ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

ΕΡΓΑΣΙΑ 5: NLTK Project

ΚΑΓΙΑΣ ΑΝΤΩΝΙΟΣ - aid23003

aid23003@uom.edu.gr

Πίνακας περιεχομένων

Εισαγωγή 3

Μέθοδοι που εφαρμόστηκαν 3

Ερώτημα Α 3

Ερώτημα Β 4

Συμπεράσματα 5

Ερώτημα Α 5

Ερώτημα Β 6

Πηγές 6

Εισαγωγή

Η εργασία αποτελείται από δύο ερωτήματα.

Το αντικείμενο του πρώτου ερωτήματος είναι η χρήση της βιβλιοθήκης NLTK της Python με τελικό στόχο τη δημιουργία 10 προτάσεων από διγράμματα (bigrams) και 10 προτάσεων από τριγράμματα (trigrams). Αρχικά θα φορτωθούν 10 βιβλία από τα οποία θα δημιουργηθεί ένα λεξικό με όλες τις λέξεις που περιλαμβάνονται σε αυτά. Στη συνέχεια, τα κείμενα θα χωριστούν σε προτάσεις και σε tokens, θα υπολογιστούν οι συχνότητες unigrams, bigrams και trigrams και θα δημιουργηθούν οι προτάσεις που αναφέρθηκαν παραπάνω. Για τα bigrams θα πρέπει οριστεί ένα token αρχής πρότασης και ένα token τέλους πρότασης. Για τα trigrams θα πρέπει οριστούν δύο tokens αρχής πρότασης και ένα token τέλους πρότασης.

Όσον αφορά το δεύτερο ερώτημα της παρούσας εργασίας, με τη χρήση και πάλι της βιβλιοθήκης NLTK θα φορτωθεί το dataset `movie_reviews` που περιλαμβάνει 2.000 κριτικές ταινιών μοιρασμένες 50-50 σε θετικές (pos) και αρνητικές (neg) και θα εκπαιδευτεί ένας ταξινομητής ώστε να ταξινομεί σωστά τις κριτικές ανάλογα με το αν είναι θετικές ή αρνητικές.

Παραδοτέα της εργασίας αποτελούν το παρόν report και το αρχείο python που δημιουργήθηκε με όλο τον κώδικα μαζί με τα απαραίτητα σχόλια.

Μέθοδοι που εφαρμόστηκαν

Ερώτημα Α

Αρχικά θα ορίσουμε τα 10 βιβλία που θα χρησιμοποιήσουμε. Τα IDs τους είναι:

- austen-emma.txt
- austen-sense.txt
- blake-poems.txt
- carroll-alice.txt
- chesterton-ball.txt
- chesterton-brown.txt
- melville-moby_dick.txt
- milton-paradise.txt
- shakespeare-hamlet.txt
- whitman-leaves.txt

Στη συνέχεια θα αποθηκεύσουμε τα κείμενα των βιβλίων σε μια μεταβλητή την οποία θα χρησιμοποιήσουμε μετά για να πάρουμε τις προτάσεις τις οποίες μετά θα

αξιοποιήσουμε για να πάρουμε τα tokens. Έπειτα, θα δημιουργήσουμε 3 λίστες για unigrams, bigrams και trigrams και θα υπολογίσουμε για κάθε λίστα τις συχνότητες. Το επόμενο βήμα είναι η δημιουργία 2 λιστών ακόμα για κάθε bigram εκ των οποίων η 1η θα περιέχει την πρώτη λέξη και η 2η θα περιέχει τη δεύτερη λέξη. Αντίστοιχη δουλειά θα γίνει για τα trigrams.

