ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Διπλωματική εργασία

Αυτόματη ταξινόμηση κειμένων σε επίπεδα γλωσσομάθειας χρησιμοποιώντας μηχανική μάθηση

Της φοιτήτριας
Ελίνας Αμβροσιάδου
Αρ. Μητρώου:

Επιβλέπων:
Κωνσταντίνος Διαμαντάρας
Βαθμίδα Καθηγητής ΔΙ.ΠΑ.Ε

Θεσσαλονίκη, Φεβρουάριος 2024

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Αυτόματη ταξινόμηση κειμένων σε επίπεδα γλωσσομάθειας χρησιμοποιώντας μηχανική μάθηση

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Αμβροσιάδου Ελίνας του Δημητρίου

Επιβλέπων : Διαμαντάρας Κωνσταντίνος
Καθηγητής ΔΙ.ΠΑ.Ε.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις Choose a date.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Όνομα Επώνυμο	Όνομα Επώνυμο	'Ονομα Επώνυμο
Choose an item.ΔΙ.ΠΑ.Ε.	Choose an item.ΔΙ.ΠΑ.Ε.	Choose an item.ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Φεβρουάριος 2024

(Υπογραφή)

.....

Click here to enter text. Click here to enter text.

Πρόλογος

Η παρούσα διπλωματική ασχολείται με την αυτόματη ταξινόμηση κειμένων σε επίπεδα γλωσσομάθειας για την ελληνική γλώσσα. Αποσκοπεί να διερευνήσει τις τεχνικές που χρησιμοποιούνται στην αναγνωσιμότητα κειμένων (readability) και να αναδείξει τη χρησιμότητα των αλγοριθμων μηχανικής μάθησης στην αυτόματη κατάταξη των άγνωστων κειμένων σε αντίστοιχα επίπεδα γλωσσομάθειας. Τα τελευταία χρόνια, η πρόοδος στην επεξεργασία φυσικής γλώσσας (NLP) και της μηχανικής εκμάθησης έχουν προσφέρει νέες ευκαιρίες για την αυτοματοποίηση και τη βελτίωση της διαδικασίας ταξινόμησης. Εκτός από την χρήση των παραδοσιακών μοντέλων μηχανικής μάθησης που θα χρησιμοποιηθούν στην ταξινόμηση των κειμένων στα πλαίσια της διπλωματικής, θα δοκιμαστεί και η προσέγγιση με τις ενσωματώσεις λέξεων, όπως το Word2Vec αλλά και το BERT. Αυτά τα μοντέλα αποτυπώνουν πιο σύνθετες σημασιολογικές πληροφορίες και γλωσσικά χαρακτηριστικά και θα ήταν χρήσιμο να ελεγθεί η απόδοση τους στη διαδικασία της ταξινόμησης κειμένων σε επίπεδα γλωσσομάθειας.

Περίληψη

Η παρούσα εργασία παρουσιάζει μια μελέτη του προβλήματος που αφορά την αυτόματη ταξινόμηση ελληνικών κειμένων βάσει του Κοινού Ευρωπαϊκού Πλαισίου Αναφοράς για τις Γλώσσες (CERF). Ο προσδιορισμός του επιπέδου του κειμένου είναι πολύ χρήσιμος γιατί βοηθάει τους διδάσκοντες της ελληνικής να αξιολογήσουν το επίπεδο αναγνωσιμότητας οποιουδήποτε ελληνικού κειμένου που πρόκειται να διδάξουν στους μαθητές τους. Για την επίλυση αυτού του προβλήματος, τα κείμενα του συνόλου δεδομένων (dataset) μετατρέπονται σε διανύσματα των 14 γλωσσικών χαρακτηριστικών που έχουν επιλεγεί από το Κέντρο Ελληνικής Γλώσσας. Αυτά τα γλωσσικά χαρακτηριστικά περιλαμβάνουν μετρήσιμους παράγοντες όπως για παράδειγμα ο αριθμός των συλλαβών και των λέξεων ανά πρόταση, ο αριθμός των πολυσύλλαβων λέξεων ενός κειμένου, ο αριθμός των προτάσεων, ο αριθμός των προθημάτων/επιθημάτων κ.λπ., ενδείξεις δηλαδή που μπορούν να υπολογιστούν και μάλιστα με τρόπο αντικειμενικό και χωρίς αποκλίσεις. Η ταξινόμηση των ληφθέντων διανυσμάτων πραγματοποιήθηκε με τυπικούς ταξινομητές μηχανικής μάθησης. Στην εργασία παρουσιάζονται αποτελέσματα τεσσάρων πιο επιτυχημένων: SVM, XGBoost, Logistic Regration, MLP. Η ακρίβεια, η ανάκληση και το F-score σε συνδυασμό με το confusion matrix και καμπύλη ROC χρησίμευσαν ως μετρικές αξιολόγησης. Τα καλύτερα αποτελέσματα ταξινόμησης για τα τρία επίπεδα CEFR τα έδειξε το SVM μοντέλο με ακρίβεια 89,83 % στο σώμα ελέγχου (άγνωστα κείμενα). Στη συνέχεια, εφαρμόστηκε η προσέγγιση με τις ενσωματώσεων λέξεων του Word2Vec. Από τις τρεις παραλλαγές που εφαρμόστηκαν, το πιο αποδοτικό μοντέλο ήταν αυτό που χρησιμοποίησε προ-εκπαιδευμένες ενσωματώσεις μαζί με τα διανύσματα των γλωσσικών χαρακτηριστικών με ακρίβεια 88%. Τέλος, εφαρμόστηκε το γλωσσικό μοντέλου BERT (δύο παραλλαγές). Η παραλλαγή που ήταν η πιο αποδοτική έδειξε ακρίβεια 87,6% στο σώμα ελέγχου. Γενικά, τα αποτελέσματα που προέκυψαν έδειξαν την αποτελεσματικότητα της αυτόματης ταξινόμησης κειμένων σε επίπεδα γλωσσομάθειας και τη δυνατότητα πρακτικής εφαρμογής του.

Λέξεις Κλειδιά: Επεξεργασία φυσικής γλώσσας, Αναγνωσιμότητα, Ταξινόμηση κειμένων, Επίπεδα γλωσσομάθειας, Μηχανική μάθηση, BERT, Word2Vec, Μετασχηματιστές.

Automatic classification of texts according to the levels of CEFR scale using machine learning

Elina Amvrosiadou

Abstract

In this thesis we present a study concerning the automatic classification of Greek texts based on the Common European Framework of Reference for Languages (CERF). Determining the level of the text is very useful because it helps Greek teachers to assess the readability level of any Greek text they are going to teach their students. To solve this problem, we convert the texts of the dataset into vectors of 14 linguistic features selected by the Centre of the Greek Language. These linguistic features include measurable factors such as the number of syllables and words per sentence, the number of polysyllabic words in a text, the number of sentences, the number of prefixes/suffixes, etc., i.e. indicators that can be calculated and indeed in an objective manner and without deviations. Classification of the obtained vectors was performed with standard machine learning classifiers. We present results of four most successful models: SVM, XGBoost, Logistic Regression, MLP. Precision, recall and F-score combined with confusion matrix and ROC curve served as evaluation metrics. The best classification results for the three CEFR levels were shown by the SVM model with an accuracy of 89.83 % on the testing set (unknown texts). Next, the Word2Vec word embeddings approach was implemented. Of the three variants applied, the most efficient model was the one that used pre-trained embeddings along with the linguistic feature vectors with an accuracy of 88%. Finally, the language BERT model (two variants) was applied. The variant that was the most efficient showed an accuracy of 87.6% in the testing set. In general, the obtained results showed the effectiveness of automatic text level detection and the possibility of its practical application.

Keywords: NLP, Readability, Text classification, Machine Learning, Deep Learning, BERT, Word2Vec, Transformers, Scikit learn.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου κ. Κωνσταντίνο Διαμαντάρα, για την ενθάρρυνση, την καθοδήγηση και την υποστήριξή του κατά τη διάρκεια της διπλωματικής μου εργασίας. Οι συμβουλές του ήταν πάντα πολύτιμες και με βοήθησαν με απλό και καθοριστικό τρόπο να κατανοήσω τα βασικά θέματα της μηχανικής μάθησης. Αποτέλεσε πηγή έμπνευσης για μένα και είμαι σίγουρη ότι οι γνώσεις που αποκόμισα θα μου φανούν χρήσιμες τα επόμενα χρόνια. Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου και ιδιαίτερα τον σύζυγό μου για την κατανόηση, την υπομονή και την αμέτρητη συμπαράσταση που μου έδειξε καθόλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Περιεχόμενα

Πρόλογος v
Περίληψη vi
Abstract vii
Ευχαριστίες viii
Περιεχόμενα ix
Κατάλογος Πινάκων xi
Κατάλογος Σχημάτων xii
Κεφάλαιο 1ο Εισαγωγή 1
1.1 Σκοπός της εργασίας 1
1.2 Στόχοι της εργασίας 1
1.3 Δομή της εργασίας 2
Κεφάλαιο 2ο Αναγνωσιμότητα κειμένων 3
2.1 Μαθηματικοί τύποι υπολογισμού της αναγνωσιμότητας 3
2.2 Περιορισμοί των τύπων υπολογισμού αναγνωσιμότητας 5
2.3 Λογισμικό αναγνωσιμότητας του Κέντρου Ελληνικής Γλώσσας 6
2.4 Σχετικές μελέτες και νέες προσεγγίσεις 7
Κεφάλαιο 3ο Θεωρητικό υπόβαθρο 9
3.1 Μετρικές ταξινόμησης 9
3.1.1 Ακρίβεια (Accuracy) 9
3.1.2 Ευστοχία (Precision) 9
3.1.3 Ανάκληση (Recall) 10
3.1.4 Μετρική F1 (F1 score) 11
3.1.5 Υποστήριξη (Support) 12
3.1.6 Πίνακας σύγχυσης (Confusion matrix) 12
3.1.7 Καμπύλη ROC 12
3.2 Αλγόριθμοι Μηχανικής Μάθησης 13
3.2.1 Support Vector Machines 13
3.2.2 Bagging and Boosting Algorithms 16
3.2.3 Random Forest 17
3.2.4 XGBoost (eXtreme Gradient Boosting) 17
3.3 Word embeddings 19
3.3.1 Word2Vec 19

3.4 Βαθιά Μάθηση 24
3.4.1 Αρχιτεκτονική Κωδικοποιητή – Αποκωδικοποιητή 24
3.4.2 Μηχανισμός προσοχής (Attention is all you need) 27
3.4.3 Μετασχηματιστές 27
3.4.4 Ερωτήματα (Queries), Κλειδιά (Keys), Τιμές (Values) 29
3.4.5 Μηχανισμός προσοχής πολλαπλών κεφαλών 31
3.4.6 BERT 31
3.4.7 Προ-εκπαιδευμένα μοντέλα BERT 35
Κεφάλαιο 4ο Μεθοδολογία και πειράματα 38
4.1 Επιλογή του Dataset 38
4.2 Χρήση της βιβλιοθήκης SpaCy 39
4.3 Επιλογή γλωσσικών χαρακτηριστικων 40
4.4 Εξαγωγή επιπλέον μετρικών από το spaCy 49
4.5 Χρήση παραδοσιακών μοντέλων 49
4.5.1 Ρύθμιση Παραμέτρων Μοντέλων Μηχανικής Μάθησης 49
4.5.2 XGBoost 50
4.5.3 Support Vector Machine (SVM) 53
4.5.4 Logistic Regration 55
4.5.5 Multi Layer Perceptron 57
4.6 Word2vec 59
4.7 BERT 60
4.7.1 BertClassifier 60
4.7.2 CustomBertClassifier 65
Κεφάλαιο 5ο Σύνοψη συμπερασμάτων 67
5.1 Συμπεράσματα 67
5.2 Πιθανές επεκτάσεις και μελλοντικές προκλήσεις 67
Βιβλιογραφία 69
Παράρτημα 73

Κατάλογος Πινάκων

Πίνακας 2.1: Αντιστοίχιση του Flesch Reading Ease με τον δείκτη Kincaid για τα ελληνικά . 5

Πίνακας 4.1: Ποσοτικά δεδομένα του συνόλου δεδομένων (dataset) 39

Πίνακας 4.2: Γλωσσικά χαρακτηριστικά που θα υπολογιστούν για την ταξινόμηση των κειμένων 46

Πίνακας 4.3: Μετρική Accuracy με split train 80% / test 20% 59

Πίνακας 4.4: Μετρική accuracy στο testing set (με split train 80%/ test 20%) 60

Πίνακας 4.5: Μετρική accuracy με split train 80% / test 20% 66

Πίνακας 5.1: Συγκεντρωτικός πίνακας απόδοσης μοντέλων με μετρική accuracy 67

Κατάλογος Σχημάτων

Σχήμα 2.1: Δείκτες αναγνωσιμότητας που υπολογίζει το λογισμικό grval 1.1 6

Σχήμα 2.2: Λογισμικό αναγνωσιμότητας του Κέντρου Ελληνικής Γλώσσας 7

Σχήμα 3.1:Χώρος ROC με 4 παραδείγματα προβλέψεων 13

Σχήμα 3.2: Πιθανά υπερεπίπεδα 14

Σχήμα 3.3: Χρήση του Kernel για τον διαχωρισμό δεδομένων 14

Σχήμα 3.4: Αναπαράσταση αλγορίθμων εκμάθησης συνόλου (bagging & boosting) 16

Σχήμα 3.5: Διάγραμμα ροής του XGBoost 19

Σχήμα 3.6: Παράδειγμα του CBOW μοντέλου 20

Σχήμα 3.7:Παράδειγμα λειτουργίας του skip-gram αλγορίθμου 22

Σχήμα 3.8: Αρχιτεκτονική του Κωδικοποιητή –Αποκωδικοποιητή σε seq2seq μοντέλο 24

Σχήμα 3.9: Κάθε κρυφή κατάσταση του αποκωδικοποιητή επηρεάζεται από το νοηματικό πλαίσιο που έχει παραχθεί από τον κωδικοποιητή 25

Σχήμα 3.10: Προσοχή όπως εφαρμόζεται από τον ανθρώπινο εγκέφαλο κατά την οπτική επεξεργασία πληροφοριών 27

Σχήμα 3.11:Δομή του μετασχηματιστή 28

Σχήμα 3.12: Μηχανισμός προσοχής (qkv) 30

Σχήμα 3.13:Προσοχή πολλαπλών κεφαλών 31

Σχήμα 3.14:Ενσωματώσεις εισόδου 32

Σχήμα 3.15:Μηχανισμός προσοχής πολλαπλών κεφαλών 33

Σχήμα 3.16: Προβολές του 1^{ου} και 9^{ου} επιπέδου της κεφαλής προσοχής 34

Σχήμα 3.17:Συσχετισμός των λέξεων με την λέξη *βλέπει* 34

Σχήμα 3.18: Η Αμφίδρομη ιδιότητα του BERT 35

Σχήμα 4.1: Ταξινόμηση κειμένων (Generic pipeline) 38

Σχήμα 4.2:Υπολογισμός αντωνυμικών τύπων 41

Σχήμα 4.3: Υπολογισμός εύκολων λέξεων 41

Σχήμα 4.4:Υπολογισμός μεγάλων λέξεων 42

Σχήμα 4.5:Υπολογισμός λεξιλογικής ποικιλίας 42

Σχήμα 4.6:Υπολογισμός προθημάτων 43

Σχήμα 4.7: Υπολογισμός επιθημάτων 43

Σχήμα 4.8:Υπολογισμός λέξεων μεσοπαθητικής μορφολογίας 43

Σχήμα 4.9: Υπολογισμός κύριων ονομάτων 44

Σχήμα 4.10: Υπολογισμός συνδέσμων 44

Σχήμα 4.11: Υπολογισμός λόγιων επιρρηματικών τύπων 45

Σχήμα 4.12: Υπολογισμός μετοχών 45
Σχήμα 4.13: Υπολογισμός κατάταξης βάση του αριθμού των λέξεων 45
Σχήμα 4.14: Εξαγωγή γλωσσικών χαρακτηριστικών από τα κείμενα..... 47
Σχήμα 4.15: Εκτύπωση των γλωσσικών χαρακτηριστικών του κάθε κειμένου 48
Σχήμα 4.16: Εξαγωγή επιπλέον γλωσσικών χαρακτηριστικών από το spaCy 49
Σχήμα 4.17: Αναζήτηση βέλτιστων υπερπαραμέτρων (XGBoost) 51
Σχήμα 4.18: Βέλτιστες υπερπαράμετροι για τον XGBoost..... 51
Σχήμα 4.19: XGBoost classification reports (train & test sets)..... 51
Σχήμα 4.20: Διάγραμμα κατάταξης χαρακτηριστικών ως προς τη σημαντικότητα 52
Σχήμα 4.21: Καμπύλες ακριβειας και log loss κατά τη διάρκεια της εκπαίδευσης και ελέγχου 52
Σχήμα 4.22: Αναζήτηση βέλτιστων υπερπαραμέτρων (SVM)..... 53
Σχήμα 4.23: Βέλτιστες υπερπαράμετροι (SVM)..... 53
Σχήμα 4.24: SVM classification reports και confusion matrices (train & test sets)..... 54
Σχήμα 4.25: Καμπύλη ROC (SVM)..... 55
Σχήμα 4.26: Βέλτιστες υπερπαράμετροι (Logistic Regration)..... 55
Σχήμα 4.27: Logistic Regration classification reports και confusion matrices (train & test sets) 56
Σχήμα 4.28: Κσμπύλη ROC (Logistic Regration)..... 56
Σχήμα 4.29: Αναζήτηση βέλτιστων υπερπαραμέτρων (MLP) 57
Σχήμα 4.30: Βέλτιστες υπερπαράμετροι (MLP) 57
Σχήμα 4.31: MLP Classification reports και confusion matrices (train & test sets) 58
Σχήμα 4.32: Καμπύλη ROC (MLP) 58
Σχήμα 4.33: 3D αναπαράσταση των CLS embeddings κατά τη διάρκεια των εποχών εκπαίδευσης..... 63
Σχήμα 4.34: 3D αναπαράσταση των CLS embeddings και confusion matrices κατά την διάρκεια της επικύρωσης (validation) 64
Σχήμα 4.35: Συνδυασμός των Bert embeddings μαζί με γλωσσικά χαρακτηριστικά (linguistic features) 65
Σχήμα 4.36: 3D αναπαράσταση των CLS embeddings στο σώμα ελέγχου 66
Σχήμα 4.37: Αποτελέσματα εκπαίδευσης του BertClassifier..... 62
Σχήμα 4.38: Αποτελέσματα εκπαίδευσης του CustomBertClassifier 65

Κεφάλαιο 1ο Εισαγωγή

1.1 Σκοπός της εργασίας

Σε αυτή την εργασία παρουσιάζεται ένα σύστημα αυτόματης ταξινόμησης ελληνικών κειμένων ως προς τον βαθμό δυσκολίας τους στα τρία επίπεδα γλωσσομάθειας βάσει του Κοινού Ευρωπαϊκού Πλαισίου Αναφοράς για τις Γλώσσες (CERF). Αποτελεί πρόβλημα μηχανικής μάθησης με επίβλεψη (supervised learning) και συγκεκριμένα πρόβλημα ταξινόμησης, όπου ως κλάσεις στόχοι (target classes) του συστήματος ορίζονται τα επίπεδα γλωσσομάθειας. Για την εκπαίδευση του μοντέλου, χρησιμοποιήθηκαν σώματα αυθεντικών ελληνικών κειμένων του Κέντρου Ελληνικής Γλώσσας, διαβαθμισμένων εκ των προτέρων στα έξι ισχύοντα επίπεδα ελληνομάθειας με αυστηρά κριτήρια. Σε αυτό το σημείο, θα ήταν χρήσιμο, να τονιστεί ότι στα πλαίσια της διπλωματικής θα χρησιμοποιηθούν τα ευρύτερα τρία επίπεδα γλωσσομάθειας (Α, Β και Γ) αντί για έξι (Α1, Α2, Β1, Β2, Γ1 και Γ2) λόγω μικρού συνόλου δεδομένων.

Στα πλαίσια της εργασίας έγινε μια ολοκληρωμένη διερεύνηση τεχνικών ταξινόμησης κειμένων, συγκρίνοντας την αποτελεσματικότητα των παραδοσιακών μοντέλων, της μεθόδου ενσωμάτωσης λέξεων Word2Vec και των προσεγγίσεων αιχμής που βασίζονται σε μετασχηματιστές όπως το BERT. Τα παραδοσιακά μοντέλα ταξινόμησης κειμένων, που βασίζονται συχνά σε κλασικούς αλγόριθμους μηχανικής μάθησης και εξετάζονται εδώ είναι τα Support Vector Machines (SVM), Logistic Regration, XGBoost, Random Forest και το Multiple Layer Perceptron (MLP), είναι εδώ και πολύ καιρό οι πιο δημοφιλείς λύσεις. Ωστόσο, οι πρόσφατες εξελίξεις στη βαθιά μάθηση, ιδιαίτερα η εμφάνιση προ-εκπαιδευμένων μοντέλων μετασχηματιστών όπως το BERT, έχουν φέρει επανάσταση στον τομέα, επιτυγχάνοντας πρωτοφανή απόδοση σε διάφορες εργασίες NLP.

1.2 Στόχοι της εργασίας

- Εύρεση και επεξεργασία του κατάλληλου σώματος κειμένου για την εκπαίδευση των μοντέλων.

- Μετατροπή των κειμένων σε διανύσματα, υπολογίζοντας τα 14 γλωσσικά χαρακτηριστικά (**handcrafted features**) που έχουν επιλεγεί από το ΚΕΓ ως τα πιο κατάλληλα για την ταξινόμηση ελληνικών κειμένων στα τρία επίπεδα γλωσσομάθειας.

- Εκπαίδευση των παραδοσιακών μοντέλων (SVM, Logistic Regration, MLP, XGBoost και Random Forest) χρησιμοποιώντας τα παραπάνω διανύσματα χαρακτηριστικών (handcrafted features).

- Εκπαίδευση των παραδοσιακών μοντέλων χρησιμοποιώντας τα διανύσματα χαρακτηριστικών που λαμβάνονται από τις **ενσωματώσεις Word2Vec**.

- Εκπαίδευση το παραδοσιακών μοντέλων χρησιμοποιώντας τα διανύσματα χαρακτηριστικών που λαμβάνονται από τις **ενσωματώσεις Word2Vec μαζί με τα handcrafted features**.

- Χρήση προ-εκπαιδευμένου μοντέλου BERT και προσαρμογή (fine-tuning) του στο δικό μας σύνολο δεδομένων. Στη συνέχεια, χρήση των **ενσωματώσεων του [CLS] token**, που περιέχει όλη την πληροφορία για το κείμενο εισόδου, για την ταξινόμηση κειμένων στα 3 επίπεδα.

- Χρήση των παραπάνω **ενσωματώσεων του [CLS] token μαζί με τα handcrafted features** για την ταξινόμηση των κειμένων.

- Σύγκριση των επιδόσεων των μοντέλων

1.3 Δομή της εργασίας

Στο κεφάλαιο 2, γίνεται αναφορά στον όρο **αναγνωσιμότητα (readability)** και περιγράφονται οι πιο γνωστοί μαθηματικοί τύποι υπολογισμού της αναγνωσιμότητας κειμένων όπως Flesch Reading Ease, Flesch-Kincaid, Δείκτης SMOG, Δείκτης Fog. Παρακάτω αναφέρονται οι περιορισμοί που συναντώνται κατά τη χρήση των τύπων αναγνωσιμότητας. Επιπλέον, στο κεφάλαιο αυτό περιγράφεται το λογισμικό αναγνωσιμότητας του Κέντρου Ελληνικής Γλώσσας, το μόνο λογισμικό που χρησιμοποιείται για την ελληνική γλώσσα. Τέλος, αναφέρονται επιγραμματικά σχετικές μελέτες και νέες προσεγγίσεις που αφορούν την αναγνωσιμότητα κειμένων.

Στο κεφάλαιο 3 παρουσιάζεται συνοπτικά το **θεωρητικό υπόβαθρο** που ήταν απαραίτητο για την εκπόνηση της διπλωματικής εργασίας. Αρχικά, περιγράφονται οι μετρικές που χρησιμοποιούνται συχνά στα προβλήματα ταξινόμησης για την αξιολόγηση της απόδοσης των μοντέλων μηχανικής μάθησης. Στη συνέχεια, παρουσιάζονται οι παραδοσιακοί αλγόριθμοι μηχανικής μάθησης που χρησιμοποιούνται τα προβλήματα ταξινόμησης όπως SVM, Random Forest, XGBoost. Επιπλέον, αναφέρεται η τεχνική των word embeddings και η ευρέως χρησιμοποιούμενη μέθοδος Word2Vec. Περιγράφονται οι κύριες αρχιτεκτονικές του: το Continuous Bag of Words (CBOW) και το Skip-gram. Τέλος, αναλύεται το μοντέλο μετασχηματιστή και συγκεκριμένα το BERT μοντέλο.

Το κεφάλαιο 4 συγκεντρώνει όλη τη **μεθοδολογία και τα πειράματα** που ακολουθήθηκαν για την εύρεση του πιο κατάλληλου μοντέλου που θα ταξινομεί τα ελληνικά κείμενα σε αντίστοιχα επίπεδα γλωσσομάθειας. Αρχικά, αναφέρονται τα 14 γλωσσικά χαρακτηριστικά που επιλέχθηκαν και οι απαραίτητες ρουτίνες που αναπτύχθηκαν για τον υπολογισμό τους. Αυτά αποτελούν την είσοδο στα παραδοσιακά μοντέλα ταξινόμησης. Στη συνέχεια, παρουσιάζονται τα αποτελέσματα της εκπαίδευσης και του ελέγχου του κάθε μοντέλου που χρησιμοποιήθηκε. Δίνονται τα απαραίτητα confusion matrices, classification reports και διαγράμματα μετρικών για να αξιολογηθούν τα μοντέλα. Αντίστοιχα, παρουσιάζονται και τα αποτελέσματα από τη χρήση του Word2Vec μοντέλου και των δύο BERT μοντέλων που αναπτύχθηκαν.

Στο κεφάλαιο 5 καταγράφονται τα συμπεράσματα από τη μελέτη της διπλωματικής, καθώς και οι μελλοντικές βελτιώσεις που προτείνονται.

Κεφάλαιο 2ο Αναγνωσιμότητα κειμένων

Γενικά, με τον όρο ***αναγνωσιμότητα (readability)*** του κειμένου ορίζεται η ευκολία της ανάγνωσης ενός κειμένου η οποία προκύπτει από την επιλογή του περιεχομένου του, τη διάρθρωση, το σχεδιασμό του καθώς και την οργάνωση του βάσει του γνωστικού υπόβαθρου του αναγνώστη, των αναγνωστικών του δεξιοτήτων, των ενδιαφερόντων του και των κινήτρων του. [1] Η αναγνωσιμότητα διαφοροποιείται ως έννοια από την ***ευαναγνωσία (legibility)*** η οποία αφορά την ευκολία της ανάγνωσης που προκύπτει από εξωτερικά χαρακτηριστικά του κειμένου όπως οι τυπογραφικοί χαρακτήρες. [2]

Η αναγνωσιμότητα ενός κειμένου παίζει σημαντικό ρόλο στην κατανόηση του κειμένου. Όταν ένα κείμενο έχει υψηλή αναγνωσιμότητα, αυτό σημαίνει ότι ο αναγνώστης του αναμένεται να πετύχει καλύτερη κατανόηση και εκμάθησή του κειμένου. Η εκτίμηση της αναγνωσιμότητας, επομένως της καταλληλότητας των κειμένων, παραδοσιακά γίνεται από τον διδάσκοντα, τον συγγραφέα κ.α. ο οποίος με βάση την εμπειρία του και έχοντας γνώση των δυνατοτήτων του κοινού στο οποίο απευθύνεται, μπορεί να αποφανθεί περί της καταλληλότητας ενός κειμένου. Η αξιοπιστία όμως μιας τέτοιου είδους υποκειμενικής εκτίμησης της αναγνωσιμότητας είναι αμφισβητήσιμη. Έχει αποδειχθεί ότι, αν ζητηθεί από τους δασκάλους να εκτιμήσουν την ηλικία του μέσου αναγνώστη για τον οποίο θεωρούν κατάλληλο ένα κείμενο, οι επιμέρους εκτιμήσεις τους θα αποκλίνουν κατά 6 με 7 χρόνια. Μόνο ο μέσος όρος ενός ικανού αριθμού εκπαιδευτικών είναι αξιόπιστος. Προκύπτει, λοιπόν, η ανάγκη ενός αντικειμενικού τρόπου εκτίμησης της αναγνωσιμότητας και μέσω αυτής της καταλληλότητας των κειμένων. [3]

2.1 Μαθηματικοί τύποι υπολογισμού της αναγνωσιμότητας

Η ανάγκη για έναν γρήγορο, αντικειμενικό και δυνητικά αυτοματοποιημένο τρόπο εκτίμησης της αναγνωσιμότητας οδήγησε στη διατύπωση **μαθηματικών τύπων υπολογισμού της αναγνωσιμότητας (readability formulas)**. Το αποτέλεσμα των τύπων είναι ο **δείκτης αναγνωσιμότητας (readability index)**, ένας αριθμός που αυξάνεται ανάλογα ή αντιστρόφως ανάλογα προς την αναγνωστική δυσκολία του κειμένου. [4]

Μέχρι σήμερα έχει προταθεί μια μεγάλη ποικιλία δεικτών αναγνωσιμότητας. Οι πιο διαδεδομένοι τύποι υπολογισμού που βασίζονται σε έναν αντικειμενικό τρόπο εξαγωγής του βαθμού αναγνωσιμότητας παρουσιάζονται παρακάτω:

- **Flesch Reading Ease**: Ο Flesch, αυστριακός στην καταγωγή και ειδικός στη βιβλιοθηκονομία και την εκπαίδευση ενηλίκων, υπήρξε ο πρώτος που κατάφερε να προβάλει ευρύτερα τη σημασία των μαθηματικών τύπων υπολογισμού της αναγνωσιμότητας. Έχοντας διεξαγάγει διάφορες έρευνες κυρίως σε αναγνωστικό υλικό για ενηλίκους, εισηγήθηκε έναν τύπο [5], ο οποίος περιελάμβανε δύο μέρη: α) Τον δείκτη Reading Ease Score, που χρησιμοποιούσε δύο μεταβλητές, τον αριθμό των συλλαβών και τον αριθμό των προτάσεων ανά 100 λέξεις. Ο δείκτης αυτός προβλέπει την ευκολία της ανάγνωσης σε μια κλίμακα από το 1 έως το 100, όπου το 30 δηλώνει «πολύ δύσκολο» κείμενο και το 70 «εύκολο». [6]

  Ο τύπος περιλαμβάνει το μέσο μήκος της πρότασης (average sentence length, ASL) και τον μέσο αριθμό συλλαβών ανά λέξη (average number of syllables per word, ASW) και υπολογίζει τον δείκτη **Flesch Reading Ease Score (FRES)** που φαίνεται παρακάτω:

$$\text{FRES} = 206.835 - 1.015 * ASL - 84.6 * ASW$$

Εάν αναλύσουμε τον τύπο, έχουμε:

$$\text{FRES} = 206.835 - 1.015 * \frac{total\ words}{total\ sentences} - 84.6 * \frac{total\ syllables}{total\ words}$$

Όσο υψηλότερη είναι η βαθμολογία, τόσο πιο εύκολα διαβάζεται το κείμενο.

Στις αρχές της δεκαετίας του 1980 ο Γαγάτσης ήταν ο πρώτος που ασχολήθηκε με την αναγνωσιμότητα ελληνικών κειμένων, μελετώντας την αναγνωσιμότητα των σχολικών εγχειριδίων των Μαθηματικών και προσάρμοσε τον **Flesch Reading Ease Score στα ελληνικά**. [7] Στηριζόμενος στην παρατήρηση ότι οι ελληνικές λέξεις είναι κατά μέσο όρο μεγαλύτερες από τις αγγλικές ή τις γαλλικές, αντικατέστησε το συντελεστή 84,6 του μέσου αριθμού συλλαβών ανά λέξη με τον 59. Έτσι ο τύπος Flesch Reading Ease Score προσαρμοσμένος στα ελληνικά αναδιαμορφώνεται ως εξής:

$$\text{FRES} = 206.835 - 1.015 * \frac{total\ words}{total\ sentences} - 59 * \frac{total\ syllables}{total\ words}$$

- **Επίπεδο βαθμού Flesch-Kincaid**: Το 1975, στο πλαίσιο της έρευνας χρηματοδοτούμενης από το αμερικανικό πολεμικό ναυτικό, τροποποιήθηκε ο παραπάνω τύπος Reading Ease του Flesch, ώστε ο δείκτης αναγνωσιμότητας που θα προκύπτει να αντιστοιχίζει τα επτά επίπεδα της αξιολογικής κλίμακας του Flesch με τις διάφορες βαθμίδες εκπαίδευσης των ΗΠΑ. [8] Ο νέος, λοιπόν, τύπος υπολογισμού αναγνωσιμότητας, γνωστός ως Flesch-Kincaid, ή απλώς Kincaid, είναι ο εξής [3]:

$$\text{G} = 0.39 * \frac{total\ words}{total\ sentences} + 11.8 * \frac{total\ syllables}{total\ words} - 15.59$$

Όπου G: τάξη (του εκπαιδευτικού συστήματος των ΗΠΑ) για την οποία είναι κατάλληλο το κείμενο. [6]

Η αντιστοίχιση μεταξύ της αξιολογικής κλίμακας του Flesch Reading Ease και των εκπαιδευτικών βαθμίδων του δείκτη Kincaid, όπως προσαρμόστηκαν στα ελληνικά από τους Ευσταθιάδη κ.ά. (2002), είναι η εξής [9]:

Flesch Reading Ease		**Flesch-Kincaid Grade Level**
100-90	πολύ εύκολο	Α΄-Β΄-Γ΄ Δημοτικού
90-80	εύκολο	Δ΄-Ε΄-ΣΤ΄ Δημοτικού
80-70	αρκετά εύκολο	Α΄-Β΄ Γυμνασίου

70-60	μέσο	Γ΄ Γυμν. Α΄ Λυκείου
60-50	αρκετά δύσκολο	Β΄- Γ΄ Λυκείου
50-30	δύσκολο	ΑΕΙ – ΤΕΙ
30-0	πολύ δύσκολο	ΑΕΙ – ΤΕΙ

Πίνακας 2.1: Αντιστοίχιση του Flesch Reading Ease με τον δείκτη Kincaid για τα ελληνικά

Ενδεικτικό της ευρείας διάδοσης του συγκεκριμένου δείκτη είναι το γεγονός ότι εφαρμόστηκε και σε άλλες γλώσσες εκτός των αγγλικών, όπως τα γαλλικά, τα γερμανικά, τα ολλανδικά και τα ιταλικά.

- **Δείκτης SMOG**: Το SMOG σημαίνει "Simple Measure of Gobbledygook". Υπολογίζει τα χρόνια εκπαίδευσης που χρειάζεται ένα άτομο για να κατανοήσει μια γραφή.

$$G = 1.0430\sqrt{complex\ words * \left(\frac{30}{sentences}\right)} + 3.1291$$

Όπου, complex words θεωρούνται λέξεις με 3 ή παραπάνω συλλαβές.

