PROJECT BESZÁMOLÓ

Bevezetés a Természetes Nyelvű Szövegfeldolgozásba

# Project bemutatása

*Subreddit multiklasszifikáció -* [*GitHub repó*](https://github.com/bczovek/subreddit-multiclassification-nlp)

# Fejlesztési idő

*70 óra*

# Adathalmaz

* [Adathalmaz forrása](https://www.kaggle.com/mswarbrickjones/reddit-selfposts)
* *A példa project adathalmaza 1013000 rekordot tartalmaz.*
* *A rekordokhoz tartozó mezők:* 
  + *Id: Egyedi azonosító*
  + *Subreddit: A poszthoz tartozó subreddit neve*
  + *Title: A poszt címe*
  + *Selftext: A poszt szövege*
* *Az adatok címkézettek.*
* *A címkék többes osztályúak, subreddit alapján.*
* *A hatékonyság növelése érdekében lecsökkentettük a subredditek számát 1013-ról, 102-re. (102000 rekord)*

# Előfeldolgozás

* *Az adatok címkei szövegesek, ezért azokat decimális számmá kell konvertálni.*
* *A Title és Selftext mezőket konkatenáltuk, mert tapasztalatunk szerint a cím is tartalmazhat fontos információt.*
* *Stopword-öket eltávolítottuk. Azért, mert minden szövegben benne vannak és nem jelentenek semmit.*
* *Lemmatizáció alkalmazása. Azért, mert a szavaknak ugyanúgy kell kinéznie, hogy felismerje őket.*
* *Sok html tag-et és linket találtunk. Eltávolító eljárást készítettünk hozzá.*
* *Eltávolítottuk az írásjeleket és a latin ékezetes karaktereket.*
* Tokenizációhoz és a szótár felépítéséhez TF-IDF technikát alkalmaztunk, mert az a hipotézisünk, hogy meg tudja határozni egy szövegben a kulcsszavakat, ami megkönnyíti azok osztályzását.
* Wordcloud és LDA modell segítségével további stopword-öket találtunk, ezeket ignorálva alkottuk meg a szótárunk, melynek mérete 5000 szó.
* *Az adathalmazt felbontottuk Train/val/test, 60/20/20 arányban tanításhoz és kiértékeléshez.*

# 

# Modellek

Három modellt is kipróbáltunk az adatbázisra:

* Logisztikus regressziós modell
* ANN
* RNN

**Logisztikus regressziós modell**

* Bemenete az 5000 dimenziós TF-IDF vektorok.
* A tanítást 100000 iteráción keresztül tart.
* **Kiértékelés**:
  + A tanuló adathalmazon a pontosság 100%
  + A teszt adathalmazon kb. 86%

**ANN**

* Szekvenciális modell, 3 Dense réteggel és 1 Dropout réteggel.
* Bemenete szintén a TF-IDF által előállított vektorok.
* *Az alábbi aktivációs függvényeket használtuk:*
  + *ReLU*
  + *Sotfmax*
* *Loss függvényünk SparseCategorcialCrossEntropy volt mert, a címkéink nem One-Hot kódoltak.*
* *A pontosságot Accuracy-val mértük.*
* *A tanítást 3 cikluson keresztül végeztük, mert a 3. ciklus után túltanulást tapasztaltunk.*
* *A batch méret 32 volt.*
* ***Kiértékelés:***
  + A tanuló adathalmazon a pontosság 96%.
  + A validációs adathalmazon 87%
  + A teszt adathalmazon 84%

**RNN**

* Szekvenciális modell, egy Embedding, egy LSTM és egy Dense réteggel.
* Az Embedding réteg, egy előre feltanított 50 dimenziós GloVe modell-t használ, a TF-IDF vektor 0.1 feletti értékkel rendelkező szavakat ágyazza be.
* *Az alábbi aktivációs függvényeket használtuk:*
  + *Tanh*
  + *Sotfmax*
* *Loss függvényünk SparseCategorcialCrossEntropy volt mert, a címkéink nem One-Hot kódoltak.*
* *A tanítást 5 cikluson keresztül végeztük, mert a 5. ciklus után túltanulást tapasztaltunk.*
* *A batch méret 32 volt.*
* ***Kiértékelés:***
  + A tanuló adathalmazon a pontosság 90%
  + A validációs adathalmazon 84%
  + A teszt adathalmazon 81%

# További fejlesztési lehetősége, Tapasztalatok

*Először Wordpiece tokenizációs technikával próbálkoztunk, a tokenek beágyazását a neurális hálóra bíztuk, ezzel kb 51-52%-os pontosságot értünk el a teszt adathalmazon. Az újabb preprocessing lépések és a TF-IDF hozott ezután jelentős javulást. A modellek paramétereinek változtatása jelentősen nem javította a teljesítményüket, a végleges paramétereik mellett hozták a legjobb eredményt. Tovább fejlesztési lehetőség, hogy a teljes adathalmazra alkalmazzuk, tehát 102 osztály helyett 1013-ra. Erőforrás hiányában kényszerültünk az osztályok számának csökkentésére és a szótár lekorlátozására. Ki lehetne próbálni nagyobb szótárral, illetve magasabb dimenziójú GloVe beágyazást használni a RNN modellnél.*