Ακολουθεί η δημιουργία δύο συναρτήσεων οι οποίες θα δημιουργούν τις προτάσεις που θέλουμε από bigrams και trigrams. Η γενική ιδέα είναι ίδια και στις δύο. Αρχικά ορίζουμε μία λίστα στην οποία θα αποθηκευτεί σταδιακά η πρόταση. Στη συνέχεια για τα bigrams επιλέγουμε τυχαία μία λέξη η οποία θα αρχίζει την πρόταση και μέσα σε ένα for loop θα βρίσκουμε κάθε φορά τη δεύτερη λέξη με την οποία αποτελεί bigram μέχρι να συμπληρωθεί ο αριθμός των λέξεων που θέλουμε να περιέχει κάθε πρόταση (π.χ. στην υλοποίησή μας κάθε πρόταση είναι 20 λέξεις). Αντίστοιχη διαδικασία ακολουθείται και για τα trigrams με τη διαφορά ότι ορίζουμε εξ αρχής δύο λέξεις για να ξεκινήσουμε τη διαδικασία (στην υλοποίησή μας οι λέξεις είναι “we will”). Έτσι, σε διπλό for loop αναζητάμε κάθε φορά την τρίτη λέξη με την οποία οι δύο προηγούμενες αποτελούν trigram. Σε κάθε περίπτωση έχει οριστεί η πρόταση να τελειώνει με τελεία.

Ερώτημα Β

Σε πρώτη φάση δημιουργούμε μία συνάρτηση η οποία «καθαρίζει» τις λέξεις μας (`clean_words`), δηλαδή αφαιρεί άχρηστα πράγματα όπως τη στίξη. Ακολούθως, δύο συναρτήσεις για τη δημιουργία λεξικών “bag of words” από unigrams (`bag_of_words`) και bigrams (`bag_of_ngrams`), επειδή όμως δε θέλουμε να διώξουμε τυχόν χρήσιμες λέξεις στα bigrams ως stopwords, θα δημιουργήσουμε ένα νέο set λέξεων που θα αποτελούν stopwords για bigrams και δε θα περιλαμβάνουν λέξεις όπως “only”, “very”, “too” κλπ. αφού θα είναι χρήσιμες για την εκπαίδευση του ταξινομητή. Τέλος, δημιουργούμε μια συνάρτηση που συνδυάζει όλα τα παραπάνω σε ένα (`bag_of_all_words`) και επιστρέφει τα χαρακτηριστικά από unigrams και bigrams, αφού «καθαρίζει» τις λέξεις και δημιουργεί τα αντίστοιχα λεξικά bag of words.

Έπειτα δημιουργούμε λίστες για την αποθήκευση θετικών και αρνητικών κριτικών ξεχωριστά, καθώς και λίστες για την αποθήκευση των χαρακτηριστικών θετικών και αρνητικών κριτικών ξεχωριστά. Στη συνέχεια ανακατεύουμε τις κριτικές μέσα στις λίστες και χωρίζουμε σε train και test sets με τη συνηθισμένη αναλογία 80/20. Το ανακάτεμα πριν τον διαχωρισμό διασφαλίζει ότι κάθε φορά που θα τρέχει ο κώδικας θα έχουμε διαφορετικό αποτέλεσμα στην εκπαίδευση του ταξινομητή και το accuracy που πετυχαίνει. Ο ταξινομητής αυτός θα είναι ένας ταξινομητής Naïve Bayes τον οποίο, αφού εκπαιδεύσουμε πάνω στα train data, θα τον τρέξουμε για τα test data και θα πάρουμε το accuracy που πετυχαίνει. Τέλος, δημιουργούμε 3 διαφορετικές κριτικές και βάζουμε τον ταξινομητή να τις κατατάξει ως θετικές ή αρνητικές.