- **Δείκτης Fog**: Λίγο μετά τις πρωτοποριακές έρευνες του Flesch, ο Robert Gunning (1952), ένας Αμερικανός επιχειρηματίας που είχε ασχοληθεί με την έκδοση εφημερίδων και σχολικών βιβλίων, εισηγήθηκε τον δείκτη Fog. Ο δείκτης έγινε ιδιαίτερα δημοφιλής χάρη στην πολύ απλή μορφή του.

Με τις έρευνές του ο Gunning καθιέρωσε τη χρήση δεικτών αναγνωσιμότητας σε πολλές εφημερίδες και περιοδικά των ΗΠΑ με σκοπό την αποφυγή της «περιττής στρυφνότητας», η οποία, σύμφωνα με τους ισχυρισμούς του, αποτελεί την κύρια αιτία της αναγνωστικής αδυναμίας των μαθητών.

Ο δείκτης υπολογίζει τα χρόνια τυπικής εκπαίδευσης που χρειάζεται ένα άτομο για να κατανοήσει το κείμενο κατά την πρώτη ανάγνωση. Για παράδειγμα, ένας δείκτης Fog 12 απαιτεί επίπεδο ανάγνωσης ενός τελειόφοιτου γυμνασίου των ΗΠΑ (περίπου 18 ετών). Χρησιμοποιείται συνήθως για να επιβεβαιώσει ότι το κείμενο μπορεί να διαβαστεί εύκολα από το κοινό για το οποίο προορίζεται. Τα κείμενα για ένα ευρύ κοινό χρειάζονται γενικά δείκτη μικρότερο από 12. Τα κείμενα που απαιτούν σχεδόν καθολική κατανόηση χρειάζονται γενικά δείκτη μικρότερο από 8. [10]

$$GFI = 0.4\left[\left(\frac{words}{sentences}\right) + 100\left(\frac{complex\ words}{words}\right)\right]$$

2.2 Περιορισμοί των τύπων υπολογισμού αναγνωσιμότητας

Παρά την ευρύτατη εξάπλωσή τους ή ίσως ακριβώς λόγω αυτής, οι τύποι υπολογισμού της αναγνωσιμότητας έχουν δεχθεί αυστηρή κριτική, μολονότι αποτελούν έναν γρήγορο και

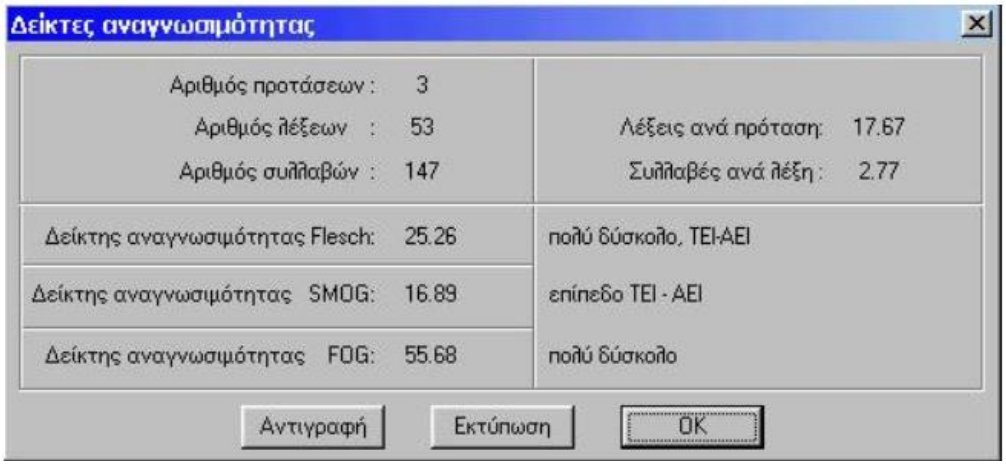
αποτελεσματικό τρόπο υπολογισμού της αναγνωσιμότητας. [3] Ουσιαστικά, δεν παρέχουν την τέλεια πρόβλεψη, αλλά μια κατά προσέγγιση εκτίμηση της αναγνωστικής δυσκολίας ενός κειμένου, γεγονός που αποδεικνύεται και από τους διαφορετικούς δείκτες αναγνωσιμότητας που δίνουν οι διάφοροι τύποι για το ίδιο κείμενο. [9] Για αυτό το λόγο κατά την αποτίμηση της δυσκολίας ενός κειμένου πρέπει να συνεκτιμώνται και άλλοι παράγοντες, όπως η δομή, το λεξιλόγιο, η σύνταξη, η συνοχή και η συνεκτικότητα του κάθε κειμένου.

2.3 Λογισμικό αναγνωσιμότητας του Κέντρου Ελληνικής Γλώσσας

Οσον αφορά την αναγνωσιμότητα των ελληνικών κειμένων, το Κέντρο Ελληνικής Γλώσσας (ΚΕΓ) ανέπτυξε το λογισμικό **grval 1.1** που υπολογίζει το βαθμό αναγνωσιμότητας κειμένων στα νέα ελληνικά με βάση τους 4 δείκτες: τον FleschReadingEase, Flesch-KincaidGradeLevel, SMOG και FleschFogIndex.

Σε αυτό το σημείο τονίζεται ότι οι δείκτες αναγνωσιμότητας που εφαρμόζονται στην αγγλική γλώσσα δεν μπορούν να χρησιμοποιηθούν αυτούσιοι χωρίς προηγουμένως να προσαρμοστούν στην ελληνική γλώσσα. Καθώς οι δείκτες FleschReadingEase, Flesch-KincaidGradeLevel, SMOG και FleschFogIndex είναι προσαρμοσμένοι στα μορφολογικά στοιχεία της αγγλικής γλώσσας, είναι σαφές ότι δε θα μπορούν να αξιολογήσουν ελληνικά κείμενα ως προς την αναγνωσιμότητά τους βασισμένοι στα μορφολογικά χαρακτηριστικά της αγγλικής. Αναπόφευκτα οι ερευνητές του ΚΕΓ προσάρμοσαν την κλίμακα των δεικτών Flesch Reading Ease, SMOG και Flesch Fog Index στα νέα ελληνικά.

Η τελική μορφή που παίρνει το grval 1.1 μετά την εισαγωγή του κειμένου προς εξέταση είναι η εξής:



Σχήμα 2.1: Δείκτες αναγνωσιμότητας που υπολογίζει το λογισμικό grval 1.1

Το 2012 το ΚΕΓ ανέπτυξε το **νέο λογισμικό** αναγνωσιμότητας ελληνικών κειμένων στο πλαίσιο του συγχρηματοδοτούμενου προγράμματος «Πράξη 54: Πιστοποίηση ελληνομάθειας: υποστήριξη και ποιοτική ανάδειξη της διδασκαλίας/εκμάθησης της ελληνικής ως ξένης/δεύτερης γλώσσας» και συνιστά επανασχεδιασμό του παλαιότερου ανάλογου λογισμικού grval 1.1 της «Πύλης για την Ελληνική Γλώσσα».

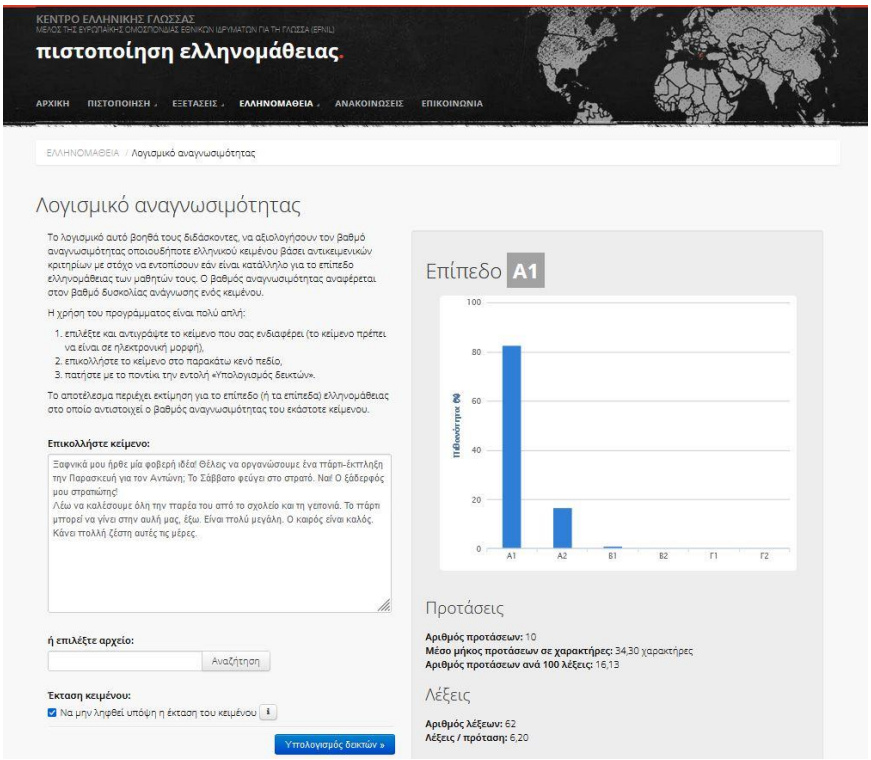
Η υλοποίηση του νέου λογισμικού αναγνωσιμότητας διενεργήθηκε σε τρεις φάσεις:

1. Προσδιορισμός των κειμενικών παραμέτρων της αναγνωσιμότητας από ειδική ερευνητική ομάδα γλωσσολόγων. Τελικά προκρίθηκαν 14 διαφορετικές παράμετροι,

στις οποίες περιλαμβάνονται: α) παράμετροι που χρησιμοποιούν μερικοί από τους πιο διαδεδομένους τύπους υπολογισμού του βαθμού αναγνωσιμότητας διεθνώς, όπως οι FleschReadingEase, Flesch-KincaidGradeLevel, SMOG και FleschFogIndex, τις οποίες χρησιμοποιούσε και το παλαιότερο λογισμικό, και β) επιπλέον παράμετροι, βάσει πρόσφατης σχετικής βιβλιογραφίας ειδικά για την ελληνική γλώσσα.

2. Εφαρμογή του αλγορίθμου της πολυωνυμικής λογιστικής παλινδρόμησης (multinomial logistic regression) χρησιμοποιώντας τις παραπάνω κειμενικές παραμέτρους για την εκπαίδευση ενός μοντέλου ταξινόμησης των κειμένων σε επίπεδα ελληνομάθειας. Το μοντέλο εκπαιδεύτηκε σε σώμα κειμένων ειδικά διαβαθμισμένων ανά επίπεδο ελληνομάθειας και ήδη χρησιμοποιημένων από συνεργάτες της πιστοποίησης ελληνομάθειας σε εξεταστικά θέματα κατανόησης γραπτού και προφορικού λόγου (586 κείμενα εκπαίδευσης και 191 ελέγχου). [11]

3. Ανάπτυξη σχετικής ιστοσελίδας με ενσωματωμένο τον νέο τύπο υπολογισμού αναγνωσιμότητας προς χρήση του κοινού που φαίνεται παρακάτω.



Σχήμα 2.2: Λογισμικό αναγνωσιμότητας του Κέντρου Ελληνικής Γλώσσας

2.4 Σχετικές μελέτες και νέες προσεγγίσεις

Οι Collins-Thompson & Callan (2004) ήταν μεταξύ των πρώτων που προσπάθησαν να εξετάσουν το πρόβλημα της ταξινόμησης των κειμένων ανάλογα με τη δυσκολία τους από την άποψη της στατιστικής μοντελοποίησης της γλώσσας. Έχουν δείξει ότι η δυσκολία στην ανάγνωση μπορεί να εκτιμηθεί με μια απλή προσέγγιση μοντελοποίησης της γλώσσας χρησιμοποιώντας έναν τροποποιημένο Naive Bayes ταξινομητή. [12]

Οι Schwarm & Ostendorf (2005) επέκτειναν αυτή τη μέθοδο χρησιμοποιώντας πολλά γλωσσικά μοντέλα, NLP εξόδους (δέντρα ανάλυσης, ετικέτες PoS) και ορισμένες «κλασικές» μετρικές αναγνωσιμότητας (μήκος πρότασης και μήκος λέξης). Ο συνδυασμός των πληροφοριών από στατιστικά γλωσσικά μοντέλα, τις παραδοσιακές μετρικές υπολογισμού αναγνωσιμότητας αλλά και άλλα εργαλεία επεξεργασίας γλώσσας αύξησαν την ακρίβεια εκτίμησης του επιπέδου αναγνωσιμότητας. [13]

Οι Pitler & Nenkova (2008) αύξησαν το σύνολο των χαρακτηριστικών που λαμβάνονται υπόψη για την πρόβλεψη του επιπέδου αναγνωσιμότητας, προσθέτοντας χαρακτηριστικά που βασίζονται στον λόγο (discourse-based features). Τα πειράματα έδειξαν πολλά υποσχόμενα αποτελέσματα για προβλήματα ταξινόμησης και παλινδρόμησης. [14]

Οι πρόσφατες εξελίξεις στη βαθιά μάθηση και τα μοντέλα με ενσωματώσεις λέξεων έχουν επίσης αρχίσει να επηρεάζουν την ανάλυση αναγνωσιμότητας. Ένας αριθμός μελετών έχει ήδη εφαρμόσει διάφορες αρχιτεκτονικές βαθιών νευρωνικών δικτύων και μοντέλα μετασχηματιστών με σκοπό την εκτίμηση της αναγνωσιμότητα σε μονόγλωσσα (Mohammadi & Khasteh 2019) [15] και πολύγλωσσα κείμενα (Azpiazu & Pera 2019) [16] με αξιοσημείωτη επιτυχία.

Κεφάλαιο 3ο Θεωρητικό υπόβαθρο

3.1 Μετρικές ταξινόμησης

Τα προβλήματα ταξινόμησης συναντιούνται σε διάφορους τομείς, συμπεριλαμβανομένων των οικονομικών, της υγειονομικής περίθαλψης, του μάρκετινγκ, της αναγνώρισης εικόνας, της αναγνώρισης ομιλίας, της επεξεργασίας φυσικής γλώσσας και άλλων. Τα μοντέλα ταξινόμησης έχουν διακριτή έξοδο, επομένως χρειαζόμαστε μια μετρική που να μπορεί να συγκρίνει τις διακριτές κλάσεις με κάποιο τρόπο με σκοπό να βρεθεί το καλύτερο μοντέλο που παρουσιάζει τα πιο ικανοποιητικά αποτελέσματα. Οι μετρικές ταξινόμησης δείχνουν πόσο καλή ή κακή είναι η ταξινόμηση, αλλά καθεμία από αυτές την αξιολογεί με διαφορετικό τρόπο. Η επιλογή του σε ποιες μετρικές θα δοθεί προτεραιότητα εξαρτάται από τους συγκεκριμένους στόχους και απαιτήσεις του προβλήματος. Είναι σύνηθες να εξετάζεται ένας συνδυασμός μετρικών για μια πιο ολοκληρωμένη κατανόηση της απόδοσης του μοντέλου.

3.1.1 Ακρίβεια (Accuracy)

Η ακρίβεια (accuracy) αποτελεί μία μετρική που εφαρμόζεται συχνά για την αξιολόγηση μοντέλων ταξινόμησης. Συγκεκριμένα είναι ο λόγος του πλήθους των σωστών προβλέψεων που έκανε το μοντέλο προς το πλήθος όλων των προβλέψεων. Στην περίπτωση της δυαδικής ταξινόμησης δίνεται από τον τύπο:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP + TF}{TP + FP + TN + FN}$$

- TP (true positive) είναι οι σωστές θετικές προβλέψεις,
- TN (true negative) οι σωστές αρνητικές προβλέψεις,
- FP (false positive) οι λανθασμένες θετικές προβλέψεις και
- FN (false negative) οι λανθασμένες αρνητικές προβλέψεις.

Είναι σημαντικό να σημειωθεί ότι η ακρίβεια είναι μια απλή μετρική και μπορεί να μην είναι αρκετή σε όλες τις περιπτώσεις, ειδικά όταν πρόκειται για μη ισορροπημένα (unbalanced) σύνολα δεδομένων, όπου έχουμε μεγάλες διαφορές ανάμεσα στις κλάσεις. Σε τέτοιες περιπτώσεις, άλλες μετρικές όπως η ευστοχία, η ανάκληση, η F1-score ή η καμπύλη ROC ενδέχεται να παρέχουν μια πιο ολοκληρωμένη εικόνα της απόδοσης του μοντέλου. Η επιλογή των μετρικών αξιολόγησης εξαρτάται από τα συγκεκριμένα χαρακτηριστικά του προβλήματος κάθε φορά. [17], [18]

3.1.2 Ευστοχία (Precision)

Η ευστοχία είναι μια μετρική που χρησιμοποιείται στην ταξινόμηση (δυαδική ή πολλών κλάσεων) και μετρά την ακρίβεια των θετικών προβλέψεων που γίνονται από ένα μοντέλο. Ορίζεται ως ο λόγος των σωστών θετικών προβλέψεων προς τον συνολικό αριθμό των περιπτώσεων που προβλέπονται ως θετικές, ανεξάρτητα από το αν η θετική πρόβλεψη ήταν σωστή ή όχι. Η ευστοχία είναι ιδιαίτερα χρήσιμη σε προβλήματα όπου το κόστος των ψευδώς θετικών είναι υψηλό. Ο τύπος της ευστοχίας είναι:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Ας εξετάσουμε ένα πρακτικό παράδειγμα που σχετίζεται με το κόστος των ψευδώς θετικών περιπτώσεων. Ας πούμε ότι έχουμε ένα πρόβλημα που αφορά τη δημιουργία φίλτρου ανεπιθύμητης αλληλογραφίας για μια υπηρεσία email. Σε αυτό το σενάριο έχουμε:

- True Positive (TP): Το φίλτρο ταξινομεί σωστά ένα spam email ως spam.
- False Positive (FP): Το φίλτρο ταξινομεί εσφαλμένα ένα κανονικό email ως ανεπιθύμητο.
- True Negative (TN): Το φίλτρο ταξινομεί σωστά ένα κανονικό email ως μη ανεπιθύμητο.
- False Negative (FN): Το φίλτρο ταξινομεί εσφαλμένα ένα spam email ως μη spam.

Τώρα, ας εξετάσουμε το κόστος που σχετίζεται με τα ψευδώς θετικά και τα ψευδώς αρνητικά σε αυτό το πλαίσιο:

Υψηλό κόστος ψευδώς θετικών: Εάν το φίλτρο επισημαίνει εσφαλμένα ένα κανονικό μήνυμα ηλεκτρονικού ταχυδρομείου ως ανεπιθύμητο (FP), ο χρήστης ενδέχεται να χάσει σημαντικά μηνύματα ηλεκτρονικού ταχυδρομείου, γεγονός που θα μπορούσε να οδηγήσει σε χαμένες ευκαιρίες ή διακοπή της επικοινωνίας. Οι χρήστες ενδέχεται να απογοητευτούν με την υπηρεσία ηλεκτρονικού ταχυδρομείου εάν αντιμετωπίζουν συνεχώς ψευδώς θετικά αποτελέσματα, γεγονός που ενδεχομένως να οδηγήσει σε απώλεια εμπιστοσύνης στο φίλτρο ανεπιθύμητης αλληλογραφίας.

Χαμηλότερο κόστος ψευδώς αρνητικών: Εάν το φίλτρο λανθασμένα επιτρέπει σε ένα ανεπιθύμητο email να φτάσει στα εισερχόμενα (FN), ο χρήστης μπορεί να δει ορισμένα ανεπιθύμητα μηνύματα ηλεκτρονικού ταχυδρομείου στα εισερχόμενά του, αλλά οι συνέπειες είναι συνήθως λιγότερο σοβαρές από το να χάσει τα σημαντικά κανονικά email. Οι χρήστες μπορούν να αναγνωρίσουν και να διαγράψουν με μη αυτόματο τρόπο τα ψευδώς αρνητικά, αλλά τα ψευδώς θετικά ενδέχεται να δυσαρεστήσουν αρκετά τον χρήστη.

Σε αυτό το σενάριο, η ευστοχία είναι ζωτικής σημασίας επειδή εστιάζει στην ελαχιστοποίηση των ψευδώς θετικών. Η υψηλή ευστοχία υποδεικνύει ότι όταν το φίλτρο ανεπιθύμητης αλληλογραφίας ταξινομεί ένα email ως ανεπιθύμητο, είναι πιθανό να είναι σωστό, μειώνοντας την πιθανότητα ψευδούς επισήμανσης σημαντικών μηνυμάτων ηλεκτρονικού ταχυδρομείου ως ανεπιθύμητων. Σε περιπτώσεις όπου το κόστος των ψευδώς θετικών είναι υψηλό, η ευστοχία γίνεται βασικός δείκτης βελτιστοποίησης, διασφαλίζοντας καλύτερη εμπειρία χρήστη και διατηρώντας την εμπιστοσύνη στο φίλτρο ανεπιθύμητης αλληλογραφίας. [17]

3.1.3 Ανάκληση (Recall)

Η ανάκληση, γνωστή και ως ευαισθησία, είναι μια μετρική που χρησιμοποιείται στην ταξινόμηση και ο σκοπός της είναι να υπολογίζει πόσες από τις τιμές που είναι πραγματικά θετικές εκτιμήθηκαν ως θετικές από το μοντέλο. Επικεντρώνεται στην ελαχιστοποίηση των ψευδώς αρνητικών, καθιστώντας το ιδιαίτερα χρήσιμο όταν το κόστος της απώλειας θετικών περιπτώσεων είναι υψηλό.

Ο τύπος της ανάκλησης δίνεται από τον παρακάτω τύπο:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Η ανάκληση απαντά στην ερώτηση: "Από όλες τις περιπτώσεις που είναι πραγματικά θετικές, πόσες εντόπισε σωστά το μοντέλο;".

Οι τιμές ανάκλησης κυμαίνονται από το 0 έως το 1, όπου το 1 υποδηλώνει τέλεια ανάκληση (όλες οι θετικές περιπτώσεις προσδιορίζονται σωστά) και το 0 υποδηλώνει καμία ανάκληση (καμία θετική περίπτωση δεν αναγνωρίζεται σωστά).

Στο πλαίσιο της ταξινόμησης πολλαπλών κλάσεων, η ανάκληση μπορεί να υπολογιστεί για κάθε κλάση ξεχωριστά. Η μέση ανάκληση σε όλες τις κατηγορίες μπορεί επίσης να υπολογιστεί, ανάλογα με τη συγκεκριμένη περίπτωση χρήσης.

Η ανάκληση είναι ιδιαίτερα σημαντική σε προβλήματα όπου το κόστος των ψευδώς αρνητικών είναι υψηλό. Για να γίνει περισσότερο κατανοητό, δίνεται ένα παράδειγμα:

Ας πούμε ότι έχουμε ένα μοντέλο μηχανικής μάθησης που εντοπίζει εισβολές δικτύου σε ένα σύστημα ασφάλειας στον κυβερνοχώρο. Στόχος είναι ο εντοπισμός κακόβουλων δραστηριοτήτων και πιθανών παραβιάσεων ασφάλειας. Τα ψευδώς αρνητικά σε αυτό το παράδειγμα, όπου το μοντέλο αποτυγχάνει να εντοπίσει μια πραγματική εισβολή, θα μπορούσαν να έχουν σοβαρές συνέπειες για την ασφάλεια του συστήματος.

Εάν το μοντέλο αποτύχει να εντοπίσει μια πραγματική εισβολή στο δίκτυο (FN), μπορεί να επιτρέψει σε κακόβουλες δραστηριότητες να περάσουν απαρατήρητες. Μια μη εντοπισμένη εισβολή θα μπορούσε να οδηγήσει σε μη εξουσιοδοτημένη πρόσβαση, παραβιάσεις δεδομένων και παραβίαση ευαίσθητων πληροφοριών, θέτοντας σημαντικούς κινδύνους για την ασφάλεια του οργανισμού.

Εάν το μοντέλο επισημαίνει λανθασμένα την κανονική κίνηση δικτύου ως εισβολή (ψευδώς θετικό), μπορεί να προκαλέσει περιττές ειδοποιήσεις ή έρευνες. Έτσι, ενώ τα ψευδώς θετικά μπορεί να προκαλέσουν κάποια ταλαιπωρία και να απαιτήσουν πρόσθετο έλεγχο, είναι γενικά λιγότερο κρίσιμα από μια πραγματική εισβολή που μένει απαρατήρητη. [18]

3.1.4 Μετρική F1 (F1 score)

Η μετρική F1 χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου ταξινόμησης. Συνδυάζει την ευστοχία και την ανάκληση σε μια ενιαία τιμή, παρέχοντας μια ισορροπία μεταξύ των δύο μετρικών. Είναι ιδιαίτερα χρήσιμη όταν πρόκειται για μη ισορροπημένα σύνολα δεδομένων όπου ο αριθμός των περιπτώσεων σε διαφορετικές κλάσεις ποικίλλει σημαντικά.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Το F1 score κυμαίνεται από 0 έως 1, όπου μια υψηλότερη τιμή υποδηλώνει καλύτερη απόδοση του μοντέλου. Φτάνει στην καλύτερη τιμή του στο 1 και στη χειρότερη στο 0. Το F1 score δίνει ίση βαρύτητα στην ευστοχία και στην ανάκληση, καθιστώντας την χρήσιμη για καταστάσεις όπου τόσο τα ψευδώς θετικά όσο και τα ψευδώς αρνητικά είναι εξίσου σημαντικά. Συνοπτικά, είναι μια χρήσιμη μετρική για την αξιολόγηση της συνολικής απόδοσης ενός μοντέλου ταξινόμησης, ειδικά σε σενάρια όπου η ευστοχία και η ανάκληση

πρέπει να εξισορροπηθούν. Παρέχει μια ενιαία εκτίμηση της ικανότητας του μοντέλου να ταξινομεί σωστά τις περιπτώσεις σε διαφορετικές κλάσεις. [19]

3.1.5 Υποστήριξη (Support)

Σε ένα πρόβλημα ταξινόμησης, η υποστήριξη για μια κλάση είναι απλώς ο αριθμός των περιπτώσεων που ανήκουν σε αυτήν την κλάση. Παρέχει πολύτιμες πληροφορίες σχετικά με την κατανομή των κλάσεων στο σύνολο δεδομένων και μπορεί να βοηθήσει στην αξιολόγηση της ανισορροπίας ή της ισορροπίας μεταξύ διαφορετικών κλάσεων.

Συνοπτικά, η υποστήριξη είναι μια απλή αλλά ενημερωτική μετρική που παρέχει πληροφορίες σχετικά με την κατανομή των κλάσεων σε ένα σύνολο δεδομένων, βοηθώντας στην κατανόηση της επικράτησης διαφορετικών κατηγοριών και στην αξιολόγηση της ισορροπίας ή της ανισορροπίας του συνόλου δεδομένων.

3.1.6 Πίνακας σύγχυσης (Confusion matrix)

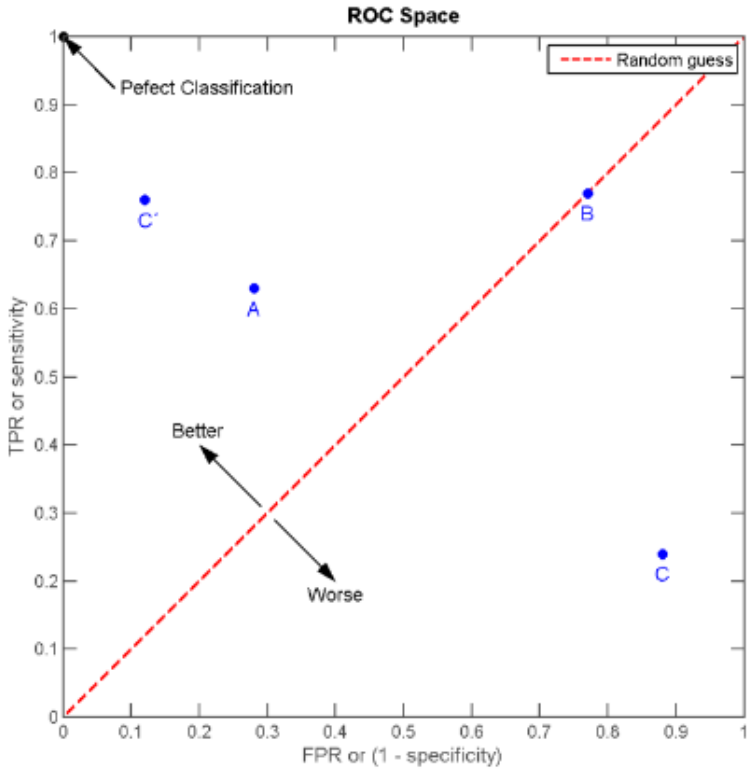
Ένας πίνακας σύγχυσης παρέχει μια λεπτομερή ανάλυση της απόδοσης ενός ταξινομητή δείχνοντας τον αριθμό των σωστά και λανθασμένα θετικών και αρνητικών προβλέψεων. Σε δυαδική ταξινόμηση, είναι ένας τετράγωνος πίνακας όπου κάθε γραμμή αντιπροσωπεύει την αληθινή κλάση και κάθε στήλη αντιπροσωπεύει την προβλεπόμενη κλάση.

Ένας πίνακας σύγχυσης δίνει πληροφορίες για το ποιες κατηγορίες ταξινομούνται εσφαλμένα και σε τι ποσότητες. Χρησιμεύει ως βάση για τον υπολογισμό διαφόρων μετρήσεων αξιολόγησης όπως η ακρίβεια, η ευστοχία, η ανάκληση και η F1 score.

3.1.7 Καμπύλη ROC

Η καμπύλη Λειτουργικού Χαρακτηριστικού Δέκτη (ROC) είναι μια γραφική αναπαράσταση που απεικονίζει τη διαγνωστική ικανότητα ενός μοντέλου ταξινόμησης. Συγκεκριμένα, εκφράζει τη σχέση του ποσοστού των αληθώς θετικών (TPR) και ψευδώς θετικών (FPR) αποτελεσμάτων καθώς μεταβάλλεται προοδευτικά προς μία κατεύθυνση το διαχωριστικό όριο.

Ένας ιδανικός ταξινομητής έχει μια καμπύλη ROC που διέρχεται από την επάνω αριστερή γωνία (0,1) της γραφικής παράστασης, που αντιπροσωπεύει 100% ευαισθησία (TPR) και 0% ψευδώς θετικό ποσοστό (FPR). Ένας τυχαίος ταξινομητής, από την άλλη πλευρά, έχει μια καμπύλη ROC που είναι κοντά στη διαγώνια γραμμή (διακεκομμένη κόκκινη γραμμή), που δεν δείχνει καλύτερη προγνωστική ικανότητα από την τυχαία πιθανότητα. Οποιοσδήποτε ταξινομητής βρίσκεται στο άνω τρίγωνο, δηλαδή αριστερά της ευθείας λέμε ότι έχει καλύτερη από την τυχαία απόδοση, ενώ αυτοί που είναι στο κάτω τρίγωνο λέμε ότι έχουν απόδοση χειρότερη από την τυχαία. Η καμπύλη ROC είναι ένα πολύτιμο εργαλείο για την αξιολόγηση της απόδοσης των μοντέλων ταξινόμησης, ειδικά όταν πρόκειται για μη ισορροπημένα σύνολα δεδομένων ή όταν είναι σημαντικό να εξισορροπηθεί η αντιστάθμιση μεταξύ ευαισθησίας και ειδικότητας. [20]



Σχήμα 3.1:Χώρος ROC με 4 παραδείγματα προβλέψεων

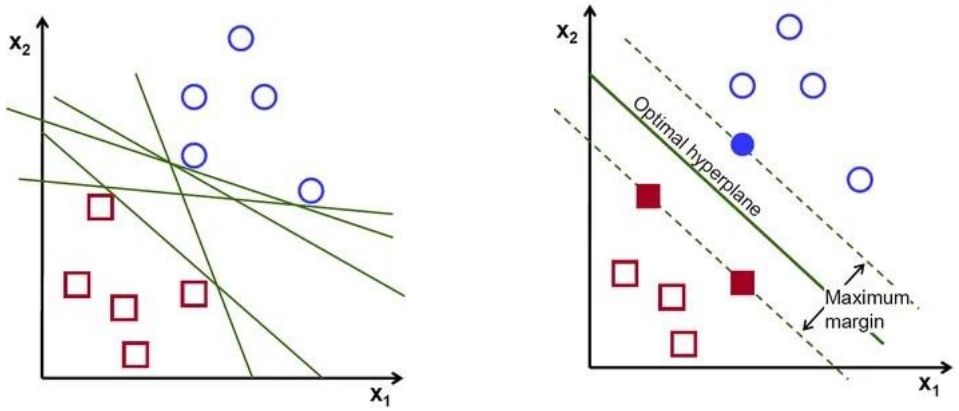
Η AUC, είναι η περιοχή κάτω από την καμπύλη ROC, που ποσοτικοποιεί τη συνολική απόδοση του μοντέλου ταξινόμησης. Η AUC υποδεικνύει την ικανότητα του ταξινομητή να διακρίνει μεταξύ των θετικών και αρνητικών κλάσεων. Μια υψηλότερη τιμή AUC (πιο κοντά στο 1) υποδηλώνει καλύτερη ικανότητα διάκρισης, που σημαίνει ότι ο ταξινομητής είναι πιο αποτελεσματικός στη σωστή κατάταξη των θετικών περιπτώσεων υψηλότερα από τις αρνητικές. Ένας τυχαίος ταξινομητής, που κάνει προβλέψεις τυχαία, θα έχει τιμή AUC κοντά στο 0,5.

