ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

Εργασία 1

Tokenization

Α. Δίνεται ένα αρχείο με raw text που περιλαμβάνει ειδήσεις από το **Wall Street Journal** (wsj_untokenized.txt). Εφαρμόστε σε αυτό το αρχείο τις ακόλουθες μεθόδους tokenization:

1. Με χρήση του **nltk.word_tokenize()** που συμπεριλαμβάνεται στο NLTK[1].
2. Με χρήση του **nltk.tokenize.wordpunct_tokenize()** που συμπεριλαμβάνεται στο NLTK.
3. Με χρήση του μοντέλου για την Αγγλική γλώσσα **en_core_web_sm** που συμπεριλαμβάνεται στο spaCy[2].
4. Φτιάχνοντας έναν δικό σας tokenizer χρησιμοποιώντας **κανονικές εκφράσεις** (regular expressions).

Θεωρήστε ότι το σωστό (ground-truth) tokenization παρέχεται από τα αρχεία που περιλαμβάνονται στο treebank του NLTK[3].

Β. Δίνεται ένα σύνολο 12 αρχείων με raw text από άρθρα της εφημερίδας **Το Βήμα** (εντός του φακέλου raw). Εφαρμόστε στο σύνολο των αρχείων τις ακόλουθες μεθόδους tokenization:

1. Με χρήση του **nltk.word_tokenize()** που συμπεριλαμβάνεται στο NLTK.
2. Με χρήση του **nltk.tokenize.wordpunct_tokenize()** που συμπεριλαμβάνεται στο NLTK.
3. Με χρήση του μοντέλου για την Ελληνική γλώσσα **el_core_news_sm** που συμπεριλαμβάνεται στο spaCy.
4. Φτιάχνοντας έναν δικό σας tokenizer χρησιμοποιώντας **κανονικές εκφράσεις**.

Θεωρήστε ότι το σωστό (ground-truth) tokenization παρέχεται από τα αντίστοιχα αρχεία στο φάκελο sbd.

Γ. Συγκρίνετε τα αποτελέσματα του tokenization του ground-truth με τις 4 μεθόδους για την καθεμία από τις παραπάνω 2 περιπτώσεις (Αγγλικά και Ελληνικά). Πόσα συνολικά tokens βρέθηκαν και πόσα υπήρχαν; Πόσα διαφορετικά tokens (types) βρέθηκαν και πόσα υπήρχαν; Εντοπίστε συγκεκριμένες περιπτώσεις που τα αποτελέσματα των μεθόδων διαφέρουν. Ποια είναι τα 30 πιο συχνά tokens που προκύπτουν από την καθεμία από τις 4 μεθόδους καθώς και σύμφωνα με το ground truth; Πόσα από τα tokens υπάρχουν στα 30 πιο συχνά όλων των μεθόδων και του ground truth; Ποιο είναι το ποσοστό των tokens που εμφανίζονται ακριβώς μία φορά;

---

[1] https://www.nltk.org/
[2] https://spacy.io/
[3] **from** nltk.corpus **import** treebank
(υπάρχουν 199 αρχεία με τις ειδήσεις που αποτελούν το wsj_untokenize.txt)

Sentence Boundary Disambiguation

A. Εφαρμόστε στο αρχείο wsj_untokenized.txt τις ακόλουθες μεθόδους που τεμαχίζουν το κείμενο σε προτάσεις:

1. Με χρήση του **nltk.sent_tokenize()** που περιλαμβάνεται στο NLTK.
2. Με χρήση του **nltk.tokenize.PunktSentenceTokenizer()** που περιλαμβάνεται στο NLTK.
3. Με χρήση του μοντέλου για την Αγγλική γλώσσα **en_core_web_sm** που συμπεριλαμβάνεται στο spaCy.
4. Φτιάχνοντας έναν δικό σας "sentencizer" χρησιμοποιώντας **κανονικές εκφράσεις**.

Θεωρήστε ότι τα σωστά όρια (ground-truth) των προτάσεων παρέχονται από τα αρχεία που περιλαμβάνονται στο treebank του NLTK.

Β. Εφαρμόστε στο σύνολο των κειμένων από **Το Βήμα** (raw) τις ακόλουθες μεθόδους που τεμαχίζουν το κείμενο σε προτάσεις:

1. Με χρήση του **nltk.word_tokenize()** που συμπεριλαμβάνεται στο NLTK.
2. Με χρήση του **nltk.tokenize.wordpunct_tokenize()** που συμπεριλαμβάνεται στο NLTK.
3. Με χρήση του μοντέλου για την Ελληνική γλώσσα **el_core_news_sm** που συμπεριλαμβάνεται στο spaCy.
4. Φτιάχνοντας έναν δικό σας "sentencizer" χρησιμοποιώντας **κανονικές εκφράσεις**.

Θεωρήστε ότι τα σωστά όρια (ground-truth) των προτάσεων παρέχονται από τα αντίστοιχα αρχεία στο φάκελο sbd.

Γ. Για καθεμία από τις 4 μεθόδους και στις 2 περιπτώσεις (Αγγλικά και Ελληνικά) υπολογίστε τα ακόλουθα μέτρα αξιολόγησης: precision, recall, F-measure[4]. Αναφέρετε τα ακόλουθα για κάθε μέθοδο: ελάχιστο μήκος πρότασης, μέγιστο μήκος πρότασης, μέσο μήκος πρότασης. Τι τιμές προκύπτουν για αυτά αν χρησιμοποιήσουμε ως μονάδα μέτρησης χαρακτήρες και πώς διαμορφώνονται για tokens; Εντοπίστε και σχολιάστε τα λάθη που κάνει η κάθε μέθοδος.

Χρήσιμα Links:

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

https://towardsdatascience.com/5-simple-ways-to-tokenize-text-in-python-92c6804edfc4

Συστήνεται η χρήση **Jupyter Notebook**[5] για την υποβολή της εργασίας.

[4] https://en.wikipedia.org/wiki/Precision_and_recall
[5] https://jupyter.org/