ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

Εργασία 1

A. Tokens, Types, Zipf's Law

Δίνεται ένα αρχείο (wsj_untokenized.txt) που περιλαμβάνει κείμενα σύντομων ειδήσεων στα Αγγλικά από την εφημερίδα **Wall Street Journal**.

Εφαρμόστε tokenization στο σύνολο των κειμένων του αρχείου με βάση τις ακόλουθες μεθόδους:

- Χρήση του **nltk.word_tokenize()** που συμπεριλαμβάνεται στο NLTK[1].
- Χρήση του μοντέλου για την Αγγλική γλώσσα **en_core_web_sm** που συμπεριλαμβάνεται στο spaCy[2]. Το συγκεκριμένο μοντέλο παρέχει διάφορα επίπεδα ανάλυσης στο κείμενο εισόδου. Στα πλαίσια της εργασίας μας ενδιαφέρει μόνο ο τεμαχισμός του κειμένου σε tokens.
- Χρήση του **BertTokenizer** που συμπεριλαμβάνεται στο HuggingFace[3]. Να χρησιμοποιηθεί η έκδοση **bert-base-cased**[4]. Ο συγκεκριμένος tokenizer έχει προκαθορισμένο λεξιλόγιο και δημιουργεί subword tokens, παρόμοια με τον αλγόριθμο byte-pair encoding.

Για καθεμία από τις παραπάνω μεθόδους tokenization να αναφέρετε τα εξής:

1. Πόσα tokens βρέθηκαν συνολικά;
2. Πόσα διαφορετικά tokens (types) βρέθηκαν;
3. Επιλέξτε τυχαία μία πρόταση και δείξτε την λίστα των tokens που παράγει η μέθοδος.
4. Σε έναν πίνακα, δείξτε τη λίστα με τα 20 πιο συχνά tokens ταξινομημένα σε φθίνουσα συχνότητα. Σε ξεχωριστές στήλες αναφέρετε για το κάθε συχνό token, το πλήθος των εμφανίσεών του, την πιθανότητα εμφάνισής του και το γινόμενο (θέση στη λίστα) * (πιθανότητα εμφάνισης).
5. Ποιο είναι το ποσοστό των tokens που εμφανίζονται ακριβώς μία φορά, ακριβώς 2 φορές και ακριβώς 3 φορές; Πώς συγκρίνονται τα ποσοστά αυτά με τις προβλέψεις του Νόμου του Zipf (σύμφωνα με τον οποίο το ποσοστό των λέξεων που εμφανίζονται ακριβώς n φορές είναι 1/n(n+1));
6. Σύμφωνα με τον Νόμο του Zipf, το γινόμενο της πιθανότητας ενός token επί την θέση του στη λίστα των tokens (ταξινομημένη κατά φθίνουσα σειρά συχνότητας) είναι ίσο με μια σταθερά A. Βρείτε ποια τιμή της σταθεράς A (εξετάστε τις επιλογές: 0.1, 0.2, 0.3, …, 1.0) ταιριάζει καλύτερα με το συγκεκριμένο σύνολο κείμενων. Δημιουργήστε ένα διάγραμμα όπου ο άξονας x είναι η θέση ενός token σε φθίνουσα σειρά συχνότητας και ο άξονας y είναι η συχνότητα του token. Και οι 2 άξονες να ακολουθούν την λογαριθμική κλίμακα. Δείξτε στο ίδιο διάγραμμα με διαφορετικά χρώματα (i) τις προβλέψεις του νόμου του Zipf (με την καλύτερη τιμή που βρέθηκε για τη σταθερά A) και (ii) τις πραγματικές μετρήσεις που έγιναν στα κείμενα.

[1] https://www.nltk.org/
[2] https://spacy.io/
[3] https://huggingface.co/docs/transformers/main_classes/tokenizer
[4] https://huggingface.co/bert-base-cased

B. N-gram Language Models

Σε αυτό το τμήμα της εργασίας θα χρησιμοποιήσουμε και πάλι κείμενα σύντομων ειδήσεων στα Αγγλικά από την εφημερίδα **Wall Street Journal**. Πιο συγκεκριμένα, στο NLTK υπάρχουν διαθέσιμα 199 αρχεία ειδήσεων τα οποία είναι επεξεργασμένα σε διάφορες μορφές. Εμείς θα χρησιμοποιήσουμε το επίπεδο ανάλυσης στο οποίο τα κείμενα έχουν χωριστεί σε tokens και προτάσεις. Παρακάτω φαίνεται ένα παράδειγμα σε Python για το πώς μπορείτε να δείτε το πρώτο κείμενο της συλλογής σε αυτήν την μορφή:

```
>>> from nltk.corpus import treebank
>>> files=treebank.fileids()
>>> len(files)
Out[1]: 199
>>> treebank.sents(files[0])
Out[2]: [['Pierre', 'Vinken', ',', '61', 'years', 'old', ',',
'will', 'join', 'the', 'board', 'as', 'a', 'nonexecutive',
'director', 'Nov.', '29', '.'], ['Mr.', 'Vinken', 'is', 'chairman',
'of', 'Elsevier', 'N.V.', ',', 'the', 'Dutch', 'publishing',
'group', '.']]
```