3.2 Αλγόριθμοι Μηχανικής Μάθησης

3.2.1 Support Vector Machines

Support Vector Machines (SVMs) είναι ένας τύπος εποπτευόμενου αλγόριθμου μηχανικής μάθησης που μπορεί να χρησιμοποιηθεί τόσο για προβλήματα ταξινόμησης όσο και παλινδρόμησης. Ο πρωταρχικός στόχος του SVM είναι να βρει ένα υπερεπίπεδο σε έναν N-διάστατο χώρο (όπου N είναι ο αριθμός των χαρακτηριστικών) που ταξινομεί ευδιάκριτα τα σημεία δεδομένων σε διαφορετικές κλάσεις. [21] Τα υπερεπίπεδα είναι όρια απόφασης που βοηθούν στην ταξινόμηση των σημείων δεδομένων. Για παράδειγμα, σε πρόβλημα ταξινόμησης κειμένων, ο αλγόριθμος SVM κατηγοριοποιεί τα κείμενα προσδιορίζοντας το καλύτερο υπερεπίπεδο ή οριακή γραμμή που χωρίζει τα δεδομένα κειμένου σε προκαθορισμένες κλάσεις. [22]

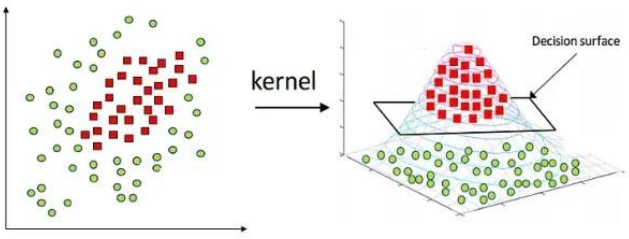
Η διάσταση του υπερεπιπέδου εξαρτάται από τον αριθμό των χαρακτηριστικών. Για 2 χαρακτηριστικά εισόδου, το υπερεπίπεδο είναι μια γραμμή που χωρίζει τα δεδομένα σε δύο κατηγορίες (δισδιάστατος χώρος). Εάν ο αριθμός των χαρακτηριστικών εισόδου είναι 3, τότε το υπερεπίπεδο γίνεται σε τρισδιάστατο επίπεδο. [23]



Σχήμα 3.2: Πιθανά υπερεπίπεδα

Γενικά, ο αλγόριθμος SVM δημιουργεί πολλαπλά υπερεπίπεδα, αλλά ο στόχος είναι να βρεθεί το βέλτιστο υπερεπίπεδο που χωρίζει με ακρίβεια και τις δύο κατηγορίες. Το καλύτερο υπερεπίπεδο είναι αυτό με τη μέγιστη απόσταση από σημεία δεδομένων και των δύο κατηγοριών. Τα διανύσματα ή τα σημεία δεδομένων πιο κοντά στο υπερεπίπεδο ονομάζονται **διανύσματα υποστήριξης**, τα οποία επηρεάζουν σε μεγάλο βαθμό τη θέση και την απόσταση του βέλτιστου υπερεπιπέδου. Χρησιμοποιώντας αυτά τα διανύσματα υποστήριξης, μεγιστοποιούμε το περιθώριο του ταξινομητή. Αυτά είναι τα σημεία που βοηθούν στο χτίσιμο του SVM. Η μεγιστοποίηση της απόστασης περιθωρίου παρέχει κάποια ασφάλεια, έτσι ώστε τα μελλοντικά σημεία δεδομένων να μπορούν να ταξινομηθούν με μεγαλύτερη σιγουριά. [24]

Για παράδειγμα, χρησιμοποιώντας το SVM, μπορείτε να δημιουργήσετε έναν ταξινομητή που θα κατατάσσει τις προτάσεις σε δυο κατηγορίες: ρητορική μίσους (μπλε κύκλος) και ουδέτερη ομιλία (κόκκινο τετράγωνο), όπως φαίνεται στο σχήμα 3.2. Στο αριστερό μέρος φαίνονται όλα τα πιθανά υπερεπίπεδα που χωρίζουν τις δύο κατηγορίες δεδομένων, και στο δεξί το βέλτιστο και καλύτερο υπερεπίπεδο που ταξινομεί τη ρητορική μίσους και την ουδέτερη ομιλία με την υψηλότερη απόσταση ή το μέγιστο περιθώριο από τα σημεία δεδομένων. Επιλέγοντας το καλύτερο δυνατό υπερεπίπεδο, το μοντέλο SVM εκπαιδεύεται στην ταξινόμηση της ομιλίας. Τώρα, κάθε φορά που το νέο σύνολο δεδομένων διαβιβάζεται μέσω από αυτό το μοντέλο μηχανικής εκμάθησης, αντιστοιχίζεται το νέο σύνολο δεδομένων με το προηγουμένως εκπαιδευμένο σύνολο δεδομένων και βάσει αυτού, μπορεί σαφώς να ταξινομηθεί σε ομιλία μίσους ή ουδέτερη.



Σχήμα 3.3: Χρήση του Kernel για τον διαχωρισμό δεδομένων

Ο αρχικός αλγόριθμος SVM έχει σχεδιαστεί για γραμμικά διαχωρίσιμα δεδομένα, όπου μια ευθεία γραμμή (ή υπερεπίπεδο σε υψηλότερες διαστάσεις) μπορεί να χρησιμοποιηθεί για τον διαχωρισμό των δεδομένων σε κλάσεις. Ωστόσο, πολλά σύνολα δεδομένων (datasets) στην πραγματικότητα δεν είναι γραμμικά διαχωρίσιμα. Για να μπορέσει το SVM να διαχειριστεί τέτοιες περιπτώσεις, εισάγει την έννοια των πυρήνων (kernel). Ο πυρήνας είναι η μαθηματική συνάρτηση, η οποία χρησιμοποιείται στο SVM για να αντιστοιχίσει τα αρχικά σημεία δεδομένων σε ένα νέο χώρο χαρακτηριστικών υψηλότερης διάστασης όπου μπορεί να επιτευχθεί ο γραμμικός διαχωρισμός τους βρίσκοντας το βέλτιστο υπερεπίπεδο. Μερικές από τις κοινές συναρτήσεις του πυρήνα είναι η γραμμική, πολυωνυμική, Radial Basis Function(RBF) και σιγμοειδής.

Γραμμικό Kernel (No Kernel)

$$K(x, y) = x^T \cdot y$$

Είναι το τυπικό εσωτερικό γινόμενο των χαρακτηριστικών εισόδου όταν τα δεδομένα είναι ήδη γραμμικά διαχωρίσιμα.

Πολυωνυμικό Kernel

$$K(x, y) = (x^T \cdot y + c)^d$$

Αυτός ο πυρήνας εισάγει πολυωνυμικούς όρους του βαθμού d για να συλλάβει μη γραμμικές σχέσεις. Η παράμετρος c είναι σταθερά.

Radial Basis Function (RBF) ή Gaussian Kernel

$$\boldsymbol{K(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)}$$

Ο πυρήνας RBF εισάγει ένα μέτρο ομοιότητας που βασίζεται στην Γκαουσιανή κατανομή. Ο εκθετικός όρος διασφαλίζει ότι η ομοιότητα μειώνεται όσο αυξάνεται η Ευκλείδεια απόσταση μεταξύ των x και y. Η παράμετρος σ ελέγχει το πλάτος της Γκαουσιανή κατανομής, ένα μικρότερο σ οδηγεί σε μια πιο έντονη και πιο εντοπισμένη κατανομή, ενώ ένα μεγαλύτερο σ οδηγεί σε μια ευρύτερη κατανομή. Ο πυρήνας RBF χρησιμοποιείται συνήθως για μη γραμμικά προβλήματα και είναι πολύ αποτελεσματικός σε προβλήματα ταξινόμησης.

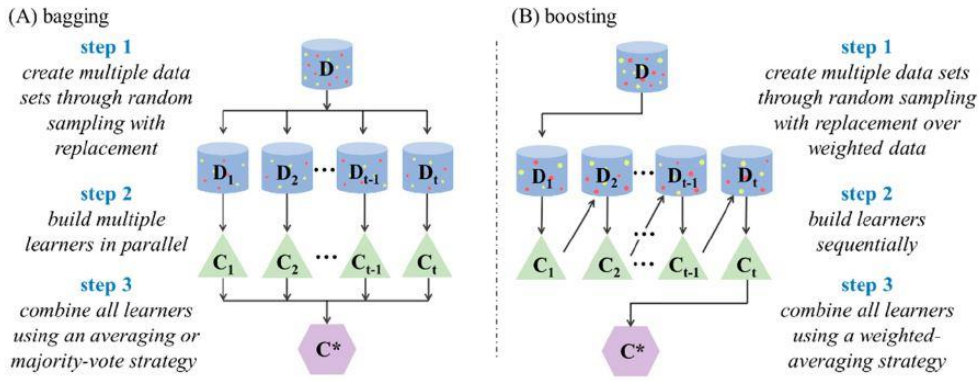
Σιγμοειδής Kernel

$$\boldsymbol{K(x, y) = \tanh(ax^T \cdot y + c)}$$

Όπου, η παράμετρος α ελέγχει την κλίση της σιγμοειδούς συνάρτησης και παίζει ρόλο στον καθορισμό του σχήματος του ορίου απόφασης. Συγκεκριμένα, καθώς η παράμετρος α αυξάνεται, η σιγμοειδής συνάρτηση γίνεται πιο απότομη, οδηγώντας σε μια πιο απότομη μετάβαση μεταξύ των διαφορετικών κλάσεων στο όριο απόφασης. Η παράμετρος c είναι μια σταθερά που μπορεί να ρυθμιστεί για να μετατοπίσει το κέντρο της σιγμοειδούς συνάρτησης. Οι παράμετροι α και c ορίζονται από τον χρήστη. [25]

3.2.2 Bagging and Boosting Algorithms

Εκμάθηση συνόλου είναι μια τεχνική μηχανικής μάθησης όπου συνδυάζονται πολλά βασικά μοντέλα προκειμένου να παραχθεί ένα βέλτιστο μοντέλο πρόβλεψης. Η ιδέα είναι να αξιοποιηθεί η ποικιλομορφία μεταξύ των μοντέλων για να σχηματιστεί ένα ισχυρότερο, πιο στιβαρό μοντέλο από οποιοδήποτε μεμονωμένο μοντέλο στο σύνολο. Βασίζεται στη λογική ότι πολλοί ταξινομητές προβλέπουν καλύτεροι από έναν ταξινομητή. Οι μέθοδοι συνόλου είναι ιδιαίτερα αποτελεσματικές όταν πρόκειται για πολύπλοκα, θορυβώδη ή αβέβαια σύνολα δεδομένων. Ωστόσο, είναι σημαντικό να σημειωθεί ότι οι μέθοδοι εκμάθησης συνόλου δεν αφορούν μόνο τα Δέντρα Αποφάσεων όπου χρησιμοποιούνται συνήθως αλλά μπορούν να εφαρμοστούν σε διάφορους τύπους βασικών μοντέλων (Γραμμική Παλινδρόμηση, Λογιστική Παλινδρόμηση, SVM κ.α.).



Σχήμα 3.4: Αναπαράσταση αλγορίθμων εκμάθησης συνόλου (bagging & boosting)

Το **Bagging (Bootstrap Aggregating)** είναι μια τεχνική εκμάθησης συνόλου που στοχεύει στη βελτίωση της απόδοσης και τη σταθερότητα των μοντέλων μηχανικής εκμάθησης συνδυάζοντας τις προβλέψεις πολλαπλών βασικών μοντέλων. Η κύρια ιδέα πίσω από το bagging είναι να μειωθεί η υπερπροσαρμογή και να αυξηθεί η γενίκευση εκπαιδεύοντας κάθε βασικό μοντέλο σε ένα διαφορετικό υποσύνολο των δεδομένων εκπαίδευσης.

Πιο αναλυτικά, το Bagging ξεκινά με τη δημιουργία πολλαπλών υποσυνόλων (bags) του συνόλου δεδομένων εκπαίδευσης μέσω μιας διαδικασίας γνωστής ως **δειγματοληψία εκκίνησης (bootstrap sampling)**. Επιλέγονται τυχαία στιγμιότυπα από το αρχικό σύνολο δεδομένων με αντικατάσταση για να σχηματιστεί κάθε υποσύνολο. Αυτό σημαίνει ότι ορισμένα στιγμιότυπα μπορεί να επαναληφθούν, ενώ άλλα μπορεί να εξαιρεθούν από κάθε υποσύνολο. Για κάθε υποσύνολο το βασικό μοντέλο εκπαιδεύεται χωριστά, επιτρέποντας την παραλληλοποίηση. Το βασικό μοντέλο μπορεί να είναι οποιοσδήποτε αλγόριθμος μηχανικής μάθησης, όπως δέντρα αποφάσεων, μηχανές υποστήριξης διανυσμάτων ή νευρωνικά δίκτυα.

Για προβλήματα ταξινόμησης, οι προβλέψεις από μεμονωμένα μοντέλα συνδυάζονται χρησιμοποιώντας την ψηφοφορία πλειοψηφίας. Για προβλήματα παλινδρόμησης, οι προβλέψεις υπολογίζονται συνήθως χρησιμοποιώντας τον μέσο όρο προβλέψεων όλων των βασικών μοντέλων.

Άλλη μια τεχνική εκμάθησης συνόλου είναι το **Boosting** που στοχεύει στη βελτίωση της απόδοσης των μοντέλων μηχανικής μάθησης συνδυάζοντας τις προβλέψεις

"αδύναμων/απλών" μοντέλων με διαδοχικό τρόπο, δίνοντας μεγαλύτερη έμφαση σε περιπτώσεις που είχαν ταξινομηθεί εσφαλμένα. Οι προβλέψεις αυτές βελτιώνονται με τις επαναλήψεις και δημιουργείται ένα ισχυρό μοντέλο συνόλου. [26]

3.2.3 Random Forest

Τα **δάση τυχαίας απόφασης (Random Forest)** είναι μια μέθοδος εκμάθησης συνόλου που λειτουργεί με την κατασκευή πολλών δέντρων αποφάσεων κατά το χρόνο εκπαίδευσης και εφαρμόζεται σε προβλήματα ταξινόμησης, παλινδρόμησης και σε άλλες εργασίες. Αποτελεί μία τροποποίηση του bagging αλγορίθμου που βελτιώνει περαιτέρω την απόδοση του μοντέλου εισάγοντας και την τυχαιότητα στη διαδικασία επιλογής χαρακτηριστικών.

Στο Random Forest, δημιουργούμε πολλαπλά δέντρα αποφάσεων χρησιμοποιώντας ένα υποσύνολο των αρχικών χαρακτηριστικών. Σε κάθε κόμβο του δέντρου, αντί να χρησιμοποιούμε όλα τα χαρακτηριστικά, επιλέγουμε τυχαία ένα υποσύνολο χαρακτηριστικών για να χωρίσουμε τα δεδομένα. Αυτή η διαδικασία επαναλαμβάνεται για κάθε κόμβο, με αποτέλεσμα να δημιουργείται ένα δέντρο αποφάσεων που χρησιμοποιεί ένα υποσύνολο χαρακτηριστικών. Με τον παραπάνω τρόπο, το Random Forest εισάγει ποικιλομορφία στα δέντρα απόφασης, γεγονός που μειώνει περαιτέρω τη διακύμανση του μοντέλου. Επιπλέον, η διαδικασία επιλογής χαρακτηριστικών αποτρέπει από την δημιουργία δέντρων υψηλής συσχέτισης, δηλαδή από την ύπαρξη ομοιότητας ή αλληλεξάρτησης μεταξύ των μεμονωμένων δέντρων απόφασης εντός του συνόλου, κάτι που αποτελεί πρόβλημα στην εκμάθηση συνόλου με την τεχνική bagging. Επομένως, το Random Forest μπορεί να βελτιώσει την ακρίβεια ενός μοντέλου μειώνοντας την υπερπροσαρμογή και αυξάνοντας την ποικιλομορφία των δέντρων.

Για κάθε δέντρο, υπάρχουν στιγμιότυπα, που αναφέρονται ως **out-of-bag (oob)**, από το αρχικό σύνολο δεδομένων που δεν συμπεριλαμβάνονται λόγω της δειγματοληψίας με αντικατάσταση (bootstrap sampling). Το σφάλμα **out-of-bag (oob)** είναι μια χρήσιμη έννοια στα Random Forests και παρέχει μια εσωτερική εκτίμηση της απόδοσης του μοντέλου χωρίς να απαιτείται ξεχωριστό σύνολο επικύρωσης (validation set).

Κάθε στιγμιότυπο που μένει έξω κατά τη διάρκεια της εκπαίδευσης ενός συγκεκριμένου δέντρου (out-of-bag στιγμιότυπο για αυτό το δέντρο) μπορεί να χρησιμοποιηθεί για την αξιολόγηση της απόδοσης του δέντρου. Οι out-of-bag προβλέψεις που αφορούν ένα συγκεκριμένο στιγμιότυπο λαμβάνονται εάν αθροίσουμε τις προβλέψεις από όλα τα δέντρα που δεν συμπεριέλαβαν αυτό το στιγμιότυπο στο σετ εκπαίδευσής τους. Το σφάλμα oob υπολογίζεται συγκρίνοντας τις προβλέψεις out-of-bag με τις πραγματικές προβλέψεις των out-of-bag στιγμιότυπων. Στα προβλήματα ταξινόμησης, το σφάλμα είναι το ποσοστό των στιγμιότυπων τα οποία κατά πλειοψηφία έχουν ταξινομηθεί σε λάθος κλάση. Στα προβλήματα παλινδρόμησης, το σφάλμα είναι συνήθως η μέση τετραγωνική διαφορά μεταξύ των προβλεπόμενων και των πραγματικών τιμών.

3.2.4 XGBoost (eXtreme Gradient Boosting)

Το **XGBoost** είναι μία βελτιστοποιημένη υλοποίηση του αλγορίθμου Gradient Boosting που παρουσιάστηκε από τους Chen & Guestrin, (2016). Αποτελεί παράδειγμα εποπτευόμενου μοντέλου μηχανικής μάθησης, όπου χρησιμοποιούμε τα δεδομένα εκπαίδευσης (με πολλαπλά χαρακτηριστικά) για να προβλέψουμε τη μεταβλητή στόχο y_i . Για να εκπαιδεύσουμε το μοντέλο, πρέπει να ορίσουμε την **αντικειμενική συνάρτηση (objective function)** η οποία

αποτελεί το μέτρο της καλής απόδοσης του μοντέλου. Στόχος της εκπαίδευσης του μοντέλου είναι να βρει τις καλύτερες παραμέτρους θ που ταιριάζουν στα δεδομένα εκπαίδευσης x_i και ετικέτες y_i.

Ένα σημαντικό χαρακτηριστικό των αντικειμενικών συναρτήσεων είναι ότι αποτελούνται από δύο μέρη: κόστος εκπαίδευσης (training loss) και regularization term:

$$obj(\theta) = L(\theta) + \Omega(\theta)$$

Όπου $\boldsymbol{L}$ είναι η συνάρτηση του κόστους εκπαίδευσης και $\boldsymbol{\Omega}$ είναι regularization term. Η training loss μετρά πόσο καλά προβλέπει το μοντέλο μας σε σχέση με τα δεδομένα εκπαίδευσης. Μια συχνή επιλογή είναι το μέσο τετραγωνικό σφάλμα, το οποίο δίνεται από:

$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2$$

όπου σε ένα γραμμικό μοντέλο οι προβλέψεις δίνονται από τη σχέση $\hat{y}_i = \sum_i \theta_j x_{ij}$, και αποτελούν γραμμικό συνδυασμό των χαρακτηριστικών εισόδου.

Μια άλλη συνάρτηση κόστους που χρησιμοποιείται συχνά είναι η log loss:

$$L(\theta) = \sum_i [y_i ln(1 + e^{-\hat{y}_i}) + (1 - y_i)\ln(1 + e^{\hat{y}_i})]$$

Η τεχνική regularization που χρησιμοποιείται στο XGBoost, εφαρμόζεται για να ελέγξει την πολυπλοκότητα του μοντέλου. Με τον τρόπο αυτό αποφεύγεται η υπερπροσαρμογή και βελτιώνεται η δυνατότητα του μοντέλου να γενικεύει. Στον αλγόριθμο XGBoost χρησιμοποιούνται δύο τύποι regularization: L1(Lasso) και L2(Ridge). [27], [28]

L1 Regularization (Lasso)

- Προσθέτει το άθροισμα των απόλυτων τιμών των βαρών (ποινή) στην αντικειμενική συνάρτηση.

$$Objective = Loss + \lambda \sum_{j=1}^{J} |w_j|$$

- Καθώς η διαδικασία βελτιστοποίησης επιδιώκει να ελαχιστοποιήσει την αντικειμενική συνάρτηση, ο όρος L1 ενθαρρύνει τον αλγόριθμο να βρει μια λύση μηδενίζοντας ορισμένα βάρη.
- Ορισμένα χαρακτηριστικά μπορεί να αγνοηθούν, καθώς τα σχετικά βάρη ορίζονται στο μηδέν.
- Αυτή η δυνατότητα επιλογής χαρακτηριστικών μπορεί να είναι επωφελής, ειδικά όταν πρόκειται για υψηλών διαστάσεων σύνολα δεδομένων όπου δεν είναι απαραίτητα όλα τα χαρακτηριστικά στις προβλέψεις του μοντέλου

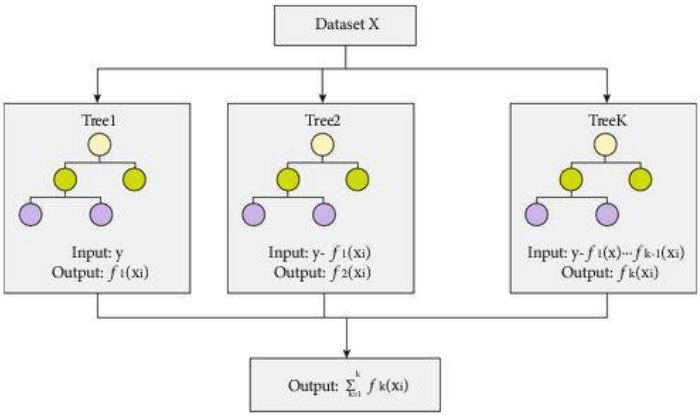
L2 Regularization (Ridge)

- Προσθέτει το άθροισμα των τετραγωνικών τιμών των βαρών στην αντικειμενική συνάρτηση.

$$Objective = Loss + \lambda \sum_{j=1}^{J} w_j^2$$

Όπου, w_j δηλώνουν τα βάρη του μοντέλου, και λ την ισχύ του regularization.

- Καθώς η διαδικασία βελτιστοποίησης επιδιώκει να ελαχιστοποιήσει την αντικειμενική συνάρτηση, ο όρος L2 ενθαρρύνει τον αλγόριθμο να βρει μια λύση όπου τα βάρη διατηρούνται μικρά, εμποδίζοντάς τα να γίνουν πολύ ακραία.
- Βοηθά στην αποφυγή της υπερπροσαρμογής προσθέτοντας έναν όρο τακτοποίησης που ενθαρρύνει τα βάρη να απλώνονται πιο ομοιόμορφα στα χαρακτηριστικά αντί να έχουν μερικά χαρακτηριστικά με πολύ μεγάλα βάρη. [28]



Σχήμα 3.5: Διάγραμμα ροής του XGBoost

3.3 Word embeddings

Τα τελευταία χρόνια, ο τομέας της επεξεργασίας φυσικής γλώσσας (NLP) έχει γνωρίσει αξιοσημείωτη πρόοδο, φέρνοντας επανάσταση στον τρόπο με τον οποίο οι μηχανές κατανοούν και επεξεργάζονται την ανθρώπινη γλώσσα. Μεταξύ των πολλών καινοτομιών στο NLP, οι ενσωματώσεις λέξεων έχουν αναδειχθεί ως μια ισχυρή τεχνική για την αναπαράσταση λέξεων σε έναν συνεχή διανυσματικό χώρο που χρησιμοποιούνται στη συνέχεια ως χαρακτηριστικά (features) για την εκπαίδευση των μοντέλων μηχανικής μάθησης. Αυτές οι διανυσματικές αναπαραστάσεις καταγράφουν σημασιολογικές ομοιότητες μεταξύ των λέξεων, επιτρέποντας στους αλγόριθμους να κατανοούν και να επεξεργάζονται καλύτερα το κείμενο της φυσικής γλώσσας.

Οι ενσωματώσεις λέξεων έχουν αποκτήσει μεγάλη δημοτικότητα, ιδιαίτερα σε εργασίες ταξινόμησης κειμένων, όπου ο στόχος είναι να κατηγοριοποιηθούν αυτόματα τα κείμενα σε προκαθορισμένες κλάσεις. Οι παραδοσιακές προσεγγίσεις στην ταξινόμηση κειμένων βασίζονταν σε αραιές, υψηλών διαστάσεων αναπαραστάσεις, όπως το bag-of-word ή το TF-IDF, οι οποίες συχνά δυσκολεύονταν να συλλάβουν τις σημασιολογικές πληροφορίες. υπήρχαν στα δεδομένα εισόδου.

3.3.1 Word2Vec

Το Word2Vec είναι είναι μια ευρέως χρησιμοποιούμενη μέθοδος στην επεξεργασία φυσικής γλώσσας μία τεχνική μάθησης χωρίς επίβλεψη, η οποία αναπτύχθηκε από τον Tomas Mikolov και άλλους ερευνητές της Google το 2013. Όπως υποδηλώνει το όνομα του, το Word2vec αντιστοιχίζει κάθε ξεχωριστή λέξη σε ένα διάνυσμα. Τα διανύσματα αυτά

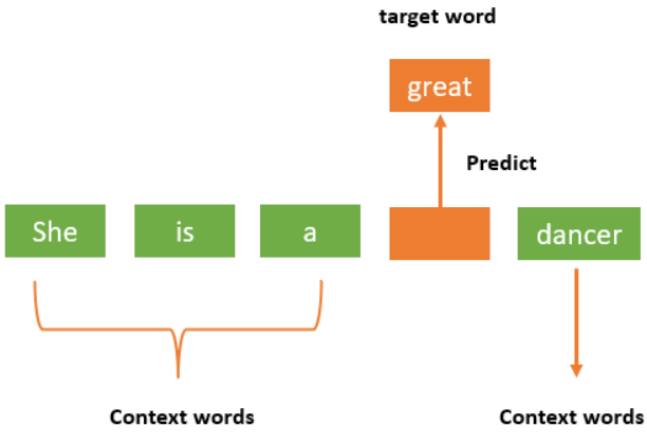
επιλέγονται προσεκτικά έτσι ώστε να αποτυπώνουν τις σημασιολογικές και συντακτικές ιδιότητες των λέξεων. Ως εκ τούτου, μια απλή μαθηματική συνάρτηση (ομοιότητα συνημιτόνου) μπορεί να υποδεικνύει το επίπεδο σημασιολογικής ομοιότητας μεταξύ των λέξεων που αντιπροσωπεύονται από αυτά τα διανύσματα. Λέξεις με παρόμοια σημασία ή χρήση τείνουν να έχουν παρόμοιες διανυσματικές αναπαραστάσεις, που σημαίνει ότι βρίσκονται πιο κοντά μεταξύ τους στο διανυσματικό χώρο ενσωμάτωσης. Αντίθετα, λέξεις με διαφορετική σημασία ή χρήση έχουν ανόμοιες αναπαραστάσεις, με διανύσματα που είναι πιο μακριά μεταξύ τους στον χώρο ενσωμάτωσης.

Το Word2Vec αποτελείται από δύο κύριες αρχιτεκτονικές: το Continuous Bag of Words (CBOW) και το Skip-gram. [29] Ας πούμε λίγα λόγια και για τις δύο αυτές μεθόδους ξεχωριστά για να σχηματίσουμε μια άποψη για τον τρόπο λειτουργίας τους.

3.3.1.1 Continuous Bag of Words (CBOW)

Το CBOW είναι ένας αλγόριθμος που βασίζεται σε ρηχά νευρωνικά δίκτυα και προβλέπει τη «λέξη στόχο» (target word) από το περιεχόμενο, τις λέξεις δηλαδή που βρίσκονται γύρω από την λέξη στόχο, οι οποίες ονομάζονται λέξεις περιβάλλοντος (context words). Μαθαίνει να προβλέπει τη λέξη-στόχο με βάση τις λέξεις που εμφανίζονται πριν και μετά από αυτήν σε ένα δεδομένο παράθυρο λέξεων περιβάλλοντος.

Το μέγεθος του παραθύρου περιβάλλοντος, είναι μια υπερπαράμετρος που καθορίζεται από



Σχήμα 3.6: Παράδειγμα του CBOW μοντέλου

τον χρήστη. Δηλώνει το εύρος των λέξεων που θα συμπεριληφθούν ως το περιεχόμενο της λέξης - στόχου. Για παράδειγμα, ένα μέγεθος παραθύρου 2 παίρνει δύο λέξεις πριν και δύο μετά από τη λέξη – στόχος ως το περιεχόμενο της εκπαίδευσης. Το μέγεθος του παραθύρου είναι μια σημαντική υπερπαράμετρος στα μοντέλα εκμάθησης ενσωματώσεων, επειδή ελέγχει τον αριθμό των λέξεων που πρέπει να θεωρηθούν ως το περιεχόμενο που καθορίζει την αναπαράσταση μιας συγκεκριμένης λέξης. Ενδέχεται να απαιτείται ένα ευρύτερο παράθυρο όταν εκπαιδεύεται ένα μοντέλο σε κείμενο που είναι γεμάτο προτάσεις που περιέχουν σύνθετες δομές (π.χ. βιοϊατρική βιβλιογραφία).

Όταν χρησιμοποιείται μεγαλύτερο μέγεθος παραθύρου, το μοντέλο λαμβάνει υπόψη ένα ευρύτερο πλαίσιο λέξεων που περιβάλλουν τη λέξη-στόχο. Αυτό επιτρέπει στο μοντέλο να

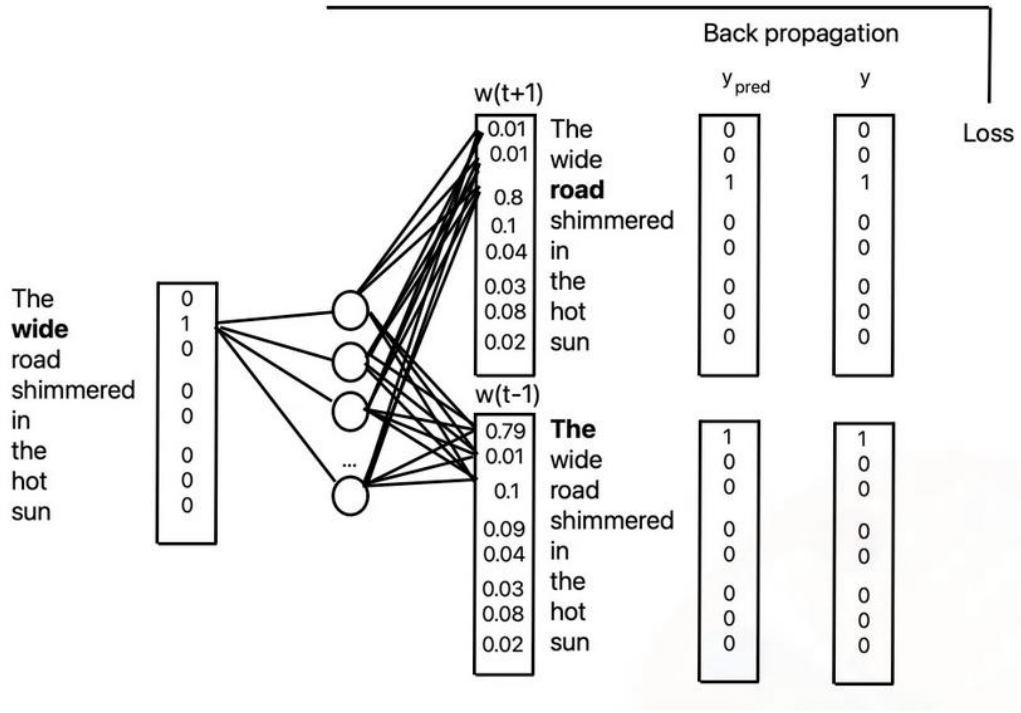
καταγράφει πιο μακρινές σημασιολογικές σχέσεις μεταξύ των λέξεων στο κείμενο. Συγκεκριμένα, ένα μεγαλύτερο μέγεθος παραθύρου τείνει να τονίζει την εκμάθηση της ομοιότητας μεταξύ των λέξεων. Λέξεις που εμφανίζονται στο ίδιο ευρύ πλαίσιο ή περιοχή θέματος είναι πιθανό να έχουν παρόμοιες ενσωματώσεις στον προκύπτον διανυσματικό χώρο.

Αντίθετα, όταν χρησιμοποιείται ένα μικρότερο μέγεθος παραθύρου, το μοντέλο εστιάζει σε ένα μικρότερο πλαίσιο λέξεων που περιβάλλουν αμέσως τη λέξη-στόχο. Αυτό περιορίζει τον όγκο των πληροφοριών που λαμβάνονται υπόψη κατά τη διάρκεια της εκπαίδευσης. Σε αυτή την περίπτωση, το μοντέλο μάθησης καταγράφει κυρίως τις λειτουργίες των λέξεων ή τις συντακτικές σχέσεις μέσα σε προτάσεις. Το μοντέλο δίνει μεγαλύτερη προσοχή στο πώς χρησιμοποιούνται οι λέξεις σε συγκεκριμένα γλωσσικά πλαίσια, όπως στις σχέσεις υποκειμένου-ρήματος-αντικειμένου ή στη γραμματική. Για παράδειγμα, με ένα μικρό μέγεθος παραθύρου, το μοντέλο μπορεί να μάθει ότι η λέξη "τρέχει" εμφανίζεται συχνά μετά τη λέξη "Αυτός/ή" και πριν από τη λέξη "γρήγορα", υποδεικνύοντας τη λειτουργία της ως ρήμα σε προτάσεις όπως "Αυτός τρέχει γρήγορα". Η επιλογή του κατάλληλου μεγέθους παραθύρου εξαρτάται από τους συγκεκριμένους στόχους της εργασίας NLP και τα χαρακτηριστικά του σώματος κειμένου που αναλύεται. [30]

Οι ενσωματώσεις λέξεων παράγονται εκπαιδεύοντας το νευρωνικό δίκτυο ώστε να ελαχιστοποιείται το σφάλμα πρόβλεψης μεταξύ της προβλεπόμενης λέξης στόχου και της πραγματικής λέξης στόχου στα δεδομένα εκπαίδευσης. Το CBOW είναι ένας τύπος μάθησης «χωρίς επίβλεψη», που σημαίνει ότι μπορεί να μάθει από δεδομένα χωρίς ετικέτα (label) και χρησιμοποιείται συχνά για την προ-εκπαίδευση ενσωματώσεων λέξεων που μπορούν να χρησιμοποιηθούν για διάφορες εργασίες NLP, όπως ανάλυση συναισθήματος, ταξινόμηση κειμένου και μηχανική μετάφραση. Το μοντέλο CBOW είναι αποτελεσματικό για εκπαίδευση σε μεγάλα σύνολα δεδομένων και τείνει να αποδίδει καλά στις συχνά χρησιμοποιούμενες λέξεις. [31]

3.3.1.2 Skip-gram

Αντίθετα, στο μοντέλο Skip-gram, το νευρωνικό δίκτυο λαμβάνει ως είσοδο την λέξη - στόχο και προσπαθεί να προβλέψει τις λέξεις του περιβάλλοντος μέσα σε ένα σταθερό μέγεθος παραθύρου. Το μοντέλο εκπαιδεύεται με σκοπό να μεγιστοποιήσει την πιθανότητα παρατήρησης των πλησιέστερων λέξεων δεδομένης της λέξης-στόχου. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο μαθαίνει να προσαρμόζει τις παραμέτρους του (βάρη) μέσω της backpropagation (μέθοδος οπισθοδιάδοσης του λάθους) για να βελτιώσει τις προβλέψεις του για τις λέξεις του περιβάλλοντος. Το Skip-gram χρησιμοποιείται συχνά όταν ο στόχος είναι να δημιουργηθούν ενσωματώσεις λέξεων που καταγράφουν συντακτικές και σημασιολογικές σχέσεις μεταξύ λέξεων με βάση τα μοτίβα συν-εμφάνισής τους στο σώμα εκπαίδευσης.