Συμπεράσματα

Ερώτημα Α

Οι 10 προτάσεις που δημιουργήθηκαν χρησιμοποιώντας bigrams είναι:

1. curries , felled , broken-hearted creatures at thirteen feet with gold , drawers ! ever-push 'd runner , departure--
2. my own look-outs ascended by explaining away with cedars , covering his fauourites flies .
3. know better 'd cornet echoing violence is set in wretched shred e'en for rousing to stars were suddenly swung it
4. flashing upon man alone I wrecked your Cooks , arraigned , Reading that wind roared and Emily With upright Stood
5. Force or enjoying yourself feel capable-- who knows it ca n't mind'em , huge a stiff , snapping furiously
6. empyrean rung up close without step she means it forward voluntarily shipped ; breakfasting on false glitter 'd auk sails
7. producing coolness had her saying little exertion , good his quarry .
8. recognised a lesson done seem determined him want admirers , Quench 'd all heading when leaping forth-- forcibly struck
9. At joust and sailors casting swift tide-rip , admirant , intruding foole can inform her having concluded the chicken ,
10. threw himself almost certain unforeknown .

Οι 10 προτάσεις που δημιουργήθηκαν χρησιμοποιώντας trigrams είναι:

1. We will and , show yield be decay regret allow show not not conquer have bear think meet be be
2. We will ever keep hereafter of ask make he save dance be punctually certainly the drive favour be be leave
3. We will break ever not accept have be tell then , who haue tell travel walk this destroy raise not
4. We will come none permit be not not bear come now return now be bestow armed , seem not come
5. We will now soon obviate die in surely begin comprehend fall my carve yet scarce pine vouchsafe leaue not you
6. We will be carry be each observe but our dismember derive leave favour reign like relent do bear stand not
7. We will interest again show , watch comprehend his one answer worke have grow think be pardon soon do you
8. We will have sometimes not sometimes yield never , not appear ever come not begin be agree think not think

9. We will have ship frequently come point return come probably come crawl surely acquire be be : set Fulfilled they
10. We will .

Ερώτημα Β

Ο ταξινομητής πέτυχε στην περίπτωσή μας accuracy ίσο με 81,5%. Όσον αφορά τα αποτελέσματα για τις 3 κριτικές (τίτλοι κριτικών από χρήστες του IMDb) που δόθηκαν ως είσοδοι για ταξινόμηση, είναι τα παρακάτω:

Κριτική	Ταξινόμηση	Πιθανότητα αρνητικής κριτικής	Πιθανότητα θετικής κριτικής
Starts as buddy movie then derails into a patchwork of clichés	neg	77.54%	22.46%
Crowe & Gosling Shine In This Winning Buddy-Comedy!	pos	8.66%	91.34%
Disjointed, uneven, sporadically funny.	neg	80.02%	19.98%

Πηγές

Bird, S., Klein, E. and Loper, E. (2013). *2. Accessing Text Corpora and Lexical Resources*. [online] Nltk.org. Available at: https://www.nltk.org/book/ch02.html [Accessed 2 Sep. 2023].

Chapagain, M. (2018). *Python NLTK: Sentiment Analysis on Movie Reviews [Natural Language Processing (NLP)]*. [online] Mukesh Chapagain Blog. Available at: https://blog.chapagain.com.np/python-nltk-sentiment-analysis-on-movie-reviews-natural-language-processing-nlp/ [Accessed 2 Sep. 2023].

dierregi (2016). *User-submitted review of 'The Nice Guys' (1)*. [online] IMDb. Available at: https://www.imdb.com/review/rw3567084/?ref_=tt_urv [Accessed 2 Sep. 2023].

namashi_1 (2016). *User-submitted review of 'The Nice Guys' (2)*. [online] IMDb. Available at: https://www.imdb.com/review/rw3481184/?ref_=tt_urv [Accessed 2 Sep. 2023].

tedboy (2016). *NLTK API*. [online] tedboy.github.io. Available at: https://tedboy.github.io/nlps/generated/nltk.html [Accessed 2 Sep. 2023].

wgingery (2016). *User-submitted review of 'The Nice Guys' (3)*. [online] IMDb. Available at: https://www.imdb.com/review/rw3474296/?ref_=tt_urv [Accessed 2 Sep. 2023].