	0.783232	0.871036	0.927928	0.893349	0.898582

Výsledky testovania:



Užívateľská príručka

Základný princíp

Užívateľ aplikácie si nastaví YouTube kanály, na ktoré sa chce zamerať, takisto ako aj počet videí, z ktorých budú dáta extrahované. Ďalej taktiež nastaví počet strán komentárov pod videom, ktoré budú stiahnuté. Následne aplikácia pomocou oficiálneho API stiahne všetky žiadané údaje a vytvorí dve jednoduché databázy vo forme CSV súborov.

Zo stiahnutých dát sa vytvorí veľký dátový rámec, v ktorom budú obsiahnuté všetky dáta o videách a komentároch potrebné na nasledujúcu analýzu. V rámci analýzy sú k dispozícii viaceré funkcie, pomocou ktorých si vieme vytvoriť určitý obraz o našich dátach:

- `plot_most_prolific()` - Na stĺpcovom grafe zobrazí počet komentárov pridaných najaktívnejším používateľom pre daný kanál v čase
- `get_top_commenters()` - Vráti zoznam používateľov s počtom komentárov, ktoré pridali, zoradený podľa tohto počtu
- `get_comments()` - Vráti zoznam dátových rámcov s komentármi od n najaktívnejších prispievateľov vo zvolených kanáloch
- `get_sentiment()` - Vracia zoznam slovníkov, v ktorých sú zaznamenané výsledky analýzy sentimentu sprostredkované našim modelom
- `flag_trolls()` - Kombinuje tri predošlé funkcie a označí potenciálnych trollov, ktorých mená vráti ako zoznam

Štruktúra

Hlavná štruktúra aplikácie pozostáva z modulov `analysis.py`, `collect_data.py`, `consts.py` a `datasets.py`.

- `analysis.py` - V tomto module sa nachádzajú funkcie zodpovedné za analýzu dát
- `collect_data.py` - Tu sa nachádzajú funkcie zabezpečujúce získavanie dát pomocou oficiálneho Google API
- `consts.py` - Tento modul obsahuje konštanty definujúce rôzne parametre ako napríklad API kľúč alebo zoznam ID kanálov, ktoré chceme prehľadávať
- `datasets.py` - Definícia triedy, ktorá má na starosti vytváranie hlavného dátové rámca, ako aj synchronizáciu údajov

Jupyter notebooky

Pre prácu s modelom a simulovanie chodu aplikácie sme využili nasledujúce tri Jupyter notebooky:

- `fine_tuned_bert.ipynb` - Tento notebook obsahuje hlavnú časť práce s modelom, jeho trénovanie a testovanie
- `test_inference.ipynb` - V rámci tohto notebooku sme skúšali funkčnosť inferenčnej fázy nášho modelu
- `test.ipynb` - Notebook, v ktorom sme simulovali chod celej aplikácie

Počas behu programu sú takisto podľa zvolených funkcií generované rôzne súbory, ako napríklad už skôr spomenuté CSV databázy.

Doplnkové funkcie

V module `analysis.py` sú definované aj doplnkové funkcie, slúžiace na hlbšiu analýzu dát. Tieto funkcie zahŕňajú metódy ako detekcia rovnakých príspevkov, vektorizácia slov a následné zhlukovanie pojmov podľa podobnosti, ako aj dimenzionálu redukciu a vykreslenie zhlukov na graf. Využívanie týchto funkcií sa však odporúča iba v experimentálnej rovine, vzhľadom na možnú nestabilitu ich fungovania.