Σχήμα 3.7:Παράδειγμα λειτουργίας του skip-gram αλγορίθμου

Όπως φαίνεται στο σχήμα 3.7, το skip gram μοντέλο δέχεται για είσοδο τη λέξη “wide” και προσπαθεί να προβλέψει τις λέξεις που την περιβάλλουν. Χρησιμοποιώντας την backpropagation, το νευρωνικό δίκτυο προσαρμόζει τα βάρη μέχρι να ελαχιστοποιηθεί το σφάλμα και ύστερα από αρκετές επαναλήψεις καταλήγει ότι η λέξη που την ακολουθεί είναι η “road”. Το Skip-gram είναι αποτελεσματικό στην καταγραφή σπάνιων λέξεων και είναι πιο ισχυρό όταν εφαρμόζεται σε μικρότερα σύνολα δεδομένων. [32], [33]

3.3.1.3 Περιορισμοί του Μοντέλου

Το Word2Vec είναι μια ισχυρή και ευρέως χρησιμοποιούμενη τεχνική για την εκμάθηση ενσωματώσεων λέξεων. Ωστόσο, το μοντέλο Word2vec έχει ορισμένους περιορισμούς, μερικοί από τους οποίους αναλύονται παρακάτω:

- Το Word2Vec λειτουργεί σε επίπεδο λέξης και δεν καταγράφει πληροφορίες υπο-λέξεων. Αυτό έχει ως αποτέλεσμα, να δυσκολεύεται με τις λέξεις εκτός λεξιλογίου και σπάνιες λέξεις που μπορεί να μην εμφανίζονται συχνά στο σώμα εκπαίδευσης. Ας εξετάσουμε ένα παράδειγμα που περιλαμβάνει τις λέξεις "apple" και "applesauce", όπου το "apple" είναι μια κοινή λέξη και το "applesauce" είναι μια λιγότερο κοινή ή εκτός λεξιλογίου λέξη. Το Word2Vec μοντέλο θα δημιουργήσει μια ισχυρή ενσωμάτωση για τη λέξη"apple" με βάση τη συχνή εμφάνισή της. Από την άλλη πλευρά, το Word2Vec μπορεί να δυσκολεύεται να δημιουργήσει ενσωμάτωση για τη λέξη "applesauce" λόγω της περιορισμένης έκθεσής του στη συγκεκριμένη λέξη κατά τη διάρκεια της εκπαίδευσης. Χωρίς να λαμβάνει υπόψη τις πληροφορίες υπο-λέξεων, το Word2Vec μπορεί να αντιμετωπίζει τη λέξη "applesauce" ως μια εντελώς ξεχωριστή και άσχετη λέξη, αντί να αναγνωρίσει τη σύνδεσή της με τη λέξη "apple" ως παράγωγη ή σύνθετη λέξη.

- Η επιλογή του μεγέθους παραθύρου στο Word2Vec καθορίζει το εύρος των λέξεων περιβάλλοντος που εξετάζονται για κάθε λέξη. Ένα σταθερό μέγεθος παραθύρου μπορεί να είναι περιοριστικό και ενδέχεται να μην αποτυπώνει επαρκώς εξαρτήσεις μεγάλης εμβέλειας ή να μην οδηγεί στην πλήρη κατανόηση του περιεχομένου σε περιπτώσεις όπου χρησιμοποιείται σαρκασμός ή ειρωνεία, σε ιδιωματικές εκφράσεις κ.α.

- Το Word2Vec αντιπροσωπεύει κάθε λέξη με ένα μόνο διάνυσμα, ανεξάρτητα από τις διαφορετικές έννοιές της σε διαφορετικά περιβάλλοντα (πολυσημία). Αυτό μπορεί να οδηγήσει σε προβληματικές αναπαραστάσεις για πολύσημες λέξεις, όπου το μοντέλο αποτυγχάνει να διακρίνει τις διαφορετικές έννοιες της λέξης. Στην περίπτωση της λέξης "apple", το Word2Vec αντιπροσωπεύει τη λέξη με ένα μόνο διάνυσμα, ανεξάρτητα από το αν χρησιμοποιείται για να δηλώσει το φρούτο ή την εταιρεία τεχνολογίας. Αυτό σημαίνει ότι το Word2Vec δεν κάνει διάκριση μεταξύ των διαφορετικών σημασιών της λέξης "apple" σε διαφορετικά περιβάλλοντα και μπορεί να παρέχει τις ίδιες ή παρόμοιες ενσωματώσεις και για τις δύο έννοιες. Αυτός ο περιορισμός προκύπτει από την έλλειψη ευαισθησίας του Word2Vec και την αντιμετώπιση κάθε λέξης ως διακριτής μονάδας, αγνοώντας το πλαίσιο περιβάλλοντος στο οποίο εμφανίζεται η λέξη.

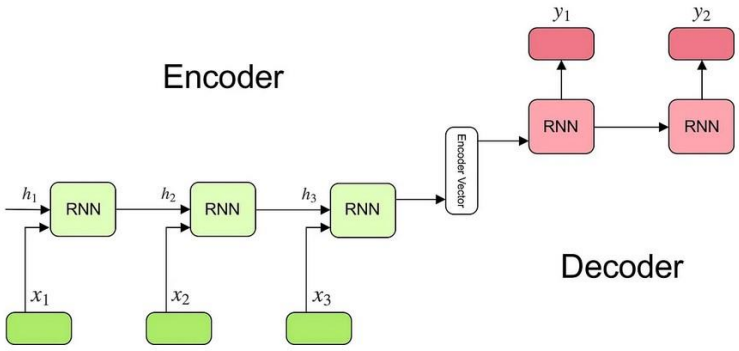
- Το Word2Vec αγνοεί τις μορφολογικές παραλλαγές λέξεων, αντιμετωπίζοντας τις διαφορετικές μορφές μιας λέξης (π.χ. “run”, “runs”, “running”) ως διακριτές οντότητες. Αυτό μπορεί να οδηγήσει σε περιττές αναπαραστάσεις για λέξεις που σχετίζονται μορφολογικά.

Για να αντιμετωπιστούν ορισμένοι από αυτούς τους περιορισμούς, έχουν αναπτυχθεί νεότερες τεχνικές για τη δημιουργία των ενσωματώσεων όπως ενσωματώσεις με βάση τα συμφραζόμενα (π.χ. BERT) που θα αναλυθεί παρακάτω.

3.4 Βαθιά Μάθηση

3.4.1 Αρχιτεκτονική Κωδικοποιητή – Αποκωδικοποιητή

Τα νευρωνικά δίκτυα με αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή (encoder-decoder networks) χρησιμοποιούνται ευρέως σε διάφορες εφαρμογές, συμπεριλαμβανομένης της αυτόματης μετάφρασης, της περίληψης κειμένου, της δημιουργίας λεζάντας εικόνων και άλλων. Χρησιμοποιούνται κυρίως σε εργασίες sequence to sequence και αποτελούν την τεχνολογία αιχμής στο πεδίο της νευρωνικής μηχανικής μάθησης. Αποτελείται από δύο κύρια στοιχεία: έναν κωδικοποιητή και έναν αποκωδικοποιητή.



Σχήμα 3.8: Αρχιτεκτονική του Κωδικοποιητή –Αποκωδικοποιητή σε seq2seq μοντέλο

3.4.1.1 Κωδικοποιητής (Encoder)

- Ο κωδικοποιητής μπορεί να αποτελείται από πολλαπλά αναδρομικά νευρωνικά δίκτυα RNN που στοιβάζονται μαζί. Κάθε δίκτυο RNN δέχεται μια ακολουθία εισόδου $x1$, $x2$, $x3 \in X$ και δημιουργεί μια διανυσματική αναπαράστασή $h1$, $h2$, h3 $\in H$ που ονομάζεται κρυφή κατάσταση(hidden state), η οποία είναι ουσιαστικά η «μνήμη» του δικτύου. Στην ακολουθία εισόδου, τα $x1$, $x2$, $x3 \in X$ θα μπορούσαν να είναι αντίστοιχα οι λέξεις μιας πρότασης που δίνεται για μετάφραση.

- Αφού διαβαστούν όλες οι είσοδοι από το μοντέλο κωδικοποιητή, η τελική κρυφή κατάσταση (hidden state) του μοντέλου αντιπροσωπεύει το νοηματικό πλαίσιο (context) ολόκληρης της ακολουθίας εισόδου.

Παράδειγμα: Θεωρήστε πως η ακολουθία που μας ενδιαφέρει είναι μια πρόταση των 5 λέξεων. Το RNN δίκτυο θα ξεδιπλωθεί σε ένα νευρωνικό δίκτυο 5-στρωμάτων, ένα στρώμα για κάθε λέξη. Θα υπάρχουν συνολικά 5 κόμβοι RNN για το μοντέλο κωδικοποιητή. Κάθε χρονική στιγμή t, διαδοχικά σε κάθε κόμβο, θα ενημερώνεται η κρυφή κατάσταση h λαμβάνοντας υπόψη την προηγούμενη κρυφή κατάσταση h_{t-1} και την τρέχουσα είσοδο x_t.

1. Τη χρονική στιγμή t_1, η προηγούμενη κρυφή κατάσταση h_0 θα θεωρηθεί ως μηδέν ή θα επιλεγεί τυχαία. Έτσι, ο πρώτος κόμβος RNN θα ενημερώσει την τρέχουσα κρυφή κατάσταση με την πρώτη είσοδο και το h_0. Κάθε κόμβος εξάγει δύο πράγματα — την ενημερωμένη κρυφή κατάσταση και την έξοδο για κάθε κόμβο. Οι έξοδοι σε κάθε κόμβο απορρίπτονται και μόνο οι κρυφές καταστάσεις θα διαδοθούν στο επόμενο κόμβο.

2. Η κρυφή κατάσταση h_i υπολογίζεται χρησιμοποιώντας τον παρακάτω τύπο:

$$h_t = f(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$$

όπου

h_t: κρυφή κατάσταση τη χρονική στιγμή t. Αντιπροσωπεύει την εσωτερική μνήμη του δικτύου ή την αναπαράσταση της ακολουθίας εισόδου μέχρι τη χρονική στιγμή t.

f: συνάρτηση ενεργοποίησης εφαρμόζεται στο άθροισμα των σταθμισμένων εισόδων. Η συνάρτηση *f* είναι συνήθως μη γραμμική όπως η tanh ή ReLU.

$W^{(hh)}$: Πίνακας βαρών της προηγούμενης κρυφής κατάστασης h_{t-1}.

$W^{(hx)}$: Πίνακας βαρών για την τρέχουσα είσοδο x_t. Τα βάρη είναι παράμετροι που μαθαίνει το δίκτυο κατά την εκπαίδευση του και αποτυπώνουν τις σχέσεις και τα μοτίβα μεταξύ των λέξεων.

h_{t-1}: Η κρυφή κατάσταση την προηγούμενη χρονική στιγμή t-1.

x_t : Η είσοδος τη χρονική στιγμή t.

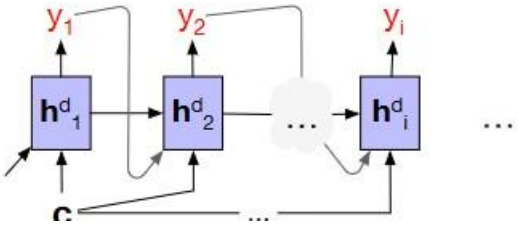
3. Τη χρονική στιγμή t_2, η προηγούμενη κρυφή κατάσταση h_1 και η δεύτερη είσοδος x_2 θα δοθούν ως είσοδοι και η κρυφή κατάσταση h_2 θα ενημερωθεί σύμφωνα με τις δύο εισόδους. Δηλαδή, κάθε κρυφή κατάσταση εξαρτάται όχι μόνο από την τρέχουσα είσοδο αλλά και από την προηγούμενη κρυφή κατάσταση. Αυτό επιτρέπει στον κωδικοποιητή να καταγράφει εξαρτήσεις και πληροφορίες από ολόκληρη την ακολουθία εισόδου καθώς αυτή ξεδιπλώνεται με την πάροδο του χρόνου. Αυτό συμβαίνει και για τα τέσσερα στάδια του παραδείγματος. [34], [35]

3.4.1.2 Διάνυσμα Κωδικοποιητή (Encoder Vector or Context)

Αυτή είναι η **τελική κρυφή κατάσταση** ή αλλιώς **το νοηματικό πλαίσιο** (context) που παράγεται από το τμήμα του κωδικοποιητή του μοντέλου. Αυτό το διάνυσμα προσπαθεί να ενσωματώσει τις πληροφορίες για όλα τα στοιχεία εισόδου, προκειμένου να βοηθήσει τον αποκωδικοποιητή να κάνει τις ακριβείς προβλέψεις. Λειτουργεί ως η αρχική κρυφή κατάσταση του τμήματος αποκωδικοποιητή του μοντέλου. Υπολογίζεται χρησιμοποιώντας τον παραπάνω τύπο.

3.4.1.3 Αποκωδικοποιητής

Ο αποκωδικοποιητή στα δεξιά παίρνει την τελική κρυφή κατάσταση του κωδικοποιητή και τη χρησιμοποιεί για να αρχικοποιήσει την πρώτη κρυφή κατάσταση του αποκωδικοποιητή. Δηλαδή, ο πρώτος κόμβος του αποκωδικοποιητή χρησιμοποιεί το νοηματικό πλαίσιο (context) που εξήχθη



Σχήμα 3.9: Κάθε κρυφή κατάσταση του αποκωδικοποιητή επηρεάζεται από το νοηματικό πλαίσιο που έχει παραχθεί από τον κωδικοποιητή

από τον κωδικοποιητή ως αρχική του κρυφή κατάσταση h^d_0. Η εξίσωση με την οποία ενημερώνεται η κρυφή κατάσταση του αποκωδικοποιητή μπορεί να εκφραστεί ως:

$$h_t = f(y_{t-1}, h_{t-1}, c_t)$$

Ο αποκωδικοποιητής δημιουργεί με διαδοχικό τρόπο μια ακολουθία εξόδων, μία έξοδο σε κάθε βήμα, μέχρι να εμφανιστεί ένας δείκτης τέλους της ακολουθίας. Κάθε κρυφή κατάσταση εξαρτάται από την προηγούμενη κρυφή, την έξοδο που δημιουργήθηκε στο προηγούμενο βήμα καθώς και το νοηματικό πλαίσιο (context) του κωδικοποιητή. [36]

3.4.1.4 Μειονεκτήματα των Encoder-Decoder RNN μοντέλων

Η Encoder-Decoder αρχιτεκτονική RNN μοντέλων που παρουσιάστηκε παραπάνω έχει μερικούς περιορισμούς:

- **Βραχυπρόθεσμη μνήμη**: Τα RNN δυσκολεύονται να συμπεριλάβουν ολόκληρη την πληροφορία, ιδιαίτερα σε περιπτώσεις μεγάλων ακολουθιών εισόδου. Αδυνατούν να καταγράψουν μακροπρόθεσμες συσχετίσεις, δηλαδή το δίκτυο να θυμάται πληροφορίες από τις εισόδους που απέχουν μεταξύ τους αρκετά βήματα. (Bahdanau et al., 2015)

- **Ευαισθησία στο μήκος ακολουθίας**: Η εκπαίδευση ενός RNN μπορεί να είναι δύσκολη, ιδιαίτερα όταν οι ακολουθίες εισόδου και εξόδου είναι διαφορετικού μήκους. Τα RNN είναι ευαίσθητα στο μήκος των ακολουθιών εισόδου. Τα διαφορετικά μήκη μπορούν να οδηγήσουν σε διαφορετικές διαστάσεις κρυφής κατάστασης, καθιστώντας δύσκολη την αποτελεσματική επεξεργασία ακολουθιών μεταβλητού μήκους.

- **Περιορισμένη παράλληλη επεξεργασία**: Τα RNN επεξεργάζονται τις ακολουθίες διαδοχικά, γεγονός που περιορίζει την ικανότητά τους να εκμεταλλεύονται την παράλληλη επεξεργασία. Αυτό μπορεί να οδηγήσει σε πιο αργούς χρόνους εκπαίδευσης, ειδικά για μεγάλες ακολουθίες.

- **Υπερπροσαρμογή**: Τα RNN είναι επιρρεπή σε υπερπροσαρμογή (overfitting), ιδιαίτερα όταν το σύνολο δεδομένων είναι μικρό, κάτι που μπορεί να αποτελέσει πρόβλημα για ορισμένες εργασίες.

- **Έλλειψη Μηχανισμού Προσοχής**: Τα παραδοσιακά RNN δεν ενσωματώνουν εγγενώς μηχανισμούς προσοχής. Οι μηχανισμοί προσοχής είναι ζωτικής σημασίας για την επιλεκτική εστίαση σε διαφορετικά μέρη της ακολουθίας εισόδου κατά τη δημιουργία κάθε στοιχείου στην ακολουθία εξόδου.

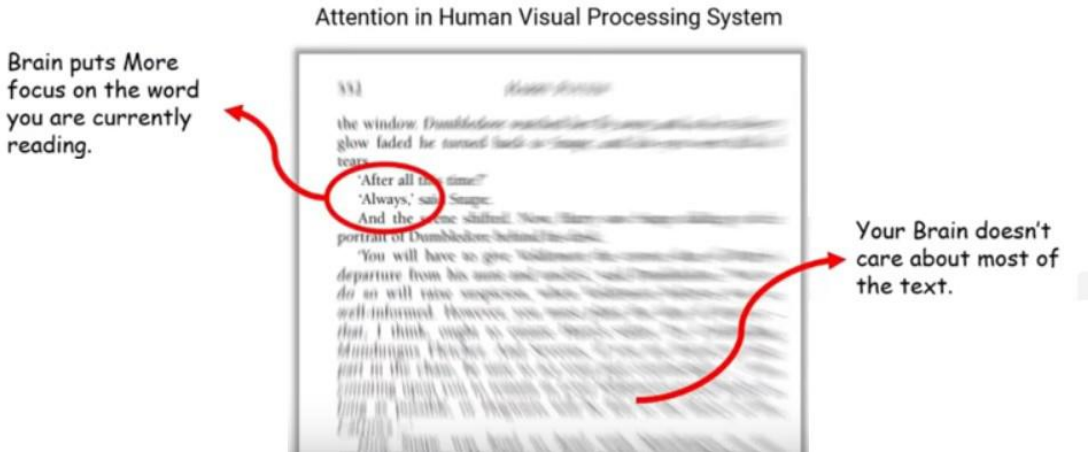
- **Πρόβλημα εξαφάνισης ή έκρηξης κλίσης** (vanishing and exploding gradient): Κατά τη διάρκεια της εκπαίδευσης, τα RNN μπορεί να αντιμετωπίσουν το πρόβλημα εξαφάνισης ή έκρηξης της κλίσης, όπου οι κλίσεις γίνονται εξαιρετικά μικρές ή μεγάλες. Αυτό μπορεί να εμποδίσει τη διαδικασία μάθησης και να επηρεάσει την ικανότητα του μοντέλου να συλλαμβάνει τις συσχετίσεις.

Λόγω αυτών των μειονεκτημάτων, πιο προηγμένες αρχιτεκτονικές όπως LSTM (Long Short-Term Memory), τα GRU (Gated Recurrent Units) και μοντέλα που βασίζονται σε

μετασχηματιστές με μηχανισμούς προσοχής έχουν αποκτήσει δημοτικότητα. Αυτές οι αρχιτεκτονικές αντιμετωπίζουν πολλούς από τους περιορισμούς που σχετίζονται με τα παραδοσιακά RNN και χρησιμοποιούνται ευρέως σε πολλές εργασίες NLP.

3.4.2 Μηχανισμός προσοχής (Attention is all you need)

Ο μηχανισμός προσοχής έχει εμπνευστεί από τον τρόπο που ο ανθρώπινος εγκέφαλος επεξεργάζεται τις πληροφορίες που βλέπει. Έτσι, για παράδειγμα κατά τη διάρκεια ανάγνωσης ενός βιβλίου, ο ανθρώπινος εγκέφαλος αγνοεί την πλειοψηφία των πληροφοριών που βρίσκονται στο οπτικό του πεδίο και επικεντρώνεται (δίνει περισσότερη προσοχή) σε συγκεκριμένα τμήματα του κειμένου. [37] Αυτό επιτρέπει στον εγκέφαλο να εστιάσει επιλεκτικά στις πληροφορίες που έχουν μεγαλύτερη σημασία και να τα διαχειριστεί αποτελεσματικά.



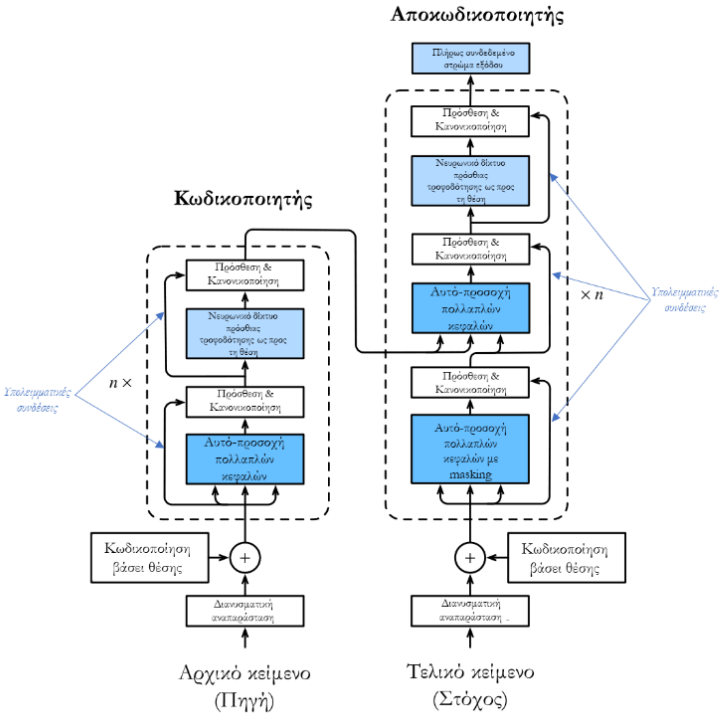
Σχήμα 3.10: Προσοχή όπως εφαρμόζεται από τον ανθρώπινο εγκέφαλο κατά την οπτική επεξεργασία πληροφοριών

Προκειμένου να επιτύχουμε το ίδιο αποτέλεσμα στα μοντέλα βαθιάς μάθησης, εισάγεται η έννοια των βαρών προσοχής σε κάθε ένα από τις εισόδους μας. Με αυτό τον τρόπο καθοδηγούμε το μοντέλο μας να δώσει περισσότερη προσοχή σε συγκεκριμένες εισόδους που είναι κρίσιμης σημασίας για την εκτέλεση της εργασίας μας. Ο μηχανισμός προσοχής στη βαθιά μάθηση χρησιμοποιήθηκε πρώτα από τον Bahdanau για την αυτόματη μετάφραση.

3.4.3 Μετασχηματιστές

Η χρήση ανατροφοδοτούμενων νευρωνικών δικτύων, (recurrent neural networks – RNNs) που προτάθηκε αρχικά ως μοντέλο του κωδικοποιητή, εμφάνισε περιορισμούς (bottlenecks) σε περιπτώσεις μεγάλων ακολουθιών εισόδου, καθώς το διάνυσμα c, που μεταφέρει το νοηματικό πλαίσιο της εισόδου, δε μπορούσε να συμπεριλάβει ολόκληρη την πληροφορία (Bahdanau et al., 2015). Η μετατροπή του c σε διάνυσμα μεταβλητού μήκους σε συνδυασμό με την εισαγωγή μηχανισμών προσοχής (attention mechanisms) έλυσε το παραπάνω πρόβλημα. Το νευρωνικό μοντέλο Transformer (Vaswani et al., 2017) είναι μια αρχιτεκτονική encoder-decoder που αξιοποιεί τους παραπάνω μηχανισμούς και έθεσε τα θεμέλια για τη σύγχρονη εποχή της επεξεργασίας φυσικής γλώσσας, επιτυγχάνοντας σημαντική βελτίωση τόσο στην ποιότητα μετάφρασης, όσο και σε πολλές άλλες NLP εργασίες.

Ένας από τους κύριους λόγους για τους οποίους οι μετασχηματιστές θεωρούνται σημαντικότεροι από τα αναδρομικά νευρωνικά δίκτυα είναι η ικανότητά τους να συλλαμβάνουν αποτελεσματικά το νοηματικό πλαίσιο και σχέσεις μεγάλης εμβέλειας μεταξύ διαφορετικών τμημάτων της εισόδου.



Σχήμα 3.11:Δομή του μετασχηματιστή

Η έννοια του μετασχηματιστή αναφέρθηκε πρώτη φορά στο άρθρο "Attention is All You Need", το οποίο δημοσιεύτηκε από μια ομάδα επιστημόνων της Google Brain. Σ'αυτό το άρθρο επαναπροσδιορίζεται η έννοια του **μηχανισμού προσοχής** (attention mechanism) και εισάγονται άλλες όπως κλειδί (key), ερώτημα (query) και τιμή (value), καθώς και ο **μηχανισμός προσοχής πολλαπλών κεφαλών** (multi head attention). [37]

Ο μετασχηματιστής βασίζεται σε μια ισχυρή αρχιτεκτονική νευρωνικών δικτύων και έχει δομή Κωδικοποιητή-Αποκωδικοποιητή, όπου το καθένα αποτελείται από πολλαπλά πανομοιότυπα στρώματα. Όπως και στις παραπάνω αρχιτεκτονικές RNN μοντέλων, ο κωδικοποιητής επεξεργάζεται την ακολουθία εισόδου και ο αποκωδικοποιητής δημιουργεί την ακολουθία εξόδου.

Ο πυρήνας του μετασχηματιστή είναι ο μηχανισμός αυτοπροσοχής, που επιτρέπει στο μοντέλο να δίνει διαφορετική βαρύτητα στα μέρη της ακολουθίας εισόδου όταν κάνει προβλέψεις. Η προσοχή πολλαπλών κεφαλών περιλαμβάνει την εκτέλεση αυτοπροσοχής πολλές φορές παράλληλα, η καθεμία με το δικό της σύνολο παραμέτρων εκμάθησης.

Δεδομένου ότι ο μετασχηματιστής επεξεργάζεται ακολουθίες παράλληλα και δεν έχει από πριν τη γνώση της σειράς τοποθέτησης των λέξεων (tokens) στην ακολουθία, χρησιμοποιούνται κωδικοποιήσεις θέσεων (positional encoding) οι οποίες προστίθενται στις ενσωματώσεις εισόδου για να παρέχουν πληροφορίες σχετικά με τις θέσεις των λέξεων.

Κάθε επίπεδο προσοχής ακολουθείται από ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης ως προς τη θέση, προσθέτοντας έναν μη γραμμικό μετασχηματισμό στοιχείων στο μοντέλο. Η

κανονικοποίηση του στρώματος και οι υπολειμματικές συνδέσεις εφαρμόζονται μετά από κάθε υπο-στρώμα (στρώμα προσοχής και πρόσθιας τροφοδότησης) για μεγαλύτερη σταθερότητα κατά τη διάρκεια της εκπαίδευσης.

Για να υπολογιστούν τα σκορ προσοχής:

$$\text{Scaled σκορ προσοχής} = \frac{QK^T}{\sqrt{d_k}}$$

Η αρχιτεκτονική του αποκωδικοποιητή είναι αρκετά κοντά με αυτή του κωδικοποιητή, με την διαφορά ότι περιέχει ένα επιπλέον στρώμα προσοχής πολλαπλών κεφαλών το οποίο ονομάζεται αυτό-προσοχή πολλαπλών κεφαλών με masking (Masked Multi-head Attention). Συνοπτικά, η αυτό-προσοχή με masking θα μπορούσε να παρομοιαστεί με το διάβασμα ενός βιβλίου μία λέξη τη φορά, όπου ο αναγνώστης φροντίζει να εστιάζει μόνο σε αυτά που έχει διαβάσει μέχρι τώρα και όχι σε αυτά που θα ακολουθήσουν. Κατά τη διάρκεια της εκπαίδευσης, θέλουμε το μοντέλο να μπορεί να κάνει προβλέψεις της επόμενης λέξης χωρίς να γνωρίζει τις λέξεις που θα ακολουθήσουν. Το masking βοηθά σ'αυτό.

Η συνάρτηση ενεργοποίησης softmax εφαρμόζεται στην έξοδο του μηχανισμού προσοχής, παράγοντας βάρη προσοχής που αθροίζονται στη μονάδα. Το στρώμα εξόδου στον αποκωδικοποιητή δημιουργεί την τελική ακολουθία εξόδου.

Οι μετασχηματιστές άλλαξαν εντελώς τον τρόπο επεξεργασίας των ακολουθιών σήμερα, αποτελούν την πρόσφατη εξέλιξη στον τομέα της μηχανικής μάθησης και έχουν καταφέρει να επιτύχουν εντυπωσιακά αποτελέσματα σε πολλές γλωσσικές εργασίες λόγω των δυνατοτήτων για παράλληλη επεξεργασία, των αποτελεσματικών μηχανισμών προσοχής και της ικανότητας σύλληψης πληροφορίας και συσχετίσεων σε πολύ μεγάλες ακολουθίες εισόδου.

3.4.4 Ερωτήματα (Queries), Κλειδιά (Keys), Τιμές (Values)

Οι έννοιες των διανυσμάτων κλειδιού, ερωτήματος και τιμής είναι θεμελιώδεις για την κατανόηση του μηχανισμού προσοχής, ο οποίος χρησιμοποιείται ευρέως στα σύγχρονα μοντέλα επεξεργασίας φυσικής γλώσσας, ιδιαίτερα στην αρχιτεκτονική του Transformer. Χρησιμοποιούνται για τον υπολογισμό των βαρών προσοχής, οι οποίες, με τη σειρά τους, καθορίζουν πόσο θα πρέπει να συνεισφέρει η κάθε θέση της ακολουθίας εισόδου στην έξοδο σε μια συγκεκριμένη θέση.

Για να κατανοήσουμε την έννοια του ερωτήματος, του κλειδιού και της τιμής στους μετασχηματιστές, ας καταλάβουμε πρώτα πώς λειτουργεί ο μηχανισμός προσοχής σε αυτά τα μοντέλα. Η προσοχή είναι ένας μηχανισμός που αποδίδει βαρύτητα σε κάθε λέξη μιας πρότασης ανάλογα με τη σημασία της. Το σταθμισμένο άθροισμα (αποτέλεσμα της προσθήκης πολλαπλασιασμένων αριθμών με διαφορετικά βάρη) αυτών των λέξεων χρησιμοποιείται στη συνέχεια για να υπολογιστεί η έξοδος του μοντέλου. Ωστόσο, η προσοχή δεν είναι απλώς ένα απλό άθροισμα των λέξεων. Λαμβάνει υπόψη τα συμφραζόμενα και τις εξαρτήσεις μεταξύ των λέξεων. Το ερώτημα και το κλειδί πολλαπλασιάζονται μαζί για να παράγουν τις βαθμολογίες προσοχής, οι οποίες στη συνέχεια χρησιμοποιούνται για τον υπολογισμό του σταθμισμένου αθροίσματος των τιμών. Αυτό το σταθμισμένο άθροισμα χρησιμοποιείται στη συνέχεια για τον υπολογισμό της εξόδου του μοντέλου.

Ας αναλύσουμε κάθε τύπο διανύσματος:

1. **Διάνυσμα ερωτήματος (Q):** Το διάνυσμα ερωτήματος είναι μια αναπαράσταση της εισόδου σε μια συγκεκριμένη θέση. Χρησιμοποιείται για την ανάκτηση σχετικών

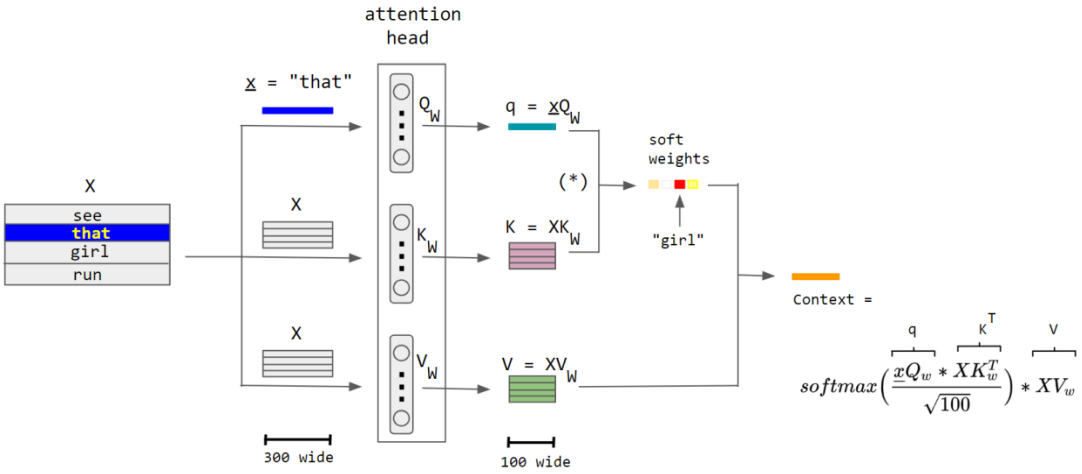
πληροφοριών από άλλα μέρη της ακολουθίας. Το διάνυσμα ερωτήματος συγκρίνεται με τα υπόλοιπα διανύσματα της ακολουθίας για να προσδιοριστεί η σημασία (βαρύτητα προσοχής) κάθε στοιχείου στην ακολουθία.

2. **Διάνυσμα κλειδιού (K)**: Το διάνυσμα κλειδιού είναι μια άλλη αναπαράσταση της ακολουθίας εισόδου και χρησιμοποιείται για να προσδιοριστεί πόση προσοχή πρέπει να δοθεί σε διαφορετικά μέρη της ακολουθίας κατά τη δημιουργία της εξόδου. Το διάνυσμα κλειδιού συγκρίνεται με διανύσματα ερωτήματος για τον υπολογισμό των βαρών προσοχής.

3. **Διάνυσμα τιμής (V)**: Το διάνυσμα τιμής συσχετίζεται με το περιεχόμενο σε κάθε θέση της ακολουθίας. Είναι οι πληροφορίες που θα χρησιμοποιηθούν στο τελικό σταθμισμένο άθροισμα κατά τον υπολογισμό της εξόδου του μηχανισμού προσοχής. Το διάνυσμα τιμής πολλαπλασιάζεται με το βάρος προσοχής που αποδίδεται στο αντίστοιχο ζεύγος κλειδιού-ερωτήματος.

Ας δείξουμε ένα παράδειγμα για καλύτερη κατανόηση.

Το παρακάτω παράδειγμα δείχνει πώς προσδιορίζονται οι συσχετίσεις όταν ένα δίκτυο έχει εκπαιδευτεί και έχει τα σωστά βάρη. Στο παρακάτω σχήμα εξετάζεται η λέξη «that» της πρότασης «see that girl run» και το δίκτυο θα πρέπει να είναι σε θέση να εντοπίσει τη λέξη girl ως την πιο σχετική με τη λέξη «that». Η πρόταση περνά από μία κεφαλή προσοχής που περιέχει 3 παράλληλες ροές- υποδίκτυα (Q=query, K=keys and V=value) 100 νευρώνων και παράγει το νοηματικό πλαίσιο (context) της πρότασης εισόδου.



Σχήμα 3.12: Μηχανισμός προσοχής (qkv)

- Το κεφαλαίο γράμμα X υποδηλώνει έναν πίνακα μεγέθους 4 × 300, που αποτελείται από τις ενσωματώσεις και των τεσσάρων λέξεων.

- Το μικρό υπογραμμισμένο γράμμα x υποδηλώνει το διάνυσμα ενσωμάτωσης της λέξης "that".