Θα πρέπει να εκπαιδεύσετε και να αξιολογήσετε τα παρακάτω γλωσσικά μοντέλα ν-γραμμάτων:

- Μοντέλο bigrams με add-k smoothing (k=1)
- Μοντέλο bigrams με add-k smoothing (k=0.01)
- Μοντέλο trigrams με add-k smoothing (k=1)
- Μοντέλο trigrams με add-k smoothing (k=0.01)

Τα πρώτα 170 αρχεία της συλλογής να χρησιμοποιηθούν για την εκπαίδευση των γλωσσικών μοντέλων και τα υπόλοιπα 29 αρχεία να χρησιμοποιηθούν για την αξιολόγηση του μοντέλου.

Στη φάση της εκπαίδευσης του μοντέλου να αντικαταστήσετε όλα τα tokens που εμφανίζονται στο σύνολο των κειμένων εκπαίδευσης λιγότερες από 3 φορές με το ειδικό token <UNK>. Τα υπόλοιπα tokens θα συμπεριλαμβάνονται στο λεξιλόγιο L. Στη φάση της αξιολόγησης, όσα tokens στα κείμενα αξιολόγησης δεν συμπεριλαμβάνονται στο L θα πρέπει να αντικατασταθούν με <UNK>.

Τα n-grams θα πρέπει να εξάγονται στα πλαίσια της κάθε πρότασης (δηλ. να μην υπάρχουν n-grams που περιλαμβάνουν το τέλος μιας πρότασης και την αρχή της επόμενης). Προσθέστε δύο ειδικά tokens (<BOS>, <EOS>) που να υποδηλώνουν την αρχή και το τέλος της κάθε πρότασης. Π.χ., η πρώτη πρόταση από το πρώτο κείμενο της συλλογής που δίνεται παραπάνω:

```
['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', 'will', 'join',
'the', 'board', 'as', 'a', 'nonexecutive', 'director', 'Nov.', '29',
'.']
```

θα μετατραπεί σε:

```
['<BOS>', 'Pierre', 'Vinken', ',', '61', 'years', 'old', ',',
'will', 'join', 'the', 'board', 'as', 'a', 'nonexecutive',
'director', 'Nov.', '29', '.', '<EOS>']
```

και στην περίπτωση των bigrams θα εξαχθούν τα ακόλουθα:

```
('<BOS>', 'Pierre'),
('Pierre', 'Vinken'),
```

```
('Vinken', ','),
  (',', '61'),
        …
('Nov.', '29'),
  ('29', '.'),
 ('.', '<EOS>')
```

Για καθένα από τα 4 παραπάνω μοντέλα ν-γραμμάτων:

1. Χρησιμοποιείστε το μέτρο του perplexity για να συγκρίνετε την επίδοση του μοντέλου στα κείμενα αξιολόγησης:

$$Perplexity = e^{-\frac{1}{N}\sum_i^N \ln p(g_i)}$$

όπου g_i είναι ένα bigram/trigram και N είναι το συνολικό πλήθος των bigrams/trigrams στα κείμενα αξιολόγησης.

2. Εφαρμόστε μετατροπή όλων των κειμένων σε πεζά γράμματα και επαναλάβετε την παραπάνω διαδικασία. Βελτιώνεται ή όχι το perplexity του μοντέλου που προκύπτει;

3. Δημιουργήστε 3 νέες προτάσεις χρησιμοποιώντας το μοντέλο ν-γραμμάτων. Οι νέες προτάσεις θα πρέπει να ξεκινούν με <BOS> (και πρώτες λέξεις της δικής σας επιλογής) και να τερματίζουν σε <EOS>. Η κάθε επόμενη λέξη να επιλέγεται τυχαία αναλογικά με την πιθανότητα του κάθε ν-γράμματος (όσο πιο μεγάλη η πιθανότητα του ν-γράμματος τόσο μεγαλύτερη η πιθανότητα να επιλεγεί η λέξη). Δεν θα πρέπει να περιλαμβάνονται tokens <UNK> στις παραγόμενες προτάσεις. Σχολιάστε την ποιότητα των παραγόμενων προτάσεων.

Χρήσιμα Links:

https://nlp.stanford.edu/IR-book/html/htmledition/zipfs-law-modeling-the-distribution-of-terms-1.html

https://www.nltk.org/book/ch03.html

https://web.stanford.edu/~jurafsky/slp3/3.pdf

https://devopedia.org/n-gram-model

Οδηγίες υποβολής:

Η εργασία είναι **ατομική**. Πρέπει να υποβάλλετε μία αναφορά με τις απαντήσεις στα ερωτήματα της εργασίας καθώς τον κώδικα που χρησιμοποιήσατε. Συστήνεται η χρήση **Jupyter Notebook**[5].

[5] https://jupyter.org/