- Η κεφαλή προσοχής περιλαμβάνει τρία (κάθετα διατεταγμένα στο σχήμα 3.12) υποδίκτυα, το καθένα με 100 νευρώνες με έναν πίνακα βαρών μεγέθους 300 × 100. Ο

αστερίσκος μέσα στην παρένθεση "(*)" υποδηλώνει το softmax(qK^T / √100), δηλαδή δεν έχει ακόμη πολλαπλασιαστεί με τον πίνακα V.

- Η διαίρεση με √100 αποτρέπει μια υψηλή διακύμανση στο qK^T που θα επέτρεπε σε μια μεμονωμένη λέξη να κυριαρχεί υπερβολικά στο softmax, με αποτέλεσμα να εστιάζεται η προσοχή μόνο σε μία λέξη. [38]

3.4.5 Μηχανισμός προσοχής πολλαπλών κεφαλών

Σημειώνεται ότι η προσοχή πολλαπλών κεφαλών (multi-head attention) βασίζεται στην παραπάνω λογική, ωστόσο αντί να εκτελείται μία μόνο λειτουργία προσοχής, προβάλλονται γραμμικά τα ερωτήματα, τα κλειδιά και οι τιμές, και εφαρμόζεται παράλληλα η συνάρτηση προσοχής, επιτρέποντας στο μοντέλο να παρακολουθεί τις πληροφορίες από διαφορετικές αναπαραστάσεις σε διαφορετικές θέσεις, κάτι που δεν είναι δυνατό με μία κεφαλή προσοχής.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$

$$\text{όπου head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

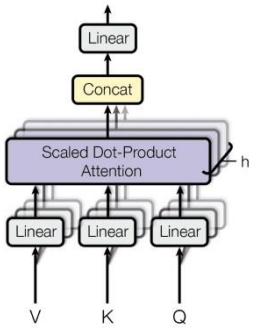
Κάθε μπλοκ προσοχής πολλαπλών κεφαλών αποτελείται από τέσσερα διαδοχικά επίπεδα:

Το πρώτο επίπεδο, όπου υπολογίζονται τα ερωτήματα, τα κλειδιά και οι τιμές από την ακολουθία εισόδου,

Το δεύτερο επίπεδο που περιέχει έναν μηχανισμό προσοχής με τη χρήση εσωτερικού γινομένου. Οι λειτουργίες που εκτελούνται τόσο στο πρώτο όσο και στο δεύτερο επίπεδο επαναλαμβάνονται h φορές και εκτελούνται παράλληλα, ανάλογα με τον αριθμό των κεφαλών που συνθέτουν το μπλοκ προσοχής πολλαπλών κεφαλών,

Το τρίτο επίπεδο όπου μια λειτουργία συνένωσης, ενώνει τις εξόδους των διαφορετικών κεφαλών ,

Το τέταρτο επίπεδο που περιέχει ένα τελικό γραμμικό (πυκνό) στρώμα που παράγει την έξοδο. [39]



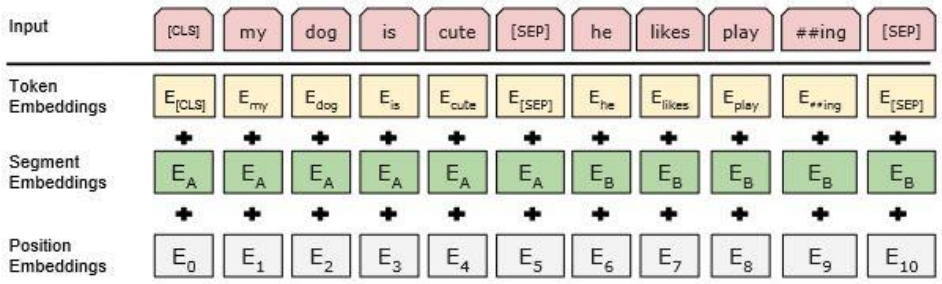
Σχήμα 3.13:Προσοχή πολλαπλών κεφαλών

3.4.6 BERT

Μετά την αρχική επιτυχία του μοντέλου μετασχηματιστή, υπήρξε μια μεγάλη προσπάθεια δημιουργίας νέων μοντέλων που να βασίζονται στην αρχιτεκτονική του, το καθένα με τις δικές του καινοτομίες και βελτιστοποιήσεις για διαφορετικές εργασίες. Το μοντέλο BERT (Bidirectional Encoder Representations from Transformers) είναι ένα προηγμένο μοντέλο

επεξεργασίας φυσικής γλώσσας (NLP) που αναπτύχθηκε από τους ερευνητές της Google το 2018 και βασίζεται στην αρχιτεκτονική των Μετασχηματιστών (Transformers) που παρουσιάστηκε από τους Vaswani et al. το 2017.

Μία από τις βασικές καινοτομίες του BERT είναι η αμφίδρομη (bidirectional) προσέγγισή του. Σε αντίθεση με τα προηγούμενα μοντέλα που επεξεργάζονταν κείμενο προς μία κατεύθυνση (είτε από αριστερά προς τα δεξιά είτε αντίστροφα), το BERT εξετάζει τόσο το αριστερό όσο και το δεξί πλαίσιο, παράγοντας έτσι βαθιές ενσωματώσεις συμφραζομένων (deep contextualized embeddings) και μια πιο ολοκληρωμένη κατανόηση του νοήματος της πρότασης. Οι ενσωματώσεις αυτές χρειάζονται πολύ μικρή προσαρμογή (fine-tuning) προκειμένου να επιτύχουν εντυπωσιακά αποτελέσματα σε σύνθετα προβλήματα του τομέα επεξεργασίας της φυσικής γλώσσας. [40]



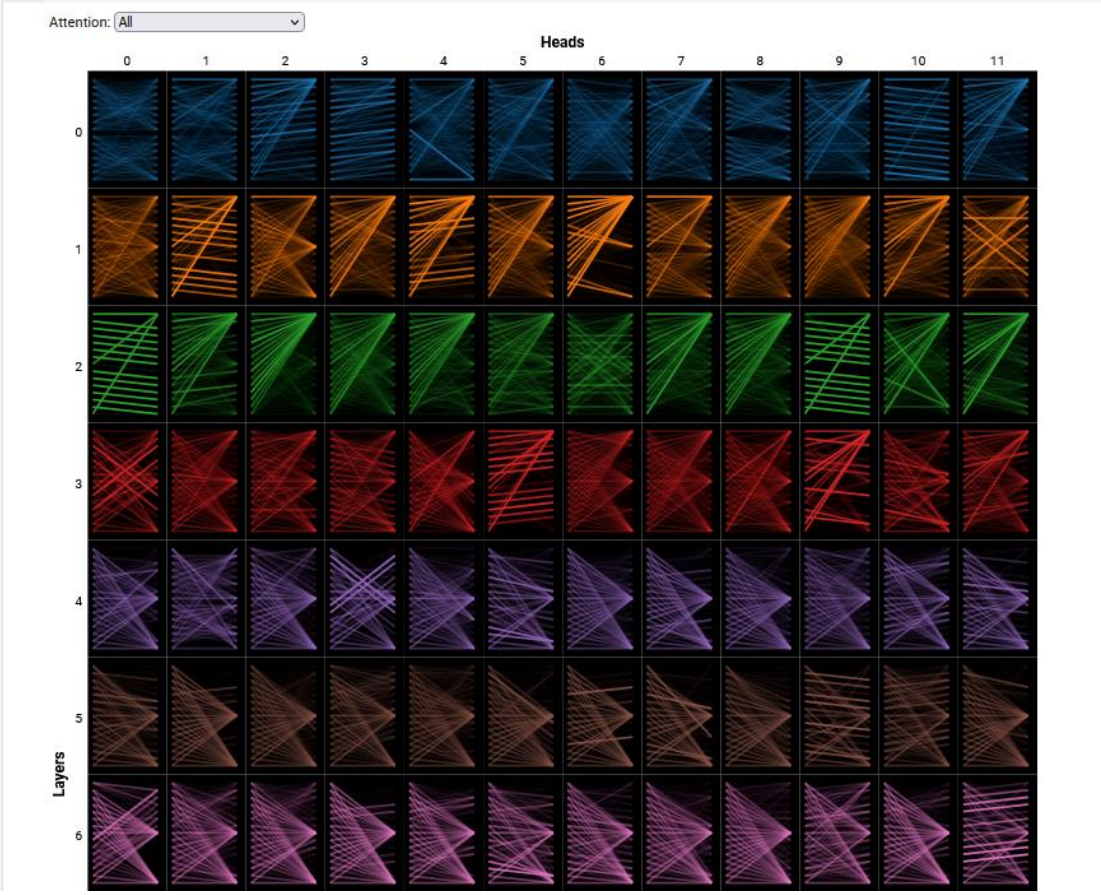
Σχήμα 3.14:Ενσωματώσεις εισόδου

Η αρχιτεκτονική του μοντέλου BERT βασίζεται μόνο στη δομή του κωδικοποιητή του Μετασχηματιστή και όχι του αποκωδικοποιητή που χρησιμοποιείται στις εργασίες seq2seq. Αυτή η απλοποίηση επιτρέπει στο BERT να εστιάζει μόνο σε αναπαραστάσεις συμφραζόμενων χωρίς την ανάγκη για δημιουργία κειμένων. Το BERT αποτελείται συνήθως από μια στοίβα μετασχηματιστών. Τα πολλαπλά επίπεδα επιτρέπουν στο BERT να αντιλαμβάνεται περίπλοκα γλωσσικά μοτίβα.

Η είσοδος κάθε επιπέδου αποτελείται από ενσωματώσεις λέξεων (token embeddings), ενσωματώσεις θέσης (position embeddings) και ενσωματώσεις τμημάτων (segment embeddings). Οι ενσωματώσεις λέξεων αντιπροσωπεύουν το νόημα μεμονωμένων λέξεων, οι ενσωματώσεις θέσης αντιστοιχούν στη θέση των λέξεων στην ακολουθία και οι ενσωματώσεις τμημάτων κάνουν διάκριση μεταξύ διαφορετικών τμημάτων στην είσοδο (π.χ. διαχωρισμός προτάσεων).

Η έξοδος ενός επιπέδου χρησιμεύει ως είσοδος στο επόμενο επίπεδο. Αυτή η επαναληπτική διαδικασία επιτρέπει στο μοντέλο να βελτιώσει την κατανόηση του κειμένου εισόδου καθώς αυτό περνά μέσα από διάφορα επίπεδα.

Ο μηχανισμός αυτό-προσοχής που υπάρχει σε κάθε επίπεδο καθώς και τα πολλαπλά επίπεδα κωδικοποιητών βοηθούν στη βαθύτερη κατανόηση κειμένου. Οι παράμετροι (βάρη και biases) των επιπέδων μετασχηματιστή μοιράζονται σε όλη τη στοίβα. Αυτή η κοινή χρήση παραμέτρων συμβάλλει στην αποτελεσματικότητα του μοντέλου και του δίνει τη δυνατότητα να μάθει μια συνεπή αναπαράσταση της γλώσσας.



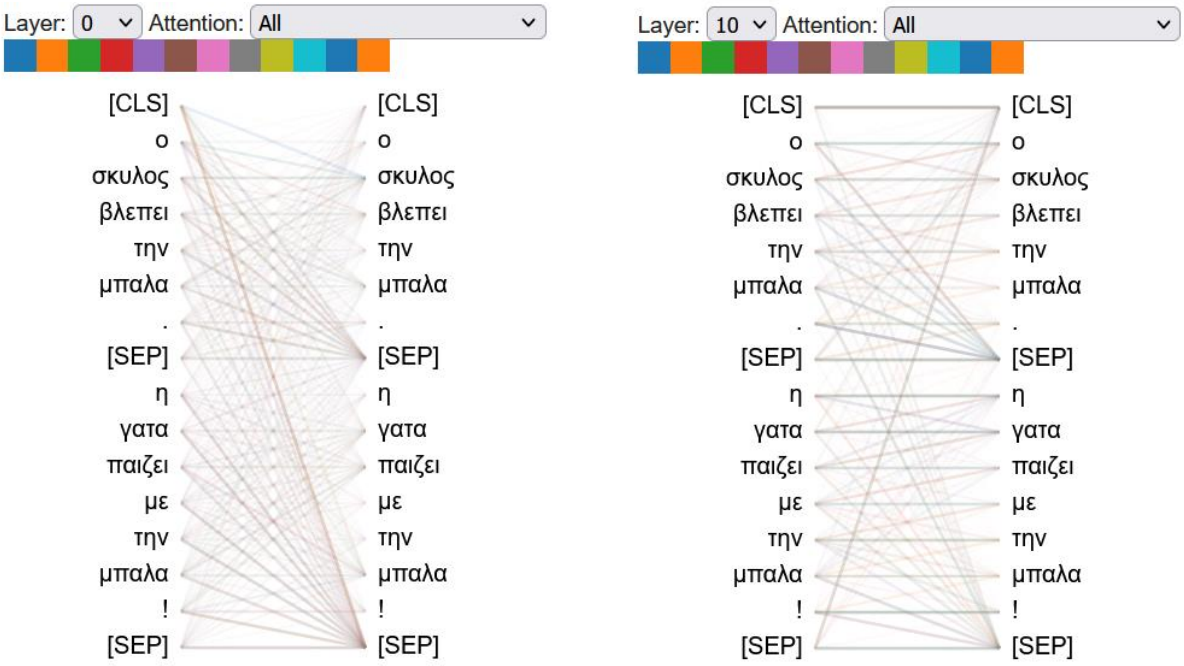
Σχήμα 3.15:Μηχανισμός προσοχής πολλαπλών κεφαλών

Η παραπάνω απεικόνιση δείχνει τον μηχανισμό προσοχής πολλαπλών κεφαλών του μοντέλου BERT (στο σχήμα 3.15 απεικονίζονται οι πρώτες 7 κεφαλές). Η συγκεκριμένη απεικόνιση έγινε με το BertViz, ένα διαδραστικό εργαλείο για την οπτικοποίηση της προσοχής σε μοντέλα γλώσσας Transformer όπως BERT, GPT2 ή T5. Εκτελείται μέσα σε ένα σημειωματάριο Jupyter ή Colab μέσω ενός απλού API Python που υποστηρίζει τα περισσότερα μοντέλα Huggingface. Το BertViz επεκτείνει το εργαλείο οπτικοποίησης Tensor2Tensor από τον Llion Jones, παρέχοντας πολλαπλές προβολές που η καθεμία προσφέρει μια μοναδική ματιά στον μηχανισμό προσοχής.

Στο συγκεκριμένο μοντέλο, όπως φαίνεται στο σχήμα 3.15, υπάρχουν πολλαπλοί μηχανισμοί προσοχής, που ονομάζονται κεφαλές, οι οποίες λειτουργούν παράλληλα μεταξύ τους. Η προσοχή πολλαπλών κεφαλών επιτρέπει στο μοντέλο να καταγράφει ένα ευρύτερο φάσμα σχέσεων μεταξύ των λέξεων από ότι θα ήταν δυνατό με έναν μόνο μηχανισμό προσοχής. Το BERT στοιβάζει επίσης πολλαπλά επίπεδα προσοχής, καθένα από τα οποία λειτουργεί στην έξοδο του επιπέδου που προέκυψε. Μέσω αυτής της επαναλαμβανόμενης σύνθεσης ενσωματώσεων λέξεων, ο BERT είναι σε θέση να σχηματίσει πολύ πλούσιες αναπαραστάσεις καθώς φτάνει στα βαθύτερα στρώματα του μοντέλου. Οι κεφαλές προσοχής δεν μοιράζονται παραμέτρους μεταξύ τους με αποτέλεσμα η κάθε κεφαλή να μαθαίνει ένα μοναδικό μοτίβο προσοχής. Η έκδοση του BERT που φαίνεται εδώ — BERT Base — έχει 12 επίπεδα και 12 κεφαλές, με αποτέλεσμα συνολικά 12 x 12 = 144 διακριτούς μηχανισμούς προσοχής.

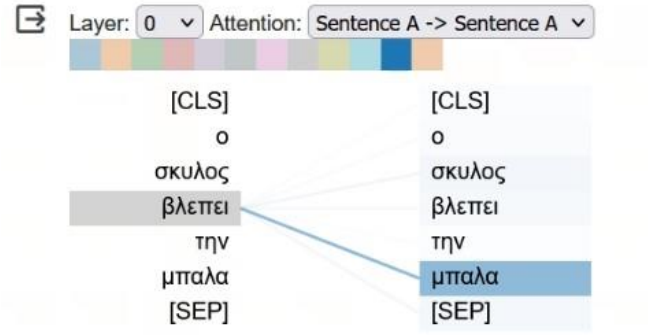
Οι γραμμές δείχνουν τα επίπεδα και οι στήλες τις κεφαλές προσοχής όπως αυτές διαμορφώθηκαν κατά την εισαγωγή των προτάσεων: «Ο σκύλος βλέπει την μπάλα» και «Η

γάτα παίζει με την μπάλα». Το κάθε κελί (πχ το πρώτο κελί που αντιστοιχεί στο πρώτο επίπεδο και πρώτη κεφαλή) ακολουθεί το δικό του μοτίβο προσοχής, με αποτέλεσμα το μοντέλο BERT να παράγει μια πλούσια σειρά μοτίβων προσοχής.



Σχήμα 3.16: Προβολές του 1ου και 9ου επιπέδου της κεφαλής προσοχής

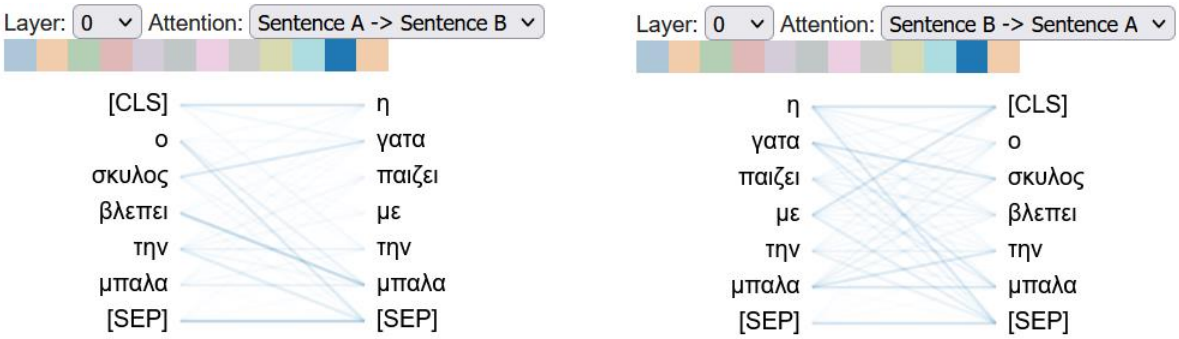
Αν ρίξουμε μια κρυφή ματιά σε κάθε κελί θα δούμε τις προβολές κεφαλών προσοχής. Συγκεκριμένα, στο σχήμα 3.16 φαίνονται οι κεφαλές του πρώτου και του προτελευταίου επιπέδου όπως αυτές διαμορφώνονται από την εισαγωγή απλού κείμενου στο προ-εκπαιδευμένο μοντέλο BERT. Η προσοχή απεικονίζεται ως μια γραμμή που συνδέει δυο λέξεις, τη λέξη στα αριστερά με την λέξη στα δεξιά. Η ένταση του χρώματος σχετίζεται με το βάρος της προσοχής (κυμαίνονται από το 0 έως το 1). Τα βάρη κοντά στο ένα εμφανίζονται ως πολύ σκούρες γραμμές, ενώ τα βάρη κοντά στο μηδέν εμφανίζονται ως αμυδρές γραμμές ή δεν είναι καθόλου ορατές. Ορισμένα ζεύγη λέξεων έχουν υψηλότερη προσοχή από τα άλλα.



Σχήμα 3.17:Συσχετισμός των λέξεων με την λέξη *βλέπει*

Τα σύμβολα [SEP] είναι ειδικά διακριτικά (token) που υποδεικνύουν τα όρια των προτάσεων και το [CLS] είναι ένα σύμβολο που προσαρτάται στο μπροστινό μέρος της εισόδου και

χρησιμοποιείται στα προβλήματα ταξινόμησης. Η οπτικοποίηση δείχνει ότι η προσοχή είναι μεγαλύτερη ανάμεσα στις λέξεις που βρίσκονται εντός της πρότασης. Το μοντέλο φαίνεται να κατανοεί ότι θα πρέπει να συσχετίσει λέξεις με άλλες λέξεις στην ίδια πρόταση για να κατανοήσει καλύτερα το περιεχόμενό τους.



Σχήμα 3.18: Η Αμφίδρομη ιδιότητα του BERT

Το μοντέλο BERT είναι αμφίδρομο (bidirectional), δηλαδή διαβάζει και προς τις δύο κατευθύνσεις. Στο σχήμα 3.18 φαίνονται οι γραμμές προσοχής όταν το μοντέλο επεξεργάζεται το κείμενο από τα αριστερά προς τα δεξιά (από την πρόταση Α στην πρόταση Β - αριστερά) και το αντίστροφο. Στο σχήμα 3.17 φαίνεται η λέξη **βλέπει** με ποιες λέξεις σχετίζεται και πόση προσοχή δίνει σε αυτές τις λέξεις. [39]

3.4.7 Προ-εκπαιδευμένα μοντέλα BERT

Το προ-εκπαιδευμένο μοντέλο BERT αναφέρεται σε ένα μοντέλο BERT που έχει εκπαιδευτεί σε ένα μεγάλο σύνολο δεδομένων κειμένου. Η φάση της προ-εκπαίδευσης είναι πολύ σημαντική για να μπορέσει το μοντέλο να συλλάβει πλούσιες αναπαραστάσεις της γλώσσας με βάση τα συμφραζόμενα.

Αρχικά προτάθηκαν δύο βασικά μοντέλα BERT :

- Βασικό μοντέλο ($BERT_{BASE}$): 12 επίπεδα κωδικοποιητών, 768 κρυμμένες μονάδες, 12 κεφαλές, 110M παράμετροι.

- Μεγάλο μοντέλο ($BERT_{LARGE}$): 24 επίπεδα κωδικοποιητών, 1024 κρυμμένες μονάδες, 16 κεφαλές, 340M παράμετροι.

Συγκεκριμένα, το μοντέλο BERT (για την αγγλική γλώσσα) είναι προ-εκπαιδευμένο στο BookCorpus, ένα σύνολο δεδομένων που αποτελείται από 11.038 αδημοσίευτα βιβλία και την αγγλική Wikipedia (εξαιρουμένων των λιστών, των πινάκων και των κεφαλίδων). Λίγο αργότερα ακολούθησαν οι εκδόσεις για τη Κινεζική και άλλες γλώσσες, καθώς και η πολυγλωσσική έκδοση.

Εφόσον επιθυμούμε να ταξινομήσουμε τα ελληνικά κείμενα, χρησιμοποιούμε το προ-εκπαιδευμένο μοντέλο BERT για την ελληνική γλώσσα. Η φάση της προ-εκπαίδευσης του συγκεκριμένου μοντέλου περιλαμβάνει τα εξής δεδομένα: α) ελληνική Wikipedia, β) βάση δεδομένων από τα πρακτικά του Ευρωπαϊκού Κοινοβουλίου και γ) Το ελληνικό μέρος του OSCAR, μια καθαρή έκδοση του Common Crawl. Το OSCAR εστιάζει συγκεκριμένα σε κείμενα πολυγλωσσικού περιεχομένου και στοχεύει να παρέχει ένα μεγάλο και ποικίλο σώμα για έρευνα στην επεξεργασία φυσικής γλώσσας (NLP) και σε άλλους συναφείς τομείς. [41]

Τα προ-εκπαιδευμένα μοντέλα BERT προσφέρουν εκδόσεις που λαμβάνουν υπόψη τα κεφαλαία (cased) και εκείνες που δεν γίνεται διάκριση πεζών –κεφαλαίων (uncased). Ένα πλεονέκτημα των uncased μοντέλων είναι ότι βοηθά στη μείωση του μεγέθους του λεξιλογίου κατά τη διάρκεια της εκπαίδευσης, καθώς όλες οι λέξεις αντιμετωπίζονται ως πεζά. Αυτό μπορεί να απλοποιήσει το μοντέλο και να μειώσει τους υπολογιστικούς πόρους που απαιτούνται.

Για παράδειγμα, όταν χρησιμοποιείται ένα προ-εκπαιδευμένο μοντέλο BERT που είναι uncased, αυτό σημαίνει ότι το μοντέλο εκπαιδεύτηκε σε κείμενα όπου όλες οι λέξεις είναι με πεζά. Αυτή η επιλογή μπορεί να είναι ωφέλιμη για ορισμένες εργασίες, όπως η ανάλυση συναισθημάτων ή η ταξινόμηση κειμένου, όπου η σημασία των λέξεων είναι συχνά ανεξάρτητη από την χρήση πεζών-κεφαλαίων.

Η καινοτομία του BERT βασίζεται κυρίως στις δυο νέες προσεγγίσεις προ-εκπαίδευσης που ονομάζονται **Masked Language Model** (MLM) και **Next Sentence Prediction** (NSP). Μέσω αυτών των καινοτομιών, κατέστη δυνατό το μοντέλο να αποκτήσει αποτελεσματικά τη γνώση που μπορεί να εξαχθεί από ένα κείμενο εξετάζοντας το και προς τις δύο κατευθύνσεις.

Το **Masked Language Model** είναι μία τεχνική κατά την προ-εκπαίδευση του μοντέλου που περιλαμβάνει τυχαία απόκρυψη λέξεων στις προτάσεις εισαγωγής. Δηλαδή, κατά τη διάρκεια της προ-εκπαίδευσης, ένα τυχαίο υποσύνολο λέξεων σε προτάσεις εισαγωγής αντικαθίσταται με ένα ειδικό διακριτικό [MASK]. Το μοντέλο δεν ενημερώνεται για το ποιες λέξεις καλύπτονται κατά τη διάρκεια της εκπαίδευσης. Ο στόχος του MML είναι να εκπαιδεύσει ένα μοντέλο που να μπορεί να προβλέπει καλυμμένες λέξεις μέσα σε μια πρόταση με βάση το πλαίσιο που παρέχεται από τις γύρω λέξεις. Η βασική καινοτομία είναι ότι το μοντέλο εκπαιδεύεται με αμφίδρομο τρόπο, επιτρέποντάς του να εξετάζει τόσο το αριστερό όσο και το δεξί πλαίσιο κάθε καλυμμένης λέξης.

Οι λεπτομέρειες της διαδικασίας MLM που ακολουθείται κατά την προ-εκπαίδευση του BERT για κάθε πρόταση είναι οι εξής:

- Το 15% των tokens είναι καλυμμένα.
- Στο 80% των περιπτώσεων, τα καλυμμένα tokens αντικαθίστανται από [MASK].
- Στο 10% των περιπτώσεων, τα καλυμμένα tokens αντικαθίστανται από ένα τυχαίο token (διαφορετικό) από αυτό που αντικαθιστούν.
- Στις υπόλοιπες περιπτώσεις 10%, τα καλυμμένα token παραμένουν ως έχουν.

Το **Next Sentence Prediction** (NSP) είναι άλλη μια καινοτομία που χρησιμοποιείται σε μοντέλα όπως BERT κατά την προ-εκπαίδευση για να βοηθήσει στην κατανόηση των σχέσεων και του νοηματικού πλαισίου μεταξύ των προτάσεων. Ο στόχος του NSP είναι να εκπαιδεύσει ένα μοντέλο για να μπορεί να προβλέπει εάν μια δεδομένη πρόταση ακολουθεί μια άλλη πρόταση στο αρχικό κείμενο. Το NSP ενθαρρύνει το μοντέλο να κατανοήσει τις συμφραζόμενες σχέσεις μεταξύ των προτάσεων, προωθώντας μια βαθύτερη κατανόηση του λόγου και του κειμένου πέρα από μία πρόταση. Αυτή η κατανόηση των σχέσεων μεταξύ των προτάσεων μπορεί στη συνέχεια να αξιοποιηθεί για διάφορες εργασίες επεξεργασίας φυσικής γλώσσας.

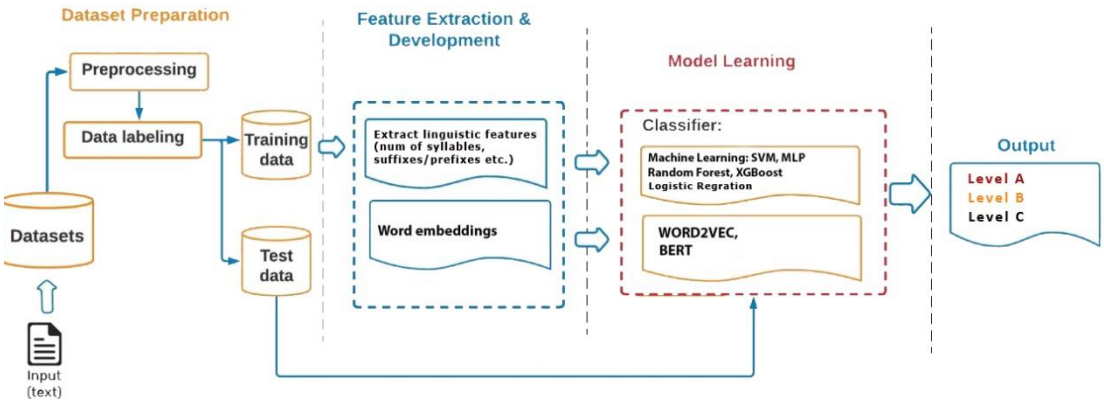
Ζεύγη προτάσεων επιλέγονται από το αρχικό έγγραφο και εισάγονται στο μοντέλο, το οποίο εκπαιδεύεται για να διακρίνει τις περιπτώσεις όταν η δεύτερη πρόταση διαδέχεται

πραγματικά την πρώτη πρόταση στο έγγραφο. Κατά τη διάρκεια της εκπαίδευσης του μοντέλου, στο 50% των παραδειγμάτων που παρέχονται, η δεύτερη πρόταση είναι πράγματι η επόμενη πρόταση στο αρχικό έγγραφο (positive pair), ενώ στο υπόλοιπο 50%, η δεύτερη πρόταση είναι μια τυχαία επιλεγμένη πρόταση από το από το έγγραφο (negative pair). Παράδειγμα θετικού ζευγαριού είναι: «Η γάτα είναι στο χαλάκι». / «Αυτή απολαμβάνει έναν υπνάκο» και αρνητικού ζευγαριού: «Ο ήλιος λάμπει». / «Ένας σκύλος γαβγίζει δυνατά». [42], [43]

Κεφάλαιο 4ο Μεθοδολογία και πειράματα

Σε αυτό το κεφάλαιο, θα παρουσιαστούν και θα επεξηγηθούν περαιτέρω οι μέθοδοι και πειράματα που χρησιμοποιήθηκαν στα πλαίσια της διπλωματικής εργασίας. Αρχικά, περιγράφεται το σύνολο δεδομένων που χρησιμοποιήθηκε για την ταξινόμηση κειμένων σε επίπεδα γλωσσομάθειας. Στη συνέχεια, παρατίθενται αναλυτικά τα 14 γλωσσικά χαρακτηριστικά που θα χρησιμοποιηθηκαν ως δεδομένα εισόδου στα μοντέλα μηχανικής μάθησης. Παρουσιάζονται τα αποτελέσματα από την εφαρμογή του κάθε μοντέλου και η εκτίμηση τους βάση των μετρικών που περιγράφτηκαν στο κεφάλαιο 3. Μετά την εφαρμογή των παραδοσιακών μοντέλων ταξινόμησης, εφαρμόστηκε η προσέγγιση των word embeddings. Χρησιμοποιήθηκε το προ-εκπαιδευμένο Word2Vec μοντέλο για την ταξινομήση των κειμένων. Σε αυτό το στάδιο εξετάστηκε η προσέγγιση του συνδυασμού των word embeddings με τα 14 γλωσσικά χαρακτηριστικά και εξετάστηκε εάν ο συνδυασμός αυτός βελτιώνει την απόδοση του μοντέλου. Τέλος, εστιάζουμε στην εφαρμογή του προ-εκπαιδευμένου BERT μοντέλου με ή χωρίς τα γλωσσικά χαρακτηριστικά.

Στο παρακάτω σχήμα φαίνεται η γενική μεθοδολογία που ακολουθήθηκε για την ταξινόμηση των κειμένων στα αντίστοιχα επίπεδα γλωσσομάθειας.



Σχήμα 4.1: Ταξινόμηση κειμένων (Generic pipeline)

4.1 Επιλογή του Dataset

Το σύνολο δεδομένων που επιλέχθηκε, αποτελείται από 885 κείμεν τα οποία είναι διαβαθμισμένα ανάλογα με το επίπεδο γλωσσομάθειας στο οποίο ανήκουν. Τα κείμενα αυτά αποτελούν το πρωτογενές υλικό βάσει του οποίου θα εκπαιδευτούν διάφορα μοντέλα μηχανικής μάθησης, με σκοπό να μπορούν να προβλέπουν την αναγνωσιμότητα ή αλλιώς την καταλληλότητα οποιουδήποτε άλλου κειμένου πάντοτε σε συνάρτηση με τα τρία επίπεδα ελληνομάθειας. Τα κείμενα αυτά έχουν τα εξής χαρακτηριστικά:

1. Ποικιλία πηγών: διαδίκτυο, εφημερίδες, περιοδικά, ενημερωτικά φυλλάδια, λογοτεχνικά ή επιστημονικά βιβλία κτλ.
2. Ευρύ φάσμα κειμενικών ειδών: επιστολές, άρθρα, σημειώματα, ομιλίες, διάλογοι, μελέτες, αγγελίες, οδηγίες χρήσης κτλ.
3. Διαβαθμισμένης δυσκολίας: το σύνολο δεδομένων χωρίζεται σε τρία μέρη, κάθε ένα από τα οποία αντιστοιχεί σε ένα επίπεδο ελληνομάθειας. Στα πλαίσια της διπλωματικής, θα χρησιμοποιηθούν 3 επίπεδα (Α, Β, Γ) και όχι 6 (Α1, Α2, Β1, Β2, Γ1,Γ2), λόγω πολύ μικρού συνόλου δεδομένων.

4. Επιλογή των κειμένων από ειδικούς της γλώσσας: τα κείμενα κάθε μέρους του σώματος εκπαίδευσης επελέγησαν ή δημιουργήθηκαν από εργαζόμενους ή εξωτερικούς συνεργάτες του Κέντρου Ελληνικής Γλώσσας βάσει του αναλυτικού προγράμματος των επιπέδων ελληνομάθειας με σκοπό τη χρήση τους στη διαμόρφωση των εξεταστικών θεμάτων των εξετάσεων πιστοποίησης επάρκειας της ελληνομάθειας.
5. Ως προς την αυθεντικότητά τους, βαίνει αυξανόμενη από το χαμηλότερο επίπεδο Α, το οποίο έχει ως επί το πλείστον κατασκευασμένα κείμενα με πολύ απλές γλωσσικές δομές, έως το υψηλότερο επίπεδο, το Γ, που περιλαμβάνει αυτούσια και ιδιαίτερα απαιτητικά κείμενα από διάφορες πηγές.

Επίπεδα γλωσσομάθειας	**Πλήθος κειμένων**	**Άθροισμα λέξεων**	**Μέσος όρος λέξεων**	**SD**
Α	388	64.916	167,31	109,89
Β	241	121.184	502,84	288,47
Γ	256	191.032	746,22	332,34
ΣΥΝΟΛΑ	**885**	**377.132**		

Πίνακας 4.1: Ποσοτικά δεδομένα του συνόλου δεδομένων (dataset)

4.2 Χρήση της βιβλιοθήκης SpaCy

Για την μετατροπή των κειμένων του συνόλου δεδομένων σε αντίστοιχα διανύσματα γλωσσικών χαρακτηριστικών χρησιμοποιείται η βιβλιοθήκη ανοιχτού κώδικα **SpaCy**, η οποία είναι γραμμένη σε γλώσσα Python και χρησιμοποιείται ευρέως για επεξεργασία φυσικής γλώσσας (NLP). Υποστηρίζει πολλές γλώσσες, μεταξύ αυτών και τα Ελληνικά. Για την διπλωματική μας επιλέχθηκε το προ-εκπαιδευμένο γλωσσικό μοντέλο **el_core_news_lg** ειδικά σχεδιασμένο για την ελληνική γλώσσα.

Λίγα λόγια για το el_core_news_lg:

"el": είναι ο κωδικός γλώσσας κατά το πρότυπο ISO 639-1 για την ελληνική γλώσσα για την οποία έχει εκπαιδευτεί το μοντέλο.

"core": δηλώνει ότι το μοντέλο είναι ένα βασικό μοντέλο, το οποίο παρέχει δυνατότητες NLP, όπως tokenization, επισήμανση μέρους του λόγου (POS tagging), αναγνώριση ονομαστικών οντοτήτων και άλλες γλωσσικές αναλύσεις.

"news": Η συμπερίληψη της λέξης "news" στο όνομα υποδηλώνει ότι το μοντέλο εκπαιδεύεται σε ποικίλα κείμενα, συμπεριλαμβανομένων άρθρων ειδήσεων, για να διασφαλίσει την κάλυψη διαφόρων τομέων και στυλ που βρίσκονται συνήθως σε περιεχόμενα που σχετίζεται με ειδήσεις.

"lg": Αυτό σημαίνει "μεγάλο". Τα μοντέλα SpaCy είναι διαθέσιμα σε διαφορετικά μεγέθη, που κυμαίνονται από μικρό (sm) έως μεσαίο (md) έως μεγάλο (lg). Τα μεγαλύτερα μοντέλα έχουν συνήθως περισσότερες παραμέτρους και είναι ικανά να καταγράφουν πλουσιότερα γλωσσικά μοτίβα. Μπορεί να προσφέρουν βελτιωμένη απόδοση, αλλά απαιτούν επίσης περισσότερους υπολογιστικούς πόρους.

Επομένως, με τη βοήθεια του **el_core_news_lg** μοντέλου υλοποιούνται διάφορες εργασίες επεξεργασίας φυσικής γλώσσας που απαιτούνται για την πειραματική μας διαδικασία, όπως η αναγνώριση λεκτικών μονάδων (Tokenization), η σήμανση μερών του λόγου (Part-of-

SpeechTagging), η αναγνώριση ονομαστικών οντοτήτων (NER – Named Entity Recognition), η λημματοποίηση (Lemmatization), διαχωρισμός σε προτάσεις (senter) κ.α.

4.3 Επιλογή γλωσσικών χαρακτηριστικων

Τα κριτήρια αναγνωσιμότητας αποτελούν ενδείξεις του βαθμού δυσκολίας ανάγνωσης και κατανόησης ενός κειμένου και μπορούν να εφαρμοστούν σε οποιαδήποτε γλώσσα, με τις κατάλληλες βέβαια προσαρμογές κάθε φορά. Πιο συγκεκριμένα, αποτελούν στατιστικούς δείκτες βαθμολόγησης ενός κειμένου όσον αφορά την δυσκολία (ή ευκολία) που το κείμενο αυτό παρουσιάζει κατά την ανάγνωσή του.

Ο υπολογισμός της αναγνωσιμότητας ενός κειμένου εξαρτάται από πολλούς παράγοντες, υπολογίσιμους ή μη. Οι μετρήσιμοι παράγοντες αναφέρονται στην "υλική" διάσταση ενός γραπτού κειμένου και στις ενδείξεις της επιφάνειας του κειμένου. Σ' αυτή την κατηγορία συντελεστών περιλαμβάνονται λόγου χάρη ο αριθμός των συλλαβών ανά λέξη και των λέξεων ανά πρόταση, ο αριθμός των πολυσύλλαβων λέξεων ενός κειμένου, ο αριθμός των προτάσεων, ο αριθμός των προθημάτων/επιθημάτων κ.λπ., ενδείξεις δηλαδή που μπορούν να υπολογιστούν και μάλιστα με τρόπο αντικειμενικό και χωρίς αποκλίσεις. [11]

Συγκεκριμένα, για την ταξινόμηση των κειμένων σε επίπεδα γλωσσομάθειας έχουν επιλεγεί 14 διαφορετικά γλωσσικά χαρακτηριστικά, στα οποία περιλαμβάνονται: α) παράμετροι που χρησιμοποιούν μερικοί από τους πιο διαδεδομένους τύπους υπολογισμού του βαθμού αναγνωσιμότητας διεθνώς, όπως οι FleschReadingEase, Flesch-KincaidGradeLevel, SMOG και FleschFogIndex και β) επιπλέον παράμετροι, βάσει πρόσφατης σχετικής βιβλιογραφίας ειδικά για την ελληνική γλώσσα. Ο προσδιορισμός των κειμενικών παραμέτρων της αναγνωσιμότητας έγινε από ειδική ερευνητική ομάδα γλωσσολόγων (ερευνητές του ΚΕΓ και εξωτερικοί συνεργάτες). Παρακάτω παρουσιάζονται οι παράμετροι και ο τρόπος υπολογισμού τους:

1. **Μέσο μήκος προτάσεων σε χαρακτήρες**

   Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο μεγαλύτερο το μέσο μήκος προτάσεων ενός κειμένου, τόσο δυσκολότερη η ανάγνωση του κειμένου. Έτσι, όσο μεγαλύτερη είναι η πρόταση, τόσο λιγότερα πληροφοριακά στοιχεία συγκρατεί ο αναγνώστης και επομένως καθίσταται δυσχερέστερη η επεξεργασία και κατανόησή της. Με τη χρήση της βιβλιοθήκης SpaCy και του pipeline component senter πραγματοποιήθηκε ο διαχωρισμός του κάθε κειμένου σε προτάσεις και στη συνέχεια υπολογίστηκε το μήκος των προτάσεων.

2. **Αριθμός προτάσεων ανά 100 λέξεις**

   Με τον παραπάνω τρόπο υπολογίζεται και ο αριθμός των προτάσεων ανά 100 λέξεις.

3. **Αριθμός λέξεων**

   Υπολογίζεται ο συνολικός αριθμός των λέξεων (tokens) ενός κειμένου αφου αφαιρέσουμε τα σημεία στίξης με τη χρηση της token.is_punct στο SpaCy. Όσο μεγαλύτερος ο αριθμός των λέξεων, τόσο δυσκολότερη η ανάγνωση του κειμένου.

4. **Αντωνυμικοί τύποι / 100 λέξεις**

Υπολογίζεται ο συνολικός αριθμός των αντωνυμικών τύπων ενός κειμένου ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: αντιστρόφως ανάλογη, όσο μεγαλύτερος ο μέσος αριθμός των αντωνυμικών τύπων, τόσο ευκολότερη η ανάγνωση του κειμένου. Για την εύρεση των αντωνυμικών τύπων χρησιμοποιείται η σήμανση μερών του λόγου (Part-of-SpeechTagging) που παρέχει η βιβλιοθήκη SpaCy. Οι αντωνυμίες γενικά εμφανίζονται με την ετικέτα "PRON" (pronoun) στην POS ανάλυση. Εδώ είναι ένα απλό παράδειγμα:

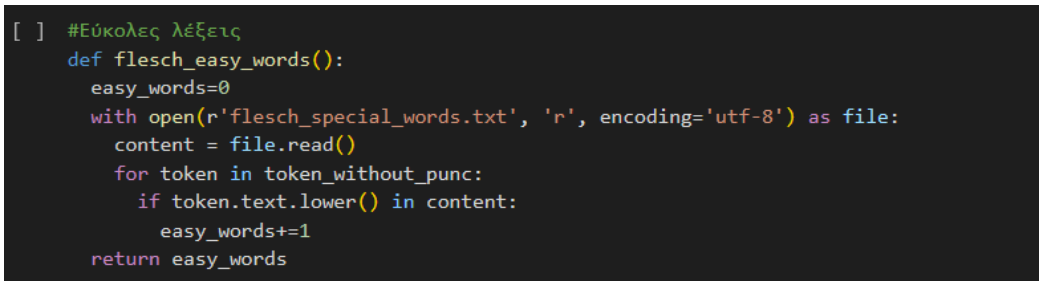
```
import csv, re, collections
import spacy, el_core_news_sm

nlp = spacy.load("el_core_news_sm")
doc=nlp("Γυμνάστε τον εγκέφαλο σας!, Όχι μόνο η χημεία αλλά και διάφορες άλλες μέθοδοι μπορούν να ενισχύσουν τη λειτουργία του εγκεφάλου μας.")

for token in doc:
  print(token.text, token.tag_)
```

```
Γυμνάστε VERB
τον DET
εγκέφαλο NOUN
σας PRON
! PUNCT
, PUNCT
Όχι ADV
μόνο ADV
η DET
χημεία NOUN
αλλά CCONJ
και CCONJ
διάφορες ADJ
άλλες PRON
μέθοδοι NOUN
μπορούν VERB
```

Σχήμα 4.2:Υπολογισμός αντωνυμικών τύπων

5. **Εύκολες λέξεις / 100 λέξεις**

Ως εύκολες λέξεις νοούνται μία συλλογή συγκεκριμένων λημματικών λεξιλογικών τύπων μαζί με τους υπόλοιπους κλιτικούς τύπους τους κατά περίπτωση, οι οποίοι αντιστοιχούν στο κατώτατο επίπεδο ελληνομάθειας Α1 (Παράρτημα-Εύκολες λέξεις). Υπολογίζεται ο συνολικός αριθμός «εύκολων» λέξεων ενός κειμένου ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: αντιστρόφως ανάλογη, όσο μεγαλύτερος ο μέσος αριθμός των εύκολων λέξεων, τόσο ευκολότερη η ανάγνωση του κειμένου.



```
[ ]  #Εύκολες λέξεις
     def flesch_easy_words():
       easy_words=0
       with open(r'flesch_special_words.txt', 'r', encoding='utf-8') as file:
         content = file.read()
         for token in token_without_punc:
           if token.text.lower() in content:
             easy_words+=1
       return easy_words
```

Σχήμα 4.3: Υπολογισμός εύκολων λέξεων

6. **Μεγάλες λέξεις / 100 λέξεις**

Ως μεγάλες λέξεις λογίζονται όσες αποτελούνται από τρεις ή περισσότερες συλλαβές. Υπολογίζεται ο συνολικός αριθμός των μεγάλων λέξεων ενός κειμένου ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο μεγαλύτερος ο μέσος αριθμός των μεγάλων λέξεων, τόσο δυσκολότερη η ανάγνωση του κειμένου.

```
# Μεγάλες λέξεις
def count_syllabes():
  big_syllabes_count=0
  for token in item:
    new=re.sub("(αι|ει|οι|υι|ου|αί|εί|οί|υί|ού|οί'|ια|αυ|ευ|αύ|εύ|ε|α|η|ι|ο|υ|ω|ά|έ|ή|ί|ό|ύ|ώ|ΐ|ϊ|ΰ|ϋ)",\
               "\\1-", token.text, flags=re.IGNORECASE)
    if (new.count('-')>2):
      big_syllabes_count+=1
  return big_syllabes_count
```

Σχήμα 4.4:Υπολογισμός μεγάλων λέξεων

7. **Λεξιλογική ποικιλία Guiraud's R (R (αριθμός λεξικών τύπων / √αριθμός λέξεων))**

   Η Guiraud's R είναι μετρική της λεξιλογικής δυσκολίας ενός κειμένου και χρησιμοποιεί τον εξής τύπο υπολογισμού:

$$R = \frac{types}{\sqrt{tokens}}$$

   Πρόκειται για τον λόγο του αριθμού των διαφορετικών λεξικών τύπων (types) προς την τετραγωνική ρίζα του αριθμού των λέξεων (tokens) ενός κειμένου. Για την καλύτερη διάκριση μεταξύ λεξικών τύπων και λέξεων παρατίθεται το εξής παράδειγμα: στην πρόταση «το μεγάλο ψάρι τρώει το μικρό ψάρι» υπάρχουν 7 λέξεις (tokens), αλλά 5 διαφορετικοί λεξικοί τύποι (types): το, μεγάλο, ψάρι, τρώει, μικρό (Γιάγκου 2009: 192). Αυτό το κειμενικό χαρακτηριστικό αναμένεται να λαμβάνει ως ανεξάρτητη μεταβλητή μία τιμή που θα είναι ευθέως ανάλογη της τιμής της εξαρτημένης μεταβλητής της δυσκολίας ανάγνωσης. Δηλαδή, όσο μεγαλύτερη θα είναι η λεξιλογική ποικιλία ενός κειμένου τόσο πιο δύσκολο θα θεωρείται στο σύνολό του.

```
freq = collections.Counter([token.text for token in item  if not (token.is_punct or token.like_num)])
guiraud=len(freq)/math.sqrt(len(token_without_punc))
print("Guiraud's R")
print("Αριθμός λεξικών τύπων: "+str(len(freq)))
print("R (αριθμός λεξικών τύπων / √αριθμός λέξεων): {} \n".format(round(guiraud,2)))
```

Σχήμα 4.5:Υπολογισμός λεξιλογικής ποικιλίας

8. **Λέξεις με προθήματα - επιθήματα / 100 λέξεις**

   Υπολογίζεται ο συνολικός αριθμός λέξεων με προθήματα και επιθήματα ενός κειμένου και στη συνέχεια η μέση συχνότητά τους ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο περισσότερα είναι τα προθήματα και επιθήματα σε ένα κείμενο τόσο δυσκολότερη είναι η ανάγνωσή του. (Παράρτημα-Προθήματα, Επιθήματα)

```
# Προθήματα
def flesch_prefixes():
  count=0
  prefixes =  set(open(r'flesch_prefixes.txt', 'r', encoding='utf-8').read().split())

  for token in token_without_punc:
    word=token.text.lower()

    for prefix in prefixes:

      if (re.match('^'+prefix+'(.+)', word)):
        count+=1
        break
  return count
```

Σχήμα 4.6:Υπολογισμός προθημάτων

```
# Επιθήματα
def flesch_postfixes():
  count=0
  postfixes =  set(open(r'flesch_postfixes.txt', 'r', encoding='utf-8').read().split())
  for token in token_without_punc:
    word=token.text.lower()

    for postfix in postfixes:
      if (word.endswith(postfix)):
        count+=1
        break
  return count
```

Σχήμα 4.7: Υπολογισμός επιθημάτων

9. **Λέξεις μεσοπαθητικής μορφολογίας / 100 λέξεις**

Υπολογίζεται ο συνολικός αριθμός των λέξεων μεσοπαθητικής μορφολογίας ενός κειμένου και η μέση συχνότητα των συγκεκριμένων τύπων ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο μεγαλύτερος ο αριθμός των μεσοπαθητικών τύπων σε ένα κείμενο τόσο δυσκολότερη είναι η ανάγνωσή του.

```
# Μεσοπαθητικά ρήματα, αποθετικά, ασταθή
def medium_passive():
  count=0
  back_fixs=['μαι', 'σαι', 'ται', 'μαστε', 'στε', 'νται', 'μουν', 'σουν', 'ταν', 'μαστε', 'μασταν', \
             'σαστε', 'σασταν', 'νταν', 'ντουσαν', 'θώ', 'θείς', 'θεί', 'θούμε', 'θείτε', 'θούν', 'τώ', \
             'τείς', 'τεί', 'τούμε', 'τείτε', 'τούν', 'θηκα', 'θηκες', 'θηκε', 'θήκαμε', 'θήκατε', 'θηκαν', \
             'θήκανε', 'τηκα', 'τηκες', 'τηκε', 'τήκαμε', 'τήκατε', 'τηκαν', 'τήκανε' ]
  not_pasive=['όταν', 'είμαι', 'είσαι', 'είναι', 'είμαστε', "είστε", "ήμουν", "ήσουν", "ήταν", "ήμασταν", \
              "ήμαστε", "ήσασταν", "ήσαστε" ]
  for token in token_without_punc:
    if any(token.text.endswith(b) for b in back_fixs) and (token.text.lower() not in not_pasive):
      count+=1
  return count
```

Σχήμα 4.8:Υπολογισμός λέξεων μεσοπαθητικής μορφολογίας

10. **Κύρια ονόματα / 100 λέξεις**

Ως κύρια ονόματα νοούνται όλες οι λέξεις που αρχίζουν με κεφαλαίο εκτός από τις περιπτώσεις που προηγείται τελεία, ερωτηματικό και θαυμαστικό και η αρχή παραγράφου. Για παράδειγμα, Κύρια Ονόματα θεωρούνται πρόσωπα, οργανισμοί, τοποθεσίες, ημερομηνίες, χρονικές εκφράσεις, ονόματα προϊόντων, ποσότητες, νομισματικές αξίες κ.α. Χρησιμοποιείται η βιβλιοθήκη SpaCy για την αναγνώριση κύριων ονομάτων (Name entity recognition - NER). Υπολογίζεται ο συνολικός

αριθμός των κύριων ονομάτων ενός κειμένου και η μέση συχνότητα των συγκεκριμένων τύπων ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: αντιστρόφως ανάλογη, όσο μεγαλύτερος ο αριθμός των κύριων ονομάτων τόσο ευκολότερη η ανάγνωση του κειμένου.

```
[ ]  # Κύρια ονόματα (using NER from spacy)
     def name_entity_rec():
       count=0
       for ent in item.ents:
         count+=1
       return count
```

Σχήμα 4.9: Υπολογισμός κύριων ονομάτων

11. **Σύνδεσμοι / 100 λέξεις**

Υπολογίζεται ο συνολικός αριθμός των συνδέσμων ενός κειμένου και η μέση συχνότητα των συγκεκριμένων τύπων ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο μεγαλύτερος ο αριθμός των συνδέσμων τόσο δυσκολότερη η ανάγνωση του κειμένου. (Παράρτημα-Σύνδεσμοι)

```
# Σύνδεσμοι
def flesch_links():
  links=0
  with open(r'flesch_links.txt', 'r', encoding='utf-8') as file:
    links = file.read().splitlines()
    s=item.text.lower()
    for link in links:
      result = len(re. findall(r"\b" + link + r"\b", s))
      if result>0:
        s = re.sub(r"\b" + link + r"\b", "0", s)
    count=len(re. findall("0", s))
    return(count)
```

Σχήμα 4.10: Υπολογισμός συνδέσμων

12. **Λόγιοι επιρρηματικοί τύποι / 100 λέξεις**

Ως λόγιοι επιρρηματικοί τύποι νοούνται όλες οι λέξεις που λήγουν σε -ως/-ως με εξαιρούμενες τις ακριβώς, αλλιώς, αμέσως, απλώς, ίσως, καθώς, κάπως, μήπως, όμως, όπως, πως, πώς, φως. Υπολογίζεται ο συνολικός αριθμός των λόγιων επιρρηματικών τύπων ενός κειμένου και η μέση συχνότητα των συγκεκριμένων τύπων ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο μεγαλύτερος ο αριθμός των λόγιων επιρρηματικών τύπων τόσο δυσκολότερη η ανάγνωση του κειμένου.

```
# Λόγιοι επιρρηματικοί τύποι
def scholar_adverbs_count():
  count=0
  not_adverbs=['ακριβώς', 'αλλιώς', 'αμέσως', 'απλώς', 'ίσως', 'καθώς', 'κάπως', \
               'μήπως', 'όμως', 'όπως', 'πως', 'πώς', 'φως']
  for token in token_without_punc:
    if (token.text.endswith(('ως','ώς')) ) and (token.text.lower() not in not_adverbs):
      count+=1
      # print(token.text)
  return count
```

Σχήμα 4.11: Υπολογισμός λόγιων επιρρηματικών τύπων

13. **Μετοχές / 100 λέξεις**

Καταμετρούνται οι τύποι της μεσοπαθητικής μετοχής με εξαιρούμενους τους τύπους που φαίνονται στο σχήμα 4.12. Υπολογίζεται ο συνολικός αριθμός των μετοχών ενός κειμένου και η μέση συχνότητα των συγκεκριμένων τύπων ανά 100 λέξεις. Συσχέτιση με τη δυσκολία ανάγνωσης: ανάλογη, όσο μεγαλύτερος ο αριθμός των μετοχών τόσο δυσκολότερη η ανάγνωση του κειμένου. (Παράρτημα-Επιθήματα μετοχών)

```
# Αριθμός επιθέτων και μετοχών: τύποι μεσοπαθητικής μετοχής
def adjectives_count():
  count=0
  not_adjectives=["χώρων", "πόντων", "πάντων", "μέντα", "μουσών", "ντοκουμέντα", \
                  "ντοκουμέντων", "κείμενο", "κειμένου", "κείμενα", "κειμένων", \
                  "αντικείμενο", "αντικειμένου", "αντικείμενα", "αντικειμένων", \
                  "κατεστημένο", "κατεστημένου", "κατεστημένα", "κατεστημένου", \
                  "γινόμενο", "γινομένου", "γινόμενα", "γινομένων", "δεδομένα", \
                  "δεδομένων", "ηγούμενος", "ηγουμένου", "ηγούμενοι", "ηγουμένων", \
                  "ηγουμένους", "υφιστάμενος", "υφιστάμενου", "υφιστάμενοι", "υφισταμένων", \
                  "ερωμένη", "ερωμένης", "ερωμένες", "ερωμένων", "η, προϊσταμένη", \
                  "προϊσταμένης", "προϊστάμενες", "προϊσταμένων"]
  suffixes =  set(open(r'flesch_adjective.txt', 'r', encoding='utf-8').read().split())
  for token in token_without_punc:

    for suffix in suffixes:
      if (token.text.endswith(suffix)) and (token.text.lower() not in not_adjectives):
        count+=1
        break
  return count
```

Σχήμα 4.12: Υπολογισμός μετοχών

14. **Κατάταξη ανάλογα με τον αριθμό των λέξεων**

Υπολογίζεται ο συνολικός αριθμός των λέξεων του κάθε κειμένου και στη συνέχεια άντιστοιχίζεται σε κάθε κείμενο ένας αριθμός από 1 έως 3, όπως φαίνεται στο παρακάτω σχήμα:

```
if (features_3<=140):
  features_14=1
elif (features_3<=600):
  features_14=2
 else:
  features_14=3
```

Σχήμα 4.13: Υπολογισμός κατάταξης βάση του αριθμού των λέξεων

Παράγοντες που λαμβάνονται υπόψη για τον υπολογισμό του βαθμού αναγνωσιμότητας των κειμένων			
1	**Μέσο μήκος προτάσεων σε χαρακτήρες**	Σχέση ανάλογη	Όσο μεγαλύτερο το μέσο μήκος προτάσεων ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
2	**Προτάσεις**	Σχέση ανάλογη	Όσο μεγαλύτερες είναι οι προτάσεις, τόσο **δυσκολότερη** η ανάγνωση και η κατανόηση τους
3	**Λέξεις**	Σχέση ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός λέξεων ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
4	**Αντωνυμικοί τύποι**	Σχέση αντιστρόφως ανάλογη	Όσο μεγαλύτερος ο μέσος αριθμός των αντωνυμικών τύπων ενός κειμένου, τόσο **ευκολότερη** η ανάγνωση του
5	**Εύκολες λέξεις**	Σχέση αντιστρόφως ανάλογη	Όσο μεγαλύτερος ο μέσος αριθμός των εύκολων λέξεων ενός κειμένου, τόσο **ευκολότερη** η ανάγνωση του
6	**Μεγάλες λέξεις**	Σχέση ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός των μεγάλων λέξεων ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
7	**Guiraud's R**	Σχέση ανάλογη	Όσο μεγαλύτερη θα είναι η λεξιλογική ποικιλία ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
8	**Προθήματα και επιθήματα**	Σχέση ανάλογη	Όσο περισσότερα είναι τα προθήματα και επιθήματα ενός κειμένου, τόσο δυσκολότερη η ανάγνωσή του
9	**Μεσοπαθητικά ρήματα, αποθετικά και ασταθή**	Σχέση ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός των μεσοπαθητικών τύπων ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
10	**Κύρια ονόματα**	Σχέση αντιστρόφως ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός των κυρίων ονομάτων ενός κειμένου, τόσο **ευκολότερη** η ανάγνωση του
11	**Σύνδεσμοι**	Σχέση ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός των συνδέσμων ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
12	**Λόγιοι επιρρηματικοί τύποι**	Σχέση ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός των λόγιων επιρρηματικών τύπων ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
13	**Αριθμός μετοχών**	Σχέση ανάλογη	Όσο μεγαλύτερος είναι ο αριθμός των μετοχών ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του
14	**Κατηγοριοποίηση των κειμένων ανά επίπεδο ελληνομάθειας με βάση την έκτασή τους**	<=140 λέξεις Α επ. <=600 λέξεις Β επ. >600 λέξεις Γ επ.	Όσο μεγαλύτερη είναι η έκταση ενός κειμένου, τόσο **δυσκολότερη** η ανάγνωση του

Πίνακας 4.2: Γλωσσικά χαρακτηριστικά που θα υπολογιστούν για την ταξινόμηση των κειμένων

Μετά την επιλογή των χαρακτηριστικών, ακολούθησε η εξαγωγή των 14 γλωσσικών χαρακτηριστικών για κάθε κείμενο, οι οποίες στη συνέχεια θα αποτελέσουν την είσοδο σε μοντέλα μηχανικής μάθησης που έχουμε επιλέξει.

	Μέσο μήκος προτάσεων σε χαρακτήρες	Προτάσεις	Συνολικός αριθμός λέξεων	Αντωνυμικοί τύποι	Εύκολες λέξεις	Μεγάλες λέξεις	Λεξιλογική ποικιλία Guiraud	Προθήματα - Επιθήματα	Λέξεις μεσοπαθητικής μορφολογίας	Κύρια ονόματα	Σύνδεσμοι	Λόγιοι επιρρηματικοί τύποι	Μετοχές	Κατάταξη ανάλογα με τον αριθμό των λέξεων
0	48.4000	9.8039	102.0	9.8039	60.7843	25.4902	7.8222	8.0490	1.9608	0.0000	12.7451	1.9608	0.9804	1.0
1	31.3934	15.8442	385.0	6.2338	61.0390	32.2078	9.8362	6.1844	1.2987	1.2987	10.3896	0.0000	0.7792	4.0
2	39.8571	13.0841	107.0	7.4766	45.7944	33.6449	6.3805	16.0654	5.6075	0.9346	5.6075	0.0000	0.0000	1.0
3	25.7857	21.8750	64.0	1.5625	21.8750	50.0000	6.7500	15.8438	3.1250	1.5625	4.6875	0.0000	0.0000	1.0
4	17.1613	33.3333	93.0	9.6774	44.0860	40.8602	6.9476	9.7849	4.3011	2.1505	5.3763	0.0000	0.0000	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
398	108.0612	5.1525	951.0	7.2555	50.7886	41.0095	14.3328	12.8528	1.9979	0.2103	11.8822	1.0515	1.1567	6.0
399	130.5000	3.7313	268.0	10.0746	58.9552	31.3433	10.4455	11.3657	1.4925	0.0000	19.4030	0.3731	0.3731	4.0
400	124.1875	4.6647	343.0	8.7464	50.1458	41.3994	11.0150	12.4694	5.8309	0.0000	14.2857	0.8746	1.4577	4.0
401	182.7143	3.0172	232.0	1.2931	46.9828	43.9655	8.7319	19.2414	2.5862	0.4310	12.0690	0.8621	1.7241	4.0
402	294.6000	2.0080	249.0	5.6225	47.7912	43.3735	9.4425	21.1566	2.0080	0.0000	9.6386	0.0000	1.6064	4.0

Σχήμα 4.14: Εξαγωγή γλωσσικών χαρακτηριστικών από τα κείμενα

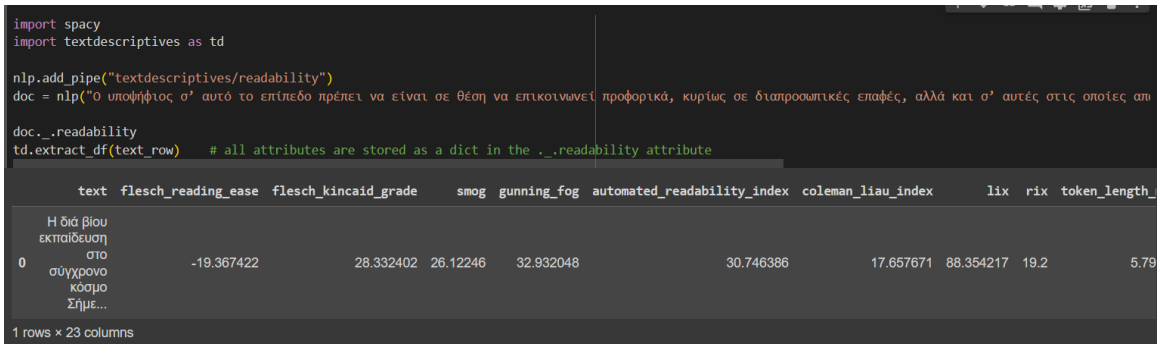
Παρακάτω φαίνονται τα αποτελέσματα που τυπώνονται για το κάθε κείμενο. Η διαδικασία αυτή επαναλαμβάνεται για τα 885 κείμενα εκπαίδευσης. Ο τελικός πίνακας που περιέχει όλα τα διανύσματα εισόδου έχει διάσταση [885 x 14] και περιέχει τα γλωσσικά χαρακτηριστικά (handcrafted features) του κάθε κειμένου.

```
[ , Πήγα κάποτε σε ένα τόπο σε μια χώρα μακρινή -  μπορεί και κοντινή -  που τον έλεγαν Τηλεθώσα
Προτάσεις
Αριθμός προτάσεων: 12
Μέσο μήκος προτάσεων σε χαρακτήρες: 48.75
Αριθμός προτάσεων ανά 100 λέξεις: 9.09

Λέξεις
Αριθμός λέξεων: 132
Λέξεις / πρόταση: 11.0

Αντωνυμικοί τύποι
Αντωνυμικοί τύποι: 10
Αντωνυμικοί τύποι / πρόταση: 0.83
Αντωνυμικοί τύποι / 100 λέξεις: 7.58

Εύκολες λέξεις
Ευκολες λέξεις: 77
Εύκολες λέξεις / πρόταση: 6.42
Εύκολες λέξεις / 100 λέξεις: 58.33

Μεγάλες λέξεις (>2 συλλαβές)
Αριθμός μεγάλων λέξεων: 28
Μεγάλες λέξεις / πρόταση: 2.33
Μεγάλες λέξεις / 30 προτάσεις: 70.0
Μεγάλες λέξεις / 100 λέξεις: 21.21

Guiraud's R
Αριθμός λεξικών τύπων: 97
R (αριθμός λεξικών τύπων / √αριθμός λέξεων): 8.44

Σύνδεσμοι
Σύνδεσμοι: 15
Σύνδεσμοι / πρόταση: 1.25
Σύνδεσμοι / 100 λέξεις: 11.36

Προθήματα και επιθήματα
Αριθμός λέξεων με προθήματα: 8
Αριθμός λέξεων με επιθήματα: 18
Σύνολο λέξεων με προθήματα - επιθήματα: 26
Λέξεις με προθήματα - επιθήματα / πρόταση: 2.17
Λέξεις με προθήματα - επιθήματα / 100 λέξεις: 19.7

Μεσοπαθητικά ρήματα, αποθετικά, ασταθή
Αριθμός λέξεων μεσοπαθητικής μορφολογίας: 1
Λέξεις μεσοπαθητικής μορφολογίας / πρόταση: 0.08
Λέξεις μεσοπαθητικής μορφολογίας / 100 λέξεις: 0.76

Κύρια ονόματα
Αριθμός κυρίων ονομάτων: 0
Κύρια ονόματα / πρόταση: 0.0
Κύρια ονόματα / 100 λέξεις: 0.0

Λόγιοι επιρρηματικοί τύποι
Αριθμός λόγιων επιρρηματικών τύπων: 0
Λόγιοι επιρρηματικοί τύποι / πρόταση: 0.0
Λόγιοι επιρρηματικοί τύποι / 100 λέξεις: 0.0

Αριθμός μετοχών και επιθέτων
Αριθμός μετοχών: 0
Μετοχές / πρόταση: 0.0
Μετοχές / 100 λέξεις: 0.0

[48.75, 9.0909, 132, 7.5758, 58.3333, 21.2121, 8.4428, 6.197, 0.7576, 0.0, 11.3636, 0.0, 0.0, 2]
```

Σχήμα 4.15: Εκτύπωση των γλωσσικών χαρακτηριστικών του κάθε κειμένου

4.4 Εξαγωγή επιπλέον μετρικών από το spaCy

Στην έκδοση 3 της βιβλιοθήκης spaCy υπάρχει ένα pipeline component που υπολογίζει περιγραφικά στατιστικά στοιχεία, μετρήσεις αναγνωσιμότητας και συντακτική πολυπλοκότητα (απόσταση εξάρτησης) του κειμένου εισόδου. Όπως φαίνεται στο παρακάτω παράδειγμα υπολογίζονται 22 βασικές παράμετροι, μερικοί από τους οποίους είναι: flesch_reading_ease, flesch_kincaid_grade,smog, gunning_fog, automated_readability_index, unique_tokens κ.α.



```
import spacy
import textdescriptives as td

nlp.add_pipe("textdescriptives/readability")
doc = nlp("Ο υποψήφιος σ' αυτό το επίπεδο πρέπει να είναι σε θέση να επικοινωνεί προφορικά, κυρίως σε διαπροσωπικές επαφές, αλλά και σ' αυτές στις οποίες απ

doc._.readability
td.extract_df(text_row)    # all attributes are stored as a dict in the ._.readability attribute
```

	text	flesch_reading_ease	flesch_kincaid_grade	smog	gunning_fog	automated_readability_index	coleman_liau_index	lix	rix	token_length_
0	Η διά βίου εκπαίδευση στο σύγχρονο κόσμο Σήμε...	-19.367422	28.332402	26.12246	32.932048	30.746386	17.657671	88.354217	19.2	5.79

1 rows × 23 columns

Σχήμα 4.16: Εξαγωγή επιπλέον γλωσσικών χαρακτηριστικών από το spaCy

Οι παραπάνω 22 παράμετροι είχαν εξαχθεί για τα κείμενα εκπαίδευσης και χρησιμοποιήθηκαν μαζί με τα 14 γλωσσικά χαρακτηριστικά που είχαν επιλεγεί από το ΚΕΓ για να διερευνηθεί η πιθανότητα βελτίωσης των αποτελεσμάτων με τη χρήση αυτών των επιπλέον στοιχείων. Τα αποτελέσματα έδειξαν ότι δεν βελτιώνουν περαιτέρω τις προβλέψεις του μοντέλου και εξαιρέθηκαν από τη διαδικασία εκπαίδευσης.

4.5 Χρήση παραδοσιακών μοντέλων

4.5.1 Ρύθμιση Παραμέτρων Μοντέλων Μηχανικής Μάθησης

Η επιλογή των κατάλληλων υπερπαραμέτρων είναι κρίσιμης σημασίας στη διαδικασία ανάπτυξης μοντέλων μηχανικής μάθησης και παίζουν καθοριστικό ρόλο στη βελτιστοποίηση της απόδοσης τους. Οι υπερπαράμετροι είναι εξωτερικές ρυθμίσεις του μοντέλου και δεν εκπαιδεύονται από τα δεδομένα. Μερικά παραδείγματα υπερπαραμέτρων π.χ στα νευρωνικά δίκτυα είναι ο ρυθμός εκμάθησης, ο αριθμός των κρυφών στρωμάτων, ο αριθμός νευρώνων σε κάθε στρώμα, dropout rate κτλ.

Επειδή, η επιλογή των διαφορετικών συνδυασμών μεταξύ των παραμέτρων είναι μια χρονοβόρα διαδικασία και ίσως να μην εξεταστούν και όλοι οι συνδυασμοί, η χρήση μιας αυτοματοποιημένης διαδικασίας για την εύρεση του καλύτερου συνδυασμού είναι πολύ χρήσιμη. Πιο συγκεκριμένα, κατά την εκπαίδευση όλων των παραδοσιακών μοντέλων έγινε η χρήση της **GridSearchCV (Grid Search Cross Validation)** που παρέχεται από τη βιβλιοθήκη scikit-learn και αποτελεί μέρος της sklearn.model_selection. Το GridSearchCV αξιολογεί την απόδοση του μοντέλου δοκιμάζοντας όλους τους δυνατούς συνδυασμούς μέσα από ένα συγκεκριμένο πλέγμα υπερπαραμέτρων, βρίσκοντας τον βέλτιστο συνδυασμό για κάθε μοντέλο.

Χρησιμοποιεί τη διασταυρούμενη επικύρωση (cross-validation), όπου διαιρείται το σύνολο δεδομένων σε k (περίπου) ίσου μεγέθους μέρη. Το μοντέλο εκπαιδεύεται σε k-1 μέρη και ελέγχεται στο υπόλοιπο μέρος. Αυτή η διαδικασία επαναλαμβάνεται k φορές και υπολογίζεται η μέση απόδοση όλων των μερών.

Εάν για παράδειγμα η **GridSearchCV** έχει 4 υπερπαραμέτρους, η καθεμία να έχει με 5 τιμές, τότε ο συνολικός αριθμός συνδυασμών υπολογίζεται ως το γινόμενο του αριθμού των τιμών για κάθε υπερπαράμετρο. Στην παράδειγμα μας, θα ήταν:

$$\text{Αριθμός συνδυασμών} = 5 \times 5 \times 5 \times 5 = 625$$

Έτσι, θα υπήρχαν 625 συνδυασμοί για αξιολόγηση κατά την αναζήτηση πλέγματος. Πρέπει να ληφθεί υπόψη ότι καθώς αυξάνεται ο αριθμός των υπερπαραμέτρων ή οι τιμές τους, ο χώρος αναζήτησης αυξάνεται εκθετικά, οδηγώντας σε υψηλότερο υπολογιστικό κόστος. Η αναζήτηση πλέγματος αξιολογεί εξαντλητικά όλους τους συνδυασμούς, καθιστώντας την υπολογιστικά ακριβή για μεγάλα σύνολα δεδομένων ή σε πολύπλοκα μοντέλα.

Συγκεκριμένα, στα πλαίσια της διπλωματικής εφαρμόστηκε η **GridSearchCV** σε όλα τα παραδοσιακά μοντέλα μηχανικής μάθησης λόγω του μικρού μεγέθους του συνόλου δεδομένων και επιλέχθηκαν οι συνδυασμοί εκείνοι που οδηγούν τα μοντέλα στη βέλτιστη απόδοση.

4.5.2 XGBoost

Στον τομέα των αλγορίθμων ενίσχυσης κλίσης, το XGBoost ξεχωρίζει ως ένα ισχυρό εργαλείο για εποπτευόμενες εργασίες εκμάθησης, γνωστό για την αποτελεσματικότητά του σε ένα ευρύ φάσμα εργασιών μηχανικής εκμάθησης, συμπεριλαμβανομένης της ταξινόμησης κειμένου. Πολλά από τα πλεονεκτήματα του όπως ανθεκτικότητα στην υπερπροσαρμογή, δυνατότητα χειρισμού μη γραμμικών σχέσεων κ.α. το έχουν καταστήσει δημοφιλή επιλογή τόσο μεταξύ των επαγγελματιών όσο και των ερευνητών για την επίτευξη υψηλής ακρίβειας στις εργασίες ταξινόμησης κειμένου.

Αρχικά, τα δεδομένα εισόδου χωρίζονται σε σώμα εκπαίδευσης (training set) και σώμα ελέγχου(test set) σε αναλογία 80:20 (708 κείμενα εκπαίδευσης : 117 κείμενα ελέγχου). Σε κάθε σώμα διατηρείται η αναλογία στις κλάσεις και υπάρχει περίπου ίδια αναλογία από κάθε κλάση στα σύνολα εκπαίδευσης και ελέγχου. Για την επίτευξη των βέλτιστων αποδόσεων του XGBoost εφαρμόστηκε η τεχνική της βελτίωσης υπερπαραμέτρων μέσω της **GridSearchCV**. Πραγματοποιήθηκε η αναζήτηση για συγκεκριμένες παραμέτρους (learning rate, n_estimators, max_depth, subsample, colsample_bytree) και τις αντίστοιχες τιμές τους όπως αυτές φαίνονται παρακάτω.. Η διεργασία δεν διήρκησε πολύ μέχρι να βρεθεί το πιο αποδοτικό μοντέλο.

```
# Define the parameter grid
param_grid = {
    'learning_rate': [0.1, 0.01, 0.001, 0.007],
    'n_estimators': [100, 120, 150, 200, 300],
    'max_depth': [3, 4, 5],
    'subsample': [0.8, 0.9, 1.0],
    'colsample_bytree': [0.8, 0.9, 1.0]
}
```

Σχήμα 4.17: Αναζήτηση βέλτιστων υπερπαραμέτρων (XGBoost)

```
Best Parameters: {'colsample_bytree': 0.9,
                  'learning_rate': 0.1,
                  'max_depth': 4,
                  'n_estimators': 300,
                  'subsample': 0.8}
```

Σχήμα 4.18: Βέλτιστες υπερπαράμετροι για τον XGBoost

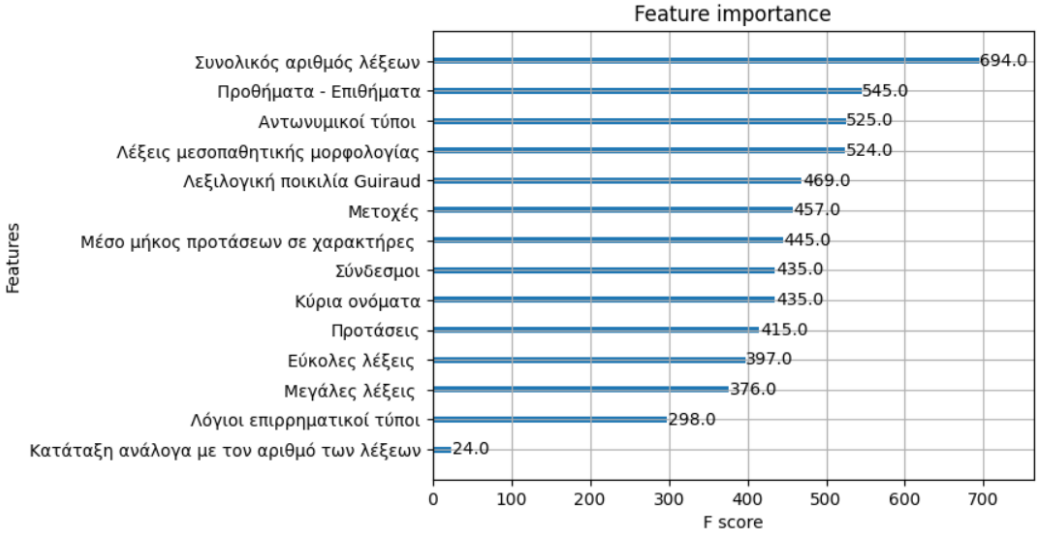
Μετά τον καθορισμό των βέλτιστων υπερπαραμέτρων του μοντέλου, ακολουθεί η εκπαίδευση του μοντέλου. Μετά το τέλος της εκπαίδευσης, οι μετρικές (accuracy, recall, f1-score, precision) στο σώμα εκπαίδευσης είναι όλες 1.0. Το μοντέλο έχει μάθει στο σώμα εκπαίδευσης και θα πρέπει να αξιολογηθεί η δυνατότητά του να γενικεύει σε άγνωστα σύνολα. Το ζητούμενο είναι να μάθει να αποτυπώνει αποτελεσματικά τις περίπλοκες σχέσεις και όχι να απομνημονεύει τα δεδομένα εκπαίδευσης.

```
XGBoost Train Accuracy: 1.0

Train Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       306
           1       1.00      1.00      1.00       187
           2       1.00      1.00      1.00       215

    accuracy                           1.00       708
   macro avg       1.00      1.00      1.00       708
weighted avg       1.00      1.00      1.00       708

XGBoost Test Accuracy: 0.8813559322033898

 Test Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96        82
           1       0.80      0.81      0.81        54
           2       0.81      0.83      0.82        41

    accuracy                           0.88       177
   macro avg       0.86      0.87      0.86       177
weighted avg       0.88      0.88      0.88       177
```

Σχήμα 4.19: XGBoost classification reports (train & test sets)

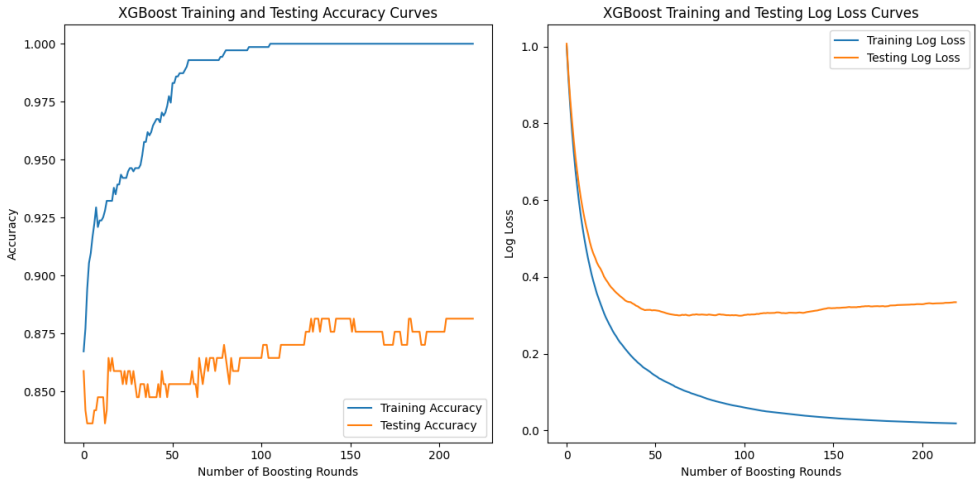
Όπως παρατηρούμε από το classification report παραπάνω, η ακρίβεια στο σώμα ελέγχου φτάνει στο 0,88. Με μια πιο προσεκτική ματιά, παρατηρείται ότι το μοντέλο έχει μάθει με πολύ μεγάλη ακρίβεια να προβλέπει την κλάση 0 (Α επίπεδο γλωσσομάθειας) και τα υπόλοιπα δύο επίπεδα να κυμαίνονται σε πιο χαμηλά ποσοστά.

Μεγάλο ενδιαφέρον έχει η χρήση της συνάρτησης plot_importance από τη βιβλιοθήκη xgboost, η οποία χρησιμοποιήθηκε για να δείξει ποια γλωσσικά χαρακτηριστικά κρίνονται πιο σημαντικά με βάση τη συμβολή τους στην απόδοση του μοντέλου. Το διάγραμμα ράβδων που παράγεται, δείχνει ότι τα πιο σημαντικά γλωσσικά χαρακτηριστικά είναι ο συνολικός αριθμός λέξεων, τα προθήματα-επιθήματα και αντωνυμικοί τύποι κ.α. Αυτό που σίγουρα δεν παίζει πολύ σπουδαίο ρόλο στην απόδοση του μοντέλου είναι η κατάταξη ανάλογα με τον αριθμό των λέξεων. Η συνάρτηση αυτή είναι πάρα πολύ χρήσιμη γιατί μπορεί να αναδείξει τα πιο σημαντικά χαρακτηριστικά του μοντέλου και να βοηθήσει στην περαιτέρω βελτίωση ως προς την επιλογή ή αποφυγή συγκεκριμένων χαρακτηριστικών.



Σχήμα 4.20: Διάγραμμα κατάταξης χαρακτηριστικών ως προς τη σημαντικότητα

Για να μπορέσουμε να παρακολουθήσουμε την απόδοση του μοντέλου κατά την διάρκεια της εκπαίδευσης επιλέξαμε δύο μετρικές την ακρίβεια και το log loss. Η **ακρίβεια** μετρά το ποσοστό των σωστά προβλεπόμενων κλάσεων. Στο παρακάτω αριστερό σχήμα φαίνεται η καμπύλη της ακρίβειας όπως διαμορφώνεται κατά την εκπαίδευση και κατά τον έλεγχο. Ο



Σχήμα 4.21: Καμπύλες ακριβειας και log loss κατά τη διάρκεια της εκπαίδευσης και ελέγχου

άξονας των x δείχνει τον αριθμό των επαναλήψεων (ή τον αριθμό των δέντρων που προστίθενται στο σύνολο) και ο άξονας των y την ακρίβεια. Από την καμπύλη εκμάθησης, μπορούμε να δούμε ότι η απόδοση του μοντέλου στο σύνολο δεδομένων εκπαίδευσης (μπλε γραμμή) είναι καλύτερη ή έχει μεγαλύτερη ακρίβεια από την απόδοση του μοντέλου στο σύνολο δεδομένων ελέγχου (πορτοκαλί γραμμή), όπως θα περίμενε κανείς γενικά. Μετά από αρκετές επαναλήψεις το μοντέλο μαθαίνει πάρα πολύ καλά τα δεδομένα εκπαίδευσης και φτάνει στο απόλυτο (1.0) αλλά δεν γενικεύει το ίδιο καλά σε άγνωστα δεδομένα (δεδομένα ελέγχου) με αποτέλεσμα να παρατηρείται κάποιου βαθμού υπερπροσαρμογή (overfitting).

Από την καμπύλη του **log loss** κατά την εκπαίδευση γίνεται φανερό ότι το μοντέλο μαθαίνει τέλεια από τα δεδομένα εισόδου και μειώνει σταδιακά το log loss. Η αντίστοιχη καμπύλη στο σώμα ελέγχου κατεβαίνει και ακολουθεί την καμπύλη της εκπαίδευσης. Σταδιακά μειώνει το log loss και στο σώμα ελέγχου. Όμως, μετά από περίπου 100 rounds παρατηρείται μια σταθεροποίηση και μικρή άνοδος στην καμπύλη σε σχέση με την καμπύλη της εκπαίδευσης. Το γεγονός αυτό υποδεικνύει ότι υπάρχει κάποιου βαθμού υπερπροσαρμογή του μοντέλου, διοτί έμαθε τέλεια στα γνωστά δεδομένα αλλά δεν γενικεύει το ίδιο στα άγνωστα.

4.5.3 Support Vector Machine (SVM)

Κατά τον ίδιο τρόπο εφαρμόστηκε και το SVM μοντέλο, με δεδομένα εισόδου τον πίνακα με τα 14 γλωσσικά χαρακτηριστικά που είχαν εξαχθεί από τα κείμενα εισόδου.

Η κανονικοποίηση είναι συχνά πολύ σημαντική για τα SVM μοντέλα, ειδικά εάν τα χαρακτηριστικά έχουν μεγάλες διαφορές μεταξύ τους. Όλες οι αριθμητικές τιμές αντιστοιχίζονται με τιμές οι οποίες κυμαίνονται εντός προκαθορισμένου εύρους και με βάση κάποιον γραμμικό μετασχηματισμό. Η διαδικασία αυτή πριν από τη χρήση των χαρακτηριστικών σε μια μηχανή υποστήριξης διανυσμάτων (SVM) είναι ένα συνηθισμένο βήμα προεπεξεργασίας. Τα SVM είναι ευαίσθητα στην κλίμακα των χαρακτηριστικών και η κανονικοποίηση βοηθά να διασφαλιστεί ότι όλα τα χαρακτηριστικά συμβάλλουν εξίσου στη διαδικασία λήψης αποφάσεων του μοντέλου. Για την κανονικοποίηση των δεδομένων χρησιμοποιήθηκε η **StandardScaler** από το scikit-learn.

Όπως και στο παραπάνω μοντέλο, χρησιμοποιήθηκε το **GridSearchCV** και βρέθηκαν οι βέλτιστες υπερπαράμετροι για τον SVM στο συγκεκριμένο πρόβλημα ταξινόμησης κειμένων.

```
param_grid = {
    'C': [0.1, 1, 10],                          # Regularization parameter
    'gamma': ['scale', 'auto', 0.001, 0.01, 0.1], # Kernel coefficient ('scale' uses 1 / (n_features * X.var()), 'auto' uses 1 / n_features)
    'kernel': ['linear', 'rbf', 'poly'],        # Kernel type
    'degree': [3, 4, 5],                        # Degree of the polynomial kernel (for 'poly' kernel)
    'coef0': [1.0, 0.0]                         # Independent term in kernel function ('poly' and 'sigmoid' kernels)
}
```

Σχήμα 4.22: Αναζήτηση βέλτιστων υπερπαραμέτρων (SVM)

```
Best Hyperparameters: {'C': 10, 'coef0': 1.0, 'degree': 3, 'gamma': 0.01, 'kernel': 'poly'}
Accuracy on the test set: 0.8983
```

Σχήμα 4.23: Βέλτιστες υπερπαράμετροι (SVM)

Από τα αποτελέσματα της εκπαίδευσης παρατηρούμε, παρατηρούμε ότι η ακρίβεια που σημειώνεται στα δεδομένα εκπαίδευσης και ελέγχου είναι πολύ κοντά. Και πάλι η κλάση 0 που αντιπροσωπεύει το επίπεδο A υπερέχει σε όλες τις μετρικές. Το μοντέλο SVM μαθαίνει να ξεχωρίζει με πολύ μεγάλη ακρίβεια τα κείμενα αυτού του επιπέδου. Η απόδοση είναι ίδια τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα ελέγχου. Αμέσως μετά, ακολουθεί το

επίπεδο Γ και στη συνέχεια το επίπεδο Β το οποίο φαίνεται να δυσκολεύει περισσότερο από τα άλλα επίπεδα το μοντέλο.

```
Accuracy: 0.8785310734463276
Confusion Matrix (Training Data):
[[295  10   1]
 [ 14 148  25]
 [  3  33 179]]
Classification Report (Training Data):
              precision    recall  f1-score   support

           0       0.95      0.96      0.95       306
           1       0.77      0.79      0.78       187
           2       0.87      0.83      0.85       215

    accuracy                           0.88       708
   macro avg       0.86      0.86      0.86       708
weighted avg       0.88      0.88      0.88       708

Accuracy: 0.8983050847457628
Confusion Matrix (Test Data):
[[79  3  0]
 [ 4 45  5]
 [ 0  6 35]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.96        82
           1       0.83      0.83      0.83        54
           2       0.88      0.85      0.86        41

    accuracy                           0.90       177
   macro avg       0.89      0.88      0.89       177
weighted avg       0.90      0.90      0.90       177
```

Σχήμα 4.24: SVM classification reports και confusion matrices (train & test sets)

Η καμπύλη ROC σχετίζεται κυρίως με προβλήματα δυαδικής ταξινόμησης. Η καμπύλη δημιουργείται σχεδιάζοντας το ποσοστό των πραγματικών θετικών στον άξονα y έναντι του ποσοστού των ψευδώς θετικών στον άξονα x. Στην περίπτωση μας για να σχηματιστεί η γραφική παράσταση της καμπύλης ROC για την ταξινόμηση πολλών κλάσεων, χρησιμοποιείται η στρατηγική one-vs-rest (OvR). Χρησιμοποιείται η **OneVsRestClassifier,** που είναι μέρος του **sklearn.multiclass**, που μετατρέπει ένα πρόβλημα ταξινόμησης πολλών κλάσεων σε πολλαπλά προβλήματα δυαδικής ταξινόμησης, καθένα από τα οποία διακρίνει τη μία κλάση έναντι των υπολοίπων.

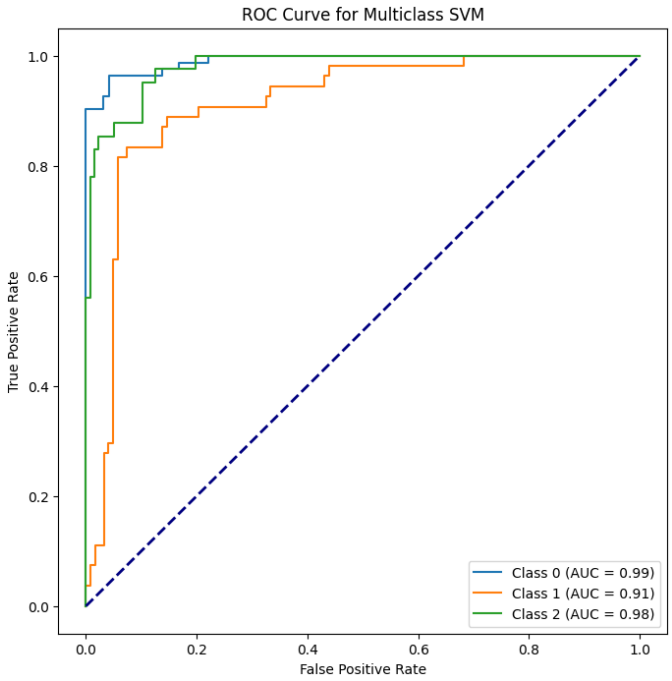
Παρακάτω φαίνεται το διάγραμμα ROC και για τις τρεις κλάσεις. Για να μπορέσουμε να κατανοήσουμε καλύτερα το διάγραμμα, θα αναφερθούμε στην έννοια της AUC η οποία είναι η περιοχή κάτω από την καμπύλη ROC και απεικονίζεται στο διάγραμμα.

Γενικά, οι τιμές AUC κυμαίνονται από 0 έως 1, όπου εάν:

- AUC = 0,5: Το μοντέλο δεν έχει διακριτική ισχύ και η απόδοσή του ισοδυναμεί με τυχαία εικασία.
- AUC > 0,5: Το μοντέλο αποδίδει καλύτερα από την τυχαία εικασία. Όσο υψηλότερη είναι η AUC, τόσο καλύτερη είναι η ικανότητα του μοντέλου να διακρίνει μεταξύ θετικών και αρνητικών περιπτώσεων.

- AUC = 1,0: Τέλεια ταξινόμηση. Το μοντέλο επιτυγχάνει πραγματικό θετικό ποσοστό 1 (ευαισθησία) και ψευδώς θετικό ποσοστό 0 (ειδικότητα) για όλες τις τιμές κατωφλίου.

Στο διάγραμμα φαίνεται ότι όλες οι καμπύλες έχουν καλή απόδοση γιατι είναι μακριά από τη διαγώνια γραμμή που αντιστοιχεί στην τυχαία ταξινόμηση. Η καμπύλη που απεικονίζει το επίπεδο A πλησιάζει περισσότερο στην πάνω αριστερή γωνία (στο 1) και έχει τα περισσότερα σωστά ταξινομημένα κείμενα και λιγότερα λάθος ταξινομημένα στο επίπεδο αυτό.



Σχήμα 4.25: Καμπύλη ROC (SVM)

4.5.4 Logistic Regration

Με τον ίδιο τρόπο υπολογίζονται οι βέλτιστες υπερπαράμετροι για τον αλγόριθμο Logistic Regration και φαίνονται στο παρακάτω σχήμα:

```
Best Parameters: {'C': 10, 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
```

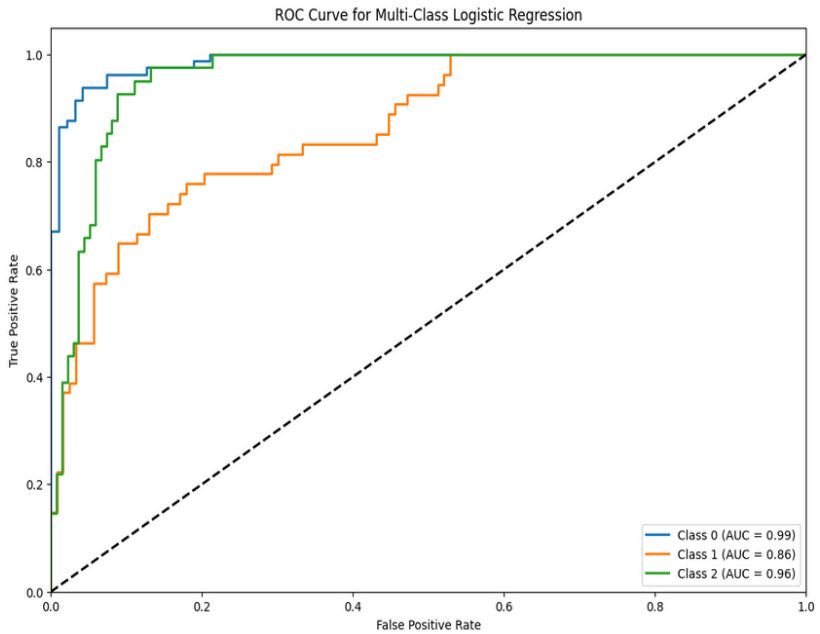
Σχήμα 4.26: Βέλτιστες υπερπαράμετροι (Logistic Regration)

Παρόμοια αποτελέσματα παίρνουμε, εφαρμόζοντας και τον αλγόριθμο Logistic Regration με το επίπεδο A να έχει τη μεγαλύτερη ακρίβεια, όπως φαίνονται στο classification report και confusion matrix.

```
Logistic Regration Train Accuracy: 0.8573446327683616
Confusion Matrix:
[[286  19   1]
 [ 11 146  30]
 [  3  37 175]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       306
           1       0.72      0.78      0.75       187
           2       0.85      0.81      0.83       215

    accuracy                           0.86       708
   macro avg       0.84      0.84      0.84       708
weighted avg       0.86      0.86      0.86       708

ogistic Regration Test Accuracy: 0.8757062146892656
Confusion Matrix:
[[75  7  0]
 [ 3 45  6]
 [ 0  6 35]]
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.91      0.94        82
           1       0.78      0.83      0.80        54
           2       0.85      0.85      0.85        41

    accuracy                           0.88       177
   macro avg       0.86      0.87      0.86       177
weighted avg       0.88      0.88      0.88       177
```

Σχήμα 4.27: Logistic Regration classification reports και confusion matrices (train & test sets)

Οι καμπύλες ROC για την κάθε κλάση φαίνονται στο παρακάτω διάγραμμα με την κλάση 1 (επίπεδο B) να είναι η πιο απομακρυσμένη καμπύλη από την αριστερή γωνία.



Σχήμα 4.28: Κσμπύλη ROC (Logistic Regration)

4.5.5 Multi Layer Perceptron

Ένα MLP (Multi Layer Perceptron) είναι ένας τύπος τεχνητού νευρωνικού δικτύου που αποτελείται από πολλαπλά στρώματα, συμπεριλαμβανομένου ενός στρώματος εισόδου, ενός ή περισσότερων κρυφών στρωμάτων και ενός στρώματος εξόδου. Όταν αναφερόμαστε σε έναν "ταξινομητή MLP με 1 στρώμα", εννοούμε ένα δίκτυο που έχει στρώμα εισόδου, ένα κρυφό στρώμα και στρώμα εξόδου. Στην πράξη, είναι σύνηθες να ξεκινήσει κανείς με ένα layer και σταδιακά να αυξήσει την πολυπλοκότητα του μοντέλου προσθέτοντας επιπλέον κρυφά layer εάν χρειάζονται.

Ο συγκεκριμένος MLPClassifier που χρησιμοποιήθηκε έχει: ένα κρυφό στρώμα με 100 νευρώνες, οι οποίοι έχουν τη "relu" συνάρτηση ενεργοποίησης για να αποφύγουμε το vanishing gradients. Οι νευρώνες στην έξοδο έχουν συνάρτηση ενεργοποίησης "softmax" διότι έχουμε πρόβλημα ταξινόμησης τριών κλάσεων.

```
# Define the parameter grid for grid search
param_grid = {
    'hidden_layer_sizes': [(50,), (100,), (50, 50)],
    'activation': ['relu', 'tanh', 'logistic'],
    'solver': ['adam', 'sgd', 'lbfgs'],
    'alpha': [0.0001, 0.001, 0.01],
    'learning_rate': ['constant', 'invscaling', 'adaptive'],
    'max_iter': [100, 200, 300],
    'batch_size': [32, 64, 128]
}
```

Σχήμα 4.29: Αναζήτηση βέλτιστων υπερπαραμέτρων (MLP)

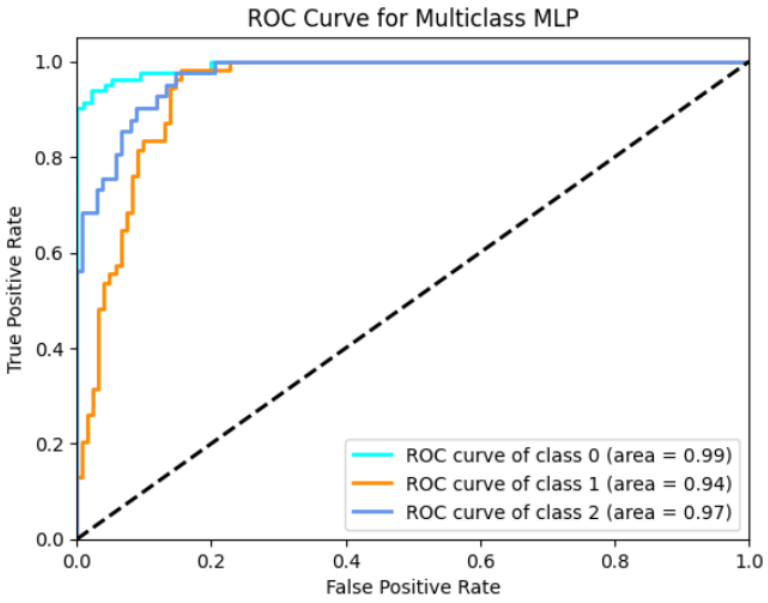
```
Best Parameters: {'activation': 'tanh', 'alpha': 0.01, 'batch_size': 128, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'max_iter': 200, 'solver': 'adam'}
```

Σχήμα 4.30: Βέλτιστες υπερπαράμετροι (MLP)

Όπως και στα παραπάνω μοντέλα, πραργματοποιήθηκε η επιλογή των βέλτιστων παραμέτρων για την εκπαίδευση του μοντέλου MLP. Όπως στο Logistic Regration αλλά και στο SVM, η ακρίβεια στο σώμα εκπαίδευσης και ελέγχου στο MLP μοντέλο έχει παρόμοιες τιμές κοντά στο 88%. Το μοντέλο MLP δείχνει να δυσκολεύεται να αποτυπώσει μοτίβα για την ταξινόμηση του B επιπέδου γλωσσομάθειας όπου σημειώνει f1-score 0.81, χαμηλότερο από τα άλλα επίπεδα.

```
MLP Test Accuracy: 0.8898305084745762
Confusion Matrix:
[[297   8   1]
 [ 13 148  26]
 [  3  27 185]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       306
           1       0.81      0.79      0.80       187
           2       0.87      0.86      0.87       215

    accuracy                           0.89       708
   macro avg       0.88      0.87      0.88       708
weighted avg       0.89      0.89      0.89       708

MLP Test Accuracy: 0.8870056497175142
Confusion Matrix:
[[78  4  0]
 [ 4 44  6]
 [ 0  6 35]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95        82
           1       0.81      0.81      0.81        54
           2       0.85      0.85      0.85        41

    accuracy                           0.89       177
   macro avg       0.87      0.87      0.87       177
weighted avg       0.89      0.89      0.89       177
```

Σχήμα 4.31: MLP Classification reports και confusion matrices (train & test sets)



Σχήμα 4.32: Καμπύλη ROC (MLP)

		Training	Testing
MODELS	**XGBoost**	**1.0**	0.8813
	SVM	0.8785	**0.8983**
	Logistic Regration	0.8573	0.8757
	MLP	0.8898	0.8870

Πίνακας 4.3: Μετρική Accuracy με split train 80% / test 20%

4.6 Word2vec

Στα πλαίσια της διπλωματικής δοκιμάστηκε και η προσέγγιση με τα word embeddings. Συγκεκριμένα, χρησιμοποιήθηκε το Word2Vec μοντέλο με τρείς τρόπους: χωρίς προ-εκπαιδευμένες ενσωματώσεις, με προ-εκπαιδευμένες ενσωματώσεις λέξεων και τέλος με προ-εκπαιδευμένες ενσωματώσεις μαζί με τα 14 γλωσσικά χαρακτηριστικά.

Αρχικά, δοκιμάστηκε το μοντέλο Word2Vec χωρίς προ-εκπαιδευμένες ενσωματώσεις λέξεων πάνω στο δικό μας μικρό σύνολο δεδομένων. Όπως ήταν αναμενόμενο, τα αποτελέσματα ήταν αντίστοιχα με έναν τυχαίο ταξινομητή. Το Word2Vec είναι ένας ισχυρός αλγόριθμος για την εκμάθηση ενσωματώσεων λέξεων όταν εκπαιδεύεται στα μεγάλα σώματα κειμένου καταγράφοντας σημασιολογικές σχέσεις μεταξύ των λέξεων. Ωστόσο, όταν εφαρμόζεται απευθείας σε μια εργασία ταξινόμησης, οι ενσωματώσεις του Word2Vec φαίνεται πως δεν αποτυπώνουν σημαντικά χαρακτηριστικά που σχετίζονται με το πρόβλημα ταξινόμησης. Ως αποτέλεσμα, ο ταξινομητής μπορεί να δυσκολεύεται να διακρίνει μοτίβα στα δεδομένα, οδηγώντας σε κακή απόδοση που θυμίζει τυχαία εικασία.

Επομένως, για να βελτιώθει η απόδοση, χρησιμοποιήθηκε το προ-εκπαιδευμένο μοντέλο *"cc.el.300.vec"* στην ελληνική γλώσσα. Το μοντέλο αυτό έχει εκπαιδευτεί στο Common Crawl και Wikipedia χρησιμοποιώντας τον αλγόριθμο fastText. Συγκεκριμένα, εκπαιδεύτηκε χρησιμοποιώντας την τεχνική CBOW με τα βάρη-θέσης, με το διάνυσμα κάθε λέξης να έχει διάσταση 300, με μέγεθος παραθύρου 5, με n-grams μήκους 5 χαρακτήρων και 10 αρνητικά παραδείγματα. [44]

Η συνάρτηση *gensim.models.KeyedVectors.load_word2vec_format* της βιβλιοθήκης Gensim χρησιμοποιείται για τη φόρτωση των προ-εκπαιδευμένων ενσωματώσεων σε μορφή Word2Vec, επιτρέποντάς να τις χρησιμοποιήσουμε στη συνέχεια στη ταξινόμηση των κειμένων του συνόλου δεδομένων μας. Χρησιμοποιούμε το όρισμα limit με τιμή 2000000 η οποία είναι και προεπιλεγμένη για να δηλώσουμε τον μέγιστο αριθμό λέξεων που θα φορτωθούν από το προ-εκπαιδευμένο μοντέλο.

Τα αποτελέσματα της εκπαίδευσης του προ-εκπαιδευμένου Word2Vec ήταν σαφώς καλύτερα από το σκέτο Word2Vec, όπως φαίνονται στον πίνακα 4.4. Τέλος, δοκιμάστηκε και ο συνδυασμός των word embeddings μαζί με τα γλωσσικά χαρακτηριστικά. Ο συνδυασμός αυτός βελτίωσε την απόδοση του κάθε μοντέλου με το MLP μοντέλο να υπερισχύει με το testing accuracy 0.88.

MODELS		Random Forest	XGBoost	Logistic Regration	SVM	MLP
	Word2vec	0.51	0.53	**0.55**	0.53	0.54
	Pre-trained Word2vec	0.73	**0.77**	0.68	0.62	0.73
	Pre-trained Word2vec + numerical features	0.83	0.854	0.84	0.75	**0.88**

Πίνακας 4.4: Μετρική accuracy στο testing set (με split train 80%/ test 20%)

4.7 BERT

Η αμφίδρομη μοντελοποίηση, οι προ-εκπαιδευμένες αναπαραστάσεις λέξεων, οι ενσωματώσεις με βάση τα συμφραζόμενα, η ικανότητα αποτύπωσης εξαρτήσεων μεγάλης εμβέλειας και η κορυφαία απόδοση του BERT τον καθιστούν ένα εξαιρετικά αποτελεσματικό και ευέλικτο εργαλείο για προβλήματα ταξινόμησης κειμένων. Θα ήταν πολύ χρήσιμο να δοκιμαστεί η προσέγγιση του BERT στα πλαίσια της διπλωματικής. Γι'αυτό τον λόγο, αναπτύχθηκαν τα μοντέλα **BertClassifier** που χρησιμοποιεί τις προ-εκπαιδευμένες ενσωματώσεις BERT και **CustomBertClassifier** που χρησιμοποιεί τον συνδυασμό των ενσωματώσεων BERT μαζί με τα 14 γλωσσικά χαρακτηριστικά (handcrafted features).

4.7.1 BertClassifier

Παρακάτω παρουσιάζονται τα επιμέρους στοιχεία της αρχιτεκτονικής του μοντέλου BertClassifier:

- **Bert Model (self.bert):**

  Χρησιμοποιεί το προ-εκπαιδευμένο μοντέλο BERT ("nlpaueb/bert-base-greek-uncased-v1") ως εργαλείο εξαγωγής χαρακτηριστικών. Το μοντέλο αυτό αναφέρεται σε ένα προ-εκπαιδευμένο μοντέλο BERT για την ελληνική γλώσσα που παρέχεται από το Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών (ΕΚΠΑ) και το Πανεπιστήμιο του Εδιμβούργου (UEB). Βασίστηκε στη βασική αρχιτεκτονική και εκπαιδεύτηκε σε πεζά ελληνικά κείμενα και αποτελέι την πρώτη έκδοση του μοντέλου. Η συνάρτηση AutoModel.from_pretrained φορτώνει αυτό το προ-εκπαιδευμένο μοντέλο BERT.

- **Dropout Layer (self.dropout):**

  Εφαρμόζεται ένα στρώμα dropout μετά την έξοδο του BERT για να μηδενιστεί τυχαία το 20% των μονάδων εισόδου στο στρώμα αυτό κατά τη διάρκεια της εκπαίδευσης. Αυτό βοηθά στην αποφυγή υπερβολικής προσαρμογής (overfitting). Δοκιμάστηκαν διάφορες τιμές του dropout (0.1, 0.2, 0.3, 0.5) και τα αποτελέσματα έδειξαν ότι η τιμή 0.2 οδήγησε σε βέλτιστη απόδοση του μοντέλου.

- **Linear Layer (self.linear):**

  Η έξοδος του dropout layer τροφοδοτείται σε ένα γραμμικό (πλήρως συνδεδεμένο) στρώμα. Το γραμμικό στρώμα έχει μέγεθος εισόδου 768 (το μέγεθος της εξόδου του BERT) και μέγεθος εξόδου 3, υποδεικνύοντας ότι έχει σχεδιαστεί για μια εργασία ταξινόμησης με τρεις κλάσεις.

- **ReLU Activation (self.relu):**

  Μετά τον γραμμικό μετασχηματισμό εφαρμόζεται μια συνάρτηση ενεργοποίησης ReLU. Το ReLU εισάγει τη μη γραμμικότητα στο μοντέλο.

Στο BERT και σε παρόμοια μοντέλα που βασίζονται σε μετασχηματιστές, η «ομαδοποιημένη έξοδος» (pooled output) είναι η συγκεντρωτική αναπαράσταση ή περίληψη ολόκληρης της ακολουθίας εισόδου που λαμβάνεται από το τελικό στρώμα του μοντέλου μετασχηματιστή. Προέρχεται από το ειδικό διακριτικό [CLS] (token ταξινόμησης) που προσαρτάται στην ακολουθία εισόδου. Η pooled_output είναι ένα διάνυσμα σταθερού μεγέθους και χρησιμοποιείται συχνά ως είσοδος σε εργασίες ταξινόμησης. Αυτή η ομαδοποιημένη έξοδος στη συνέχεια περνά μέσα από πρόσθετα επίπεδα (π.χ. γραμμικό επίπεδο και ενεργοποίηση) στο μοντέλο.

Κατά την διαδικασία του fine-tuning, χρησιμοποιήθηκε η συνάρτηση **get_linear_schedule_with_warmup** της βιβλιοθήκης transformers. Η συνάρτηση αυτή χρησιμοποιείται συνήθως για τον προγραμματισμό ρυθμών εκμάθησης με προθέρμανση στο πλαίσιο εκπαίδευσης μοντέλων BERT ή παρόμοιων μοντέλων που βασίζονται σε μετασχηματιστές. Η ιδέα πίσω από την προθέρμανση είναι να αυξηθεί σταδιακά ο ρυθμός μάθησης στην αρχή της εκπαίδευσης. Μετά τη φάση της εκπαίδευσης, ο ρυθμός εκμάθησης πέφτει για να τελειοποιήσει τη διαδικασία εκπαίδευσης. Αυτή είναι μια συνήθης πρακτική που εφαρμόζεται κατά την εκπαίδευση νευρωνικών δικτύων, ειδικά μοντέλων μετασχηματιστών όπως το BERT. Η σταδιακή αύξηση του ρυθμού μάθησης στην αρχή βοηθά το μοντέλο να συγκλίνει πιο αποτελεσματικά και το αποτρέπει από το να κάνει υπερβολικά μεγάλα βήματα στον χώρο των παραμέτρων νωρίς στην εκπαίδευση.

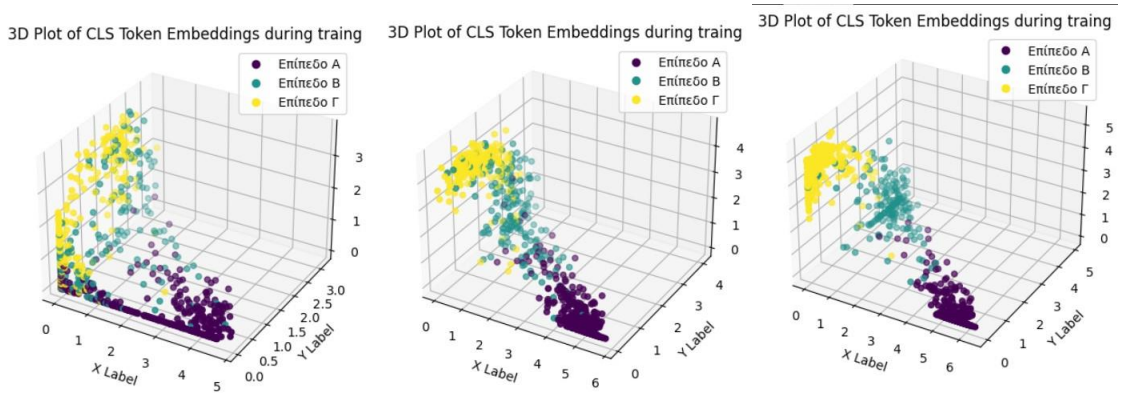
Στην περίπτωση μας, ο ρυθμός εκμάθησης ξεκινάει από το 0 και αυξάνεται σταδιακά κατά τη φάση της προθέρμανσης, που αποτελεί το 5% των συνολικών βημάτων (total_steps=89), μέχρι να φτάσει το επιθυμητό ρυθμό εκπαίδευσης που είναι 3e-5.

Μετά τη ρύθμιση των παραμέτρων και της αρχιτεκτονικής του BertClassifier ξεκινάει η εκπαίδευση του μοντέλου. Από την καταγραφή των αποτελεσμάτων, συμπεραίνουμε ότι το μοντέλο μέσα σε 8 εποχές μαθαίνει τέλεια στο σώμα εκπαίδευσης φτάνοντας στο accuracy 1.0. Στο σώμα επικύρωσης παρατηρείται επίσης μια σταδιακή αύξηση μάθησης φτάνοντας στο val.accuracy 0.92. Μετα από 5 εποχές παρατηρείται ελαφριά αύξηση στο validation loss που θα μπορούσε να υποδηλώνει ότι το μοντέλο αρχίζει να υπερπροσαρμόζεται στα δεδομένα εκπαίδευσης. Η υπερπροσαρμογή συμβαίνει όταν το μοντέλο μαθαίνει να απομνημονεύει τα παραδείγματα εκπαίδευσης αντί να καταγράφει τα μοτίβα που γενικεύονται καλά σε νέα, άγνωστα δεδομένα. Τα αποτελέσματα αυτά αποτυπώνονται στο σχήμα 4.37.

Σχήμα 4.33: Αποτελέσματα εκπαίδευσης του BertClassifier

Μεγάλο ενδιαφέρον παρουσιάζει και η σχεδίαση των ενσωματώσεων Bert στο τρισδιάστατο χώρο. Όμως, η σχεδίαση ενσωματώσεων υψηλών διαστάσεων, όπως οι ενσωματώσεις BERT που έχουν συνήθως 768 διαστάσεις, σε ένα τρισδιάστατο χώρο περιλαμβάνει τεχνικές μείωσης διαστάσεων. Η μείωση διαστάσεων στοχεύει στην αποτύπωση της βασικής δομής των δεδομένων υψηλών διαστάσεων σε έναν χώρο χαμηλότερης διάστασης διατηρώντας παράλληλα τα εγγενή χαρακτηριστικά του όσο το δυνατόν περισσότερο.

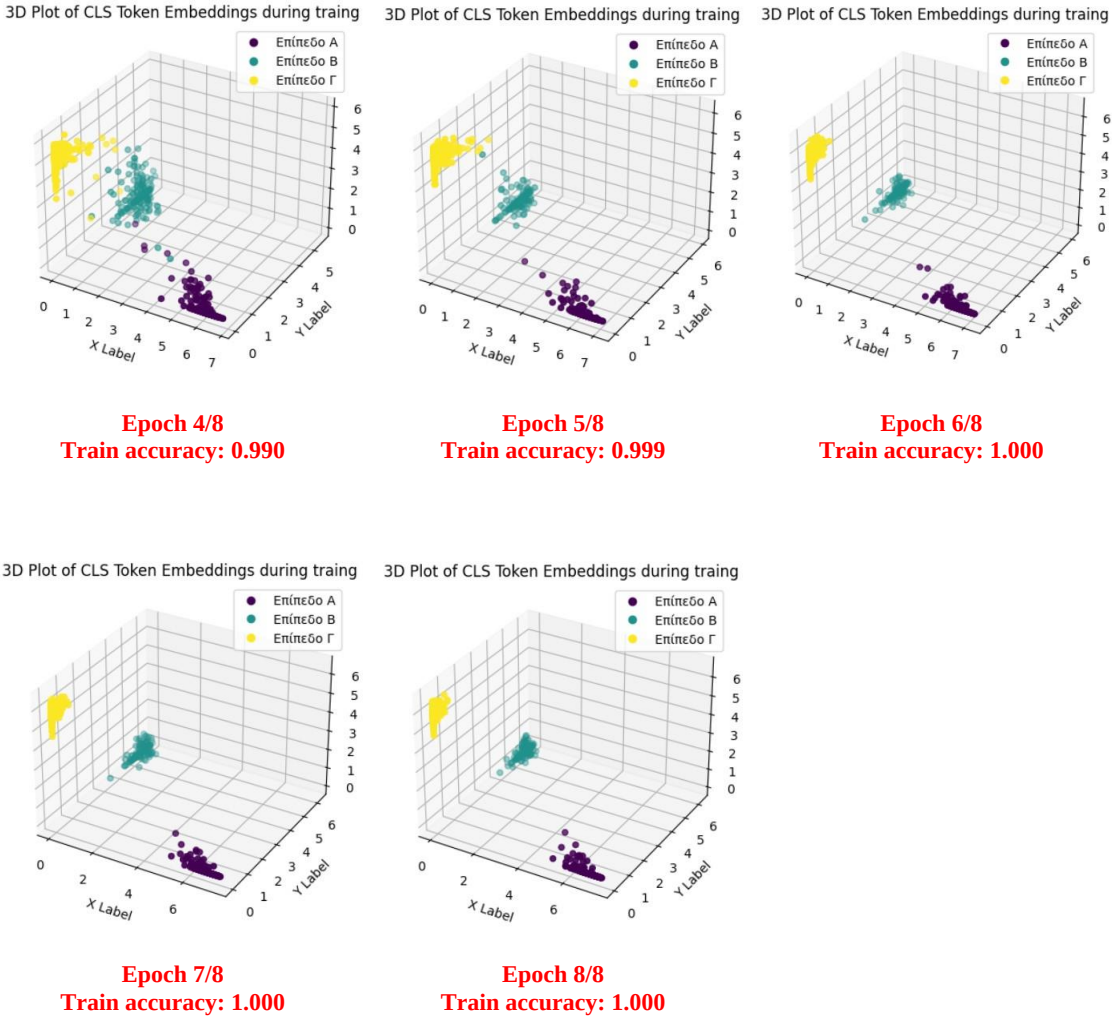
Μια κοινή τεχνική για τη μείωση διαστάσεων που χρησιμοποιήθηκε στη διπλωματική είναι η ανάλυση κύριων συνιστωσών (PCA). Το PCA προσδιορίζει τις κατευθύνσεις που καταγράφουν τη μεγαλύτερη διακύμανση στα δεδομένα και προβάλλει τα δεδομένα σε έναν υποχώρο χαμηλότερης διάστασης που ορίζεται από αυτά τα στοιχεία. Αυτή είναι μια συνήθης πρακτική που εφαρμόζεται κατά την εκπαίδευση νευρωνικών δικτύων, ειδικά μοντέλων μετασχηματιστών όπως το BERT. Επομένως, χρησιμοποιήθηκε η τεχνική PCA για να μειωθεί η διάσταση των ενσωματώσεων BERT από 768 σε 3 διαστάσεις και να οπτικοποιηθούν οι CLS ενσωματώσεις σε τρισδιάστατη γραφική παράσταση όπως φαίνονται παρακάτω:



Epochs 1/8
Train accuracy: 0.716

Epochs 2/8
Train accuracy: 0.880

Epochs 3/8
Train accuracy: 0.955



Epoch 4/8
Train accuracy: 0.990

Epoch 5/8
Train accuracy: 0.999

Epoch 6/8
Train accuracy: 1.000

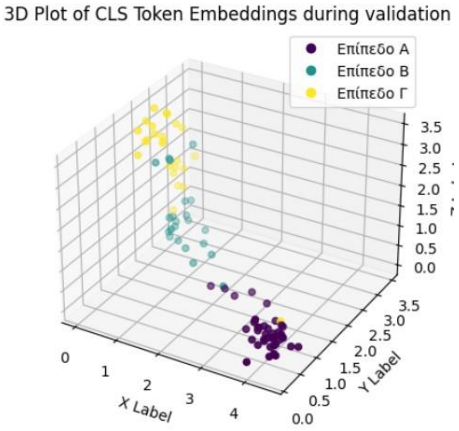
Epoch 7/8
Train accuracy: 1.000

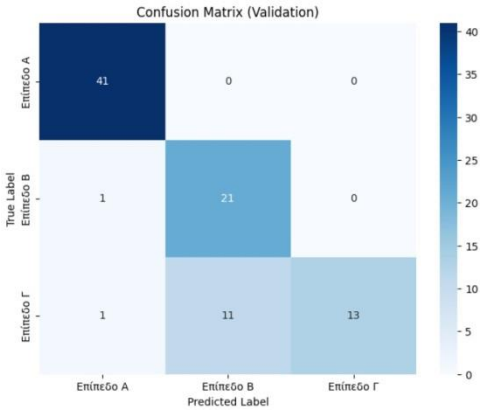
Epoch 8/8
Train accuracy: 1.000

Σχήμα 4.34: 3D αναπαράσταση των CLS embeddings κατά τη διάρκεια των εποχών εκπαίδευσης

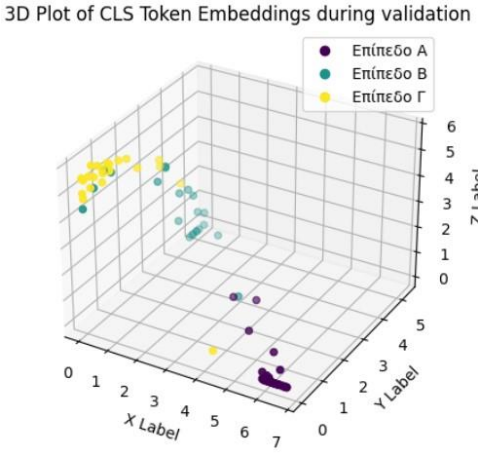
Στα παραπάνω 3διάστατα διαγράμματα μπορούμε να παρατηρήσουμε τα BERT embeddings όπως απεικονίζονται στο χώρο κατά τη διάρκεια κάθε εποχής, χρησιμοποιώντας τη συνάρτηση scatter του matplotlib. Έχουμε ένα τρισδιάστατο σχέδιο διασποράς όπου κάθε σημείο αντιπροσωπεύει ένα κείμενο εισόδου (μια ενσωμάτωση) και το χρώμα κάθε σημείου αντιστοιχεί στην ετικέτα του. Αυτή η οπτικοποίηση μπορεί να βοηθήσει στην κατανόηση της κατανομής των ενσωματώσεων και του τρόπου με τον οποίο σχετίζονται με τις ετικέτες στο σύνολο δεδομένων.

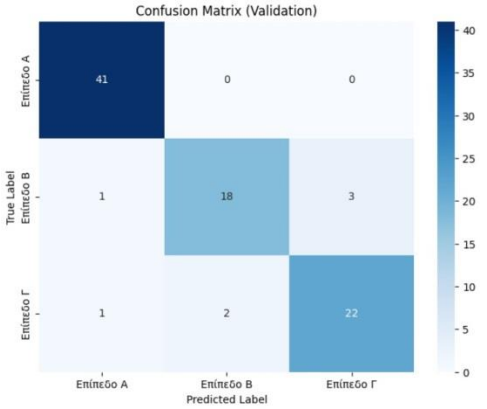
Τα παραπάνω διαγράμματα αντικατοπτρίζουν τη δυναμική φύση της μαθησιακής διαδικασίας, καθώς το μοντέλο προσαρμόζεται στην εργασία ταξινόμησης και στα δεδομένα στα οποία εκπαιδεύεται (fine-tuning). Οι ενσωματώσεις του διακριτικού CLS μετατοπίζονται στο χώρο των ενσωματώσεων καθώς το μοντέλο μαθαίνει σχετικές πληροφορίες ταξινόμησης και προσαρμόζεται στις παραμέτρους ταξινόμησης για να βελτιώσει την απόδοση του. Στο σχήμα 4.34 δίνονται αντίστοιχα και τα 3D διαγράμματα κατά τη διάρκεια του validation.



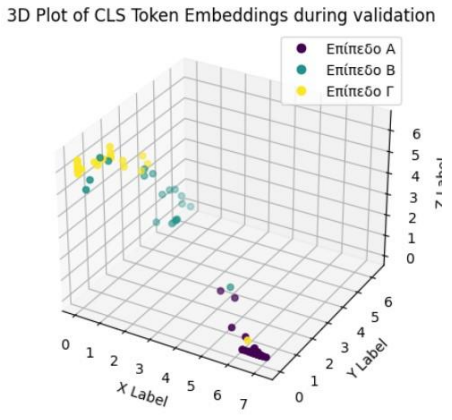


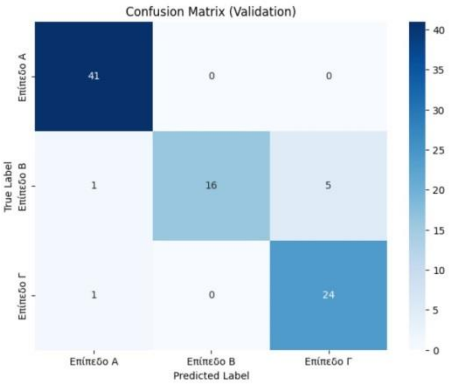
Epoch 1/8
Val accuracy: 0.852





Epoch 3/8
Val accuracy:0.920



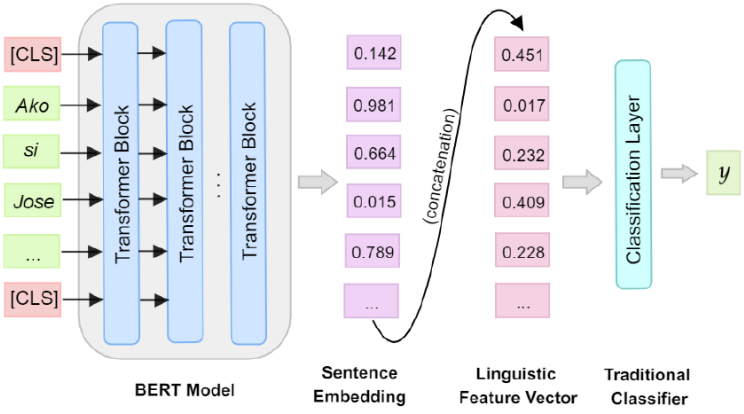


Epoch 8/8
Val accuracy: 0.920

Σχήμα 4.35: 3D αναπαράσταση των CLS embeddings και confusion matrices κατά την διάρκεια της επικύρωσης (validation)

4.7.2 CustomBertClassifier

Το CustomBertClassifier είναι παραλλαγή του BertClassifier με τη διαφορά ότι το τελικό Linear Layer αυτού του μοντέλου έχει ως είσοδο το BERT διάνυσμα διάστασης 768 συν τα 14 γλωσσικά χαρακτηριστικά (handcrafted features) όπου όλα τα 782 χαρακτηριστικά που προκύπτουν μετά το concatenation θα ληφθούν υπόψη για την ταξινόμηση του κάθε κειμένου. [45] Η διαδικασία αυτή είναι παρόμοια με το παραπάνω σχήμα:



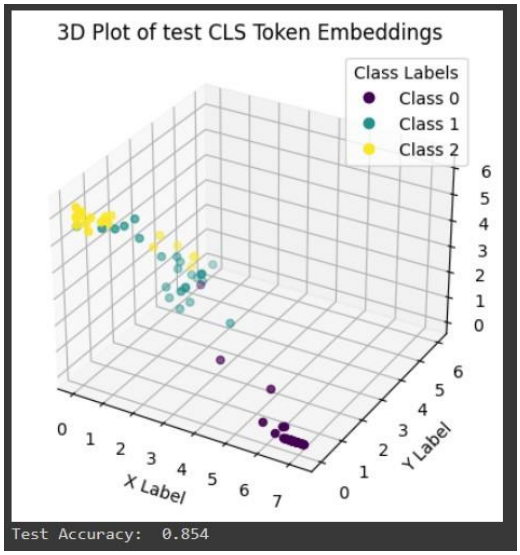
Σχήμα 4.36: Συνδυασμός των Bert embeddings μαζί με γλωσσικά χαρακτηριστικά (linguistic features)

Με την εκπαίδευση αυτού του μοντέλου παίρνουμε τα αποτελέσματα που δείχνει το σχήμα 4.38, τα οποία είναι παρόμοια με το BertClassifier αλλά δεν ξεπερνάν την απόδοση του BertClassifier.

```
BertClassifier + numerical features
Epoch 1/8, Learning Rate: 0.0
100%|██████████| 45/45 [00:57<00:00,  1.28s/it]
Epochs: 1 | Train Loss:  0.044               | Train Accuracy:  0.699              | Val Loss:  0.362              | Val Accuracy:  0.841
Epoch 2/8, Learning Rate: 2.647058823529412e-05
100%|██████████| 45/45 [00:58<00:00,  1.31s/it]
Epochs: 2 | Train Loss:  0.019               | Train Accuracy:  0.887              | Val Loss:  0.313              | Val Accuracy:  0.852
Epoch 3/8, Learning Rate: 2.26890756302521e-05
100%|██████████| 45/45 [00:59<00:00,  1.32s/it]
Epochs: 3 | Train Loss:  0.005               | Train Accuracy:  0.973              | Val Loss:  0.372              | Val Accuracy:  0.818
Epoch 4/8, Learning Rate: 1.8907563025210083e-05
100%|██████████| 45/45 [01:00<00:00,  1.34s/it]
Epochs: 4 | Train Loss:  0.002               | Train Accuracy:  0.993              | Val Loss:  0.372              | Val Accuracy:  0.864
Epoch 5/8, Learning Rate: 1.5126050420168067e-05
100%|██████████| 45/45 [01:00<00:00,  1.35s/it]
Epochs: 5 | Train Loss:  0.001               | Train Accuracy:  0.997              | Val Loss:  0.386              | Val Accuracy:  0.886
Epoch 6/8, Learning Rate: 1.134453781512605e-05
100%|██████████| 45/45 [01:00<00:00,  1.35s/it]
Epochs: 6 | Train Loss:  0.001               | Train Accuracy:  0.999              | Val Loss:  0.380              | Val Accuracy:  0.875
Epoch 7/8, Learning Rate: 7.563025210084033e-06
100%|██████████| 45/45 [01:00<00:00,  1.35s/it]
Epochs: 7 | Train Loss:  0.000               | Train Accuracy:  0.999              | Val Loss:  0.410              | Val Accuracy:  0.875
Epoch 8/8, Learning Rate: 3.7815126050420167e-06
100%|██████████| 45/45 [01:00<00:00,  1.35s/it]
Epochs: 8 | Train Loss:  0.000               | Train Accuracy:  0.999              | Val Loss:  0.407              | Val Accuracy:  0.875
```

Σχήμα 4.37: Αποτελέσματα εκπαίδευσης του CustomBertClassifier

Από το 3D διάγραμμα των CLS στο σώμα ελέγχου (testing set) παρατηρούμε ότι τα επίπεδα Β και Γ βρίσκονται πολύ κοντά στο χώρο των ενσωματώσεων και δεν είναι τόσο ξεκάθαρη η διάκριση ανάμεσα στα δύο αυτά επίπεδα όσο είναι για το επίπεδο Α.



Σχήμα 4.38: 3D αναπαράσταση των CLS embeddings στο σώμα ελέγχου

	Train	Validation	Test
BertClassifier	1.0	0.92	0.876
CustomBertClassifier	0.99	0.87	0.85

Πίνακας 4.5: Μετρική accuracy με split train 80% / test 20%

Από τα αποτελέσματα που προκύπτουν μετά την εκπαίδευση των μοντέλων BertClassifier και CustomBertClassifier, συμπεραίνουμε ότι ο συνδυασμός των Bert embeddings μαζί με τα 14 γλωσσικά χαρακτηριστικά δεν έδωσε καλύτερα αποτελέσματα από τον BertClassifier που χρησιμοποιεί μόνο τα embeddings. Γενικά, ο BertClassifier σημείωσε πολύ καλά αποτελέσματα στο σύνολο εκπαίδευσης και επικύρωσης (training & validation set). Στο σώμα ελέγχου, το ποσοστό έφτασε στο 87,5% το οποίο είναι αξιοσημείωτο. Αν και δεν ξεπέρασε τα παραδοσιακά μοντέλα μηχανικής μάθησης, ο BertClassifier μπόρεσε να βρει και να αποτυπώσει μοτίβα σε ένα μικρό σύνολο δεδομένων που είχαμε και έφτασε να κάνει σχεδόν την τέλεια κατηγοριοποίηση για το επίπεδο Α και πολύ ικανοποιητική για επίπεδα Β και Γ. Σίγουρα θα πρέπει να εξεταστεί η απόδοση του μοντέλου εάν εκπαιδευτεί σε μεγαλύτερο σύνολο κειμένων(fine-tuning), όπου πιθανότατα θα έχει πολύ καλύτερη απόδοση γιατί θα μπορέσει να εντοπίσει τις λεπτομέρειες εκείνες που θα είναι χρήσιμες για την ταξινόμηση των κειμένων.

Κεφάλαιο 5ο Σύνοψη συμπερασμάτων

5.1 Συμπεράσματα

	XGBoost	**SVM**	**MLP**	**Logistic Regration**	**Random Forest**
Linguistic features	0.8813	**0.8983**	0.8870	0.8757	0.85
Word2vec embeddings	**0.77**	0.62	0.73	0.68	0.73
BERT embeddings	-	-	**0.876**	-	-
Word2vec combined features	0.854	0.75	**0.88**	0.84	0.83
BERT combined features	-	-	**0.85**	-	-

Πίνακας 5.1: Συγκεντρωτικός πίνακας απόδοσης μοντέλων με μετρική accuracy

Από τα αποτελέσματα που λάβαμε από όλα τα μοντέλα συμπεραίνουμε ότι η χρήση των 14 γλωσσικών χαρακτηριστικών στα παραδοσιακά μοντέλα ταξινόμησης είχαν την καλύτερη απόδοση από την προσέγγιση των embeddings του Word2Vec αλλά και του Bert. Το SVM μοντέλο ήταν αυτό που σημείωσε τη μεγαλύτερη ακρίβεια στο σώμα ελέγχου της τάξης 89.83% συγκριτικά με όλα τα μοντέλα.

Τα παραδοσιακά μοντέλα μηχανικής εκμάθησης όπως τα SVM, MLP, Random Forest και XGBoost συχνά δείχνουν καλή απόδοση με μικρά σύνολα δεδομένων σε σύγκριση με τα μοντέλα βαθιάς εκμάθησης. Το BERT, ως προ-εκπαιδευμένο μοντέλο βαθιάς μάθησης, συχνά απαιτεί ένα μεγάλο σύνολο δεδομένων για να μπορεί να καταγράψει περίπλοκα γλωσσικά μοτίβα. Συνιστάται η ύπαρξη τουλάχιστον μερικών χιλιάδων διαβαθμισμένων κειμένων για βέλτιστη απόδοση του μοντέλου. Το BERT έχει εκατομμύρια παραμέτρους και τείνει να επωφελείται από μεγαλύτερα σύνολα δεδομένων.

Επομένως, το γεγονός ότι τα μοντέλα Word2Vec και Bert δεν έδειξαν καλύτερα αποτελέσματα από τα παραδοσιακά πιθανόν να οφείλεται στο μικρό μέγεθος του συνόλου δεδομένων που είχε ως αποτέλεσμα τα μοντέλα να μην προλαβαίνουν να αποτυπώσουν τα σύνθετα μοτίβα που απαιτούνται για την ταξινόμηση των κειμένων σε επίπεδα γλωσσομάθειας.

5.2 Πιθανές επεκτάσεις και μελλοντικές προκλήσεις

- Ανάπτυξη εξειδικευμένων ελληνικών συνόλων δεδομένων κατάλληλα ταξινομημένων ως προς τον βαθμό δυσκολίας τους στα έξι επίπεδα γλωσσομάθειας βάσει του Κοινού Ευρωπαϊκού Πλαισίου Αναφοράς για τις Γλώσσες (CERF). Η ανάπτυξη μεγαλύτερων, ολοκληρωμένων και ισορροπημένων συνόλων δεδομένων είναι ζωτικής σημασίας. Με ένα μικρό σύνολο δεδομένων, όπως χρησιμοποιήσαμε στα πλαίσια της διπλωματικής, υπάρχει ο κίνδυνος υπερπροσαρμογής, όπου το μοντέλο απομνημονεύει τα παραδείγματα εκπαίδευσης αντί να μαθαίνει γενικά

γλωσσικά μοτίβα. Γενικά, χρειάζεται πειραματισμός σε διαφορετικά μεγέθη συνόλων δεδομένων για να μπορούμε πιο σωστά να παρακολουθήσουμε την απόδοση του μοντέλου BERT που σημειώνει πολύ καλα αποτελέσματα στα προβλήματα ταξινόμησης. [45]

- Θα μπορούσε να αναπτυχθεί μια ιστοσελίδα όπου ο χρήστης θα εισήγαγε τα κείμενα προς ταξινόμηση σε επίπεδα CERF.

- Να προστεθούν επιπλέον κλασικές μετρικές και στυλομετρικά χαρακτηριστικά όπως αυτά που αφορούν τη λεξιλογική ποικιλομορφία: entropy, gini, hapax legomena, h, RR και άλλα. [46]

Βιβλιογραφία

[1] W. H. Dubay, Smart Language: Readers, Readability and the Grading of Text, California, January 2007.

[2] O. L. L. Thomas R. Herzog, «Searching for Legibility» *Environment and Behavior,* pp. 459-477, 2003.

[3] Μ. Γιάγκου, «Σώματα Κειμένων και Γλωσσική Εκπαίδευση: Δυνατότητες Αξιοποίησης στη Διδασκαλία της Ελληνικής και Συγκρότηση Παιδαγωγικά Κατάλληλων Σωμάτων Κειμένων» 2009. [Ηλεκτρονικό]. Available: https://thesis.ekt.gr/thesisBookReader/id/24615?lang=el#page/1/mode/2up. [Πρόσβαση 14 02 2024].

[4] D. R. McCallum και J. L. Peterson, «Computer-based readability indexes» σε *Proceedings of the ACM '82 conference, 1982, 44-48.*, 1982.

[5] R. Flesch, «A New Readability Yardstick» *Journal of Applied Psychology, 32, 221-233,* 1948.

[6] Δ. Τζιμώκας και Μ. Ματθαιουδάκη, «Δείκτες αναγνωσιμότητας: Ζητήματα εφαρμογής και αξιοπιστίας» [Ηλεκτρονικό]. Available: https://ikee.lib.auth.gr/record/270005/files/Deiktes.pdf. [Πρόσβαση 2024 02 14].

[7] Α. Γαγάτσης, «Η αναγνωσιμότητα των σχολικών βιβλίων των μαθηματικών του Δημοτικού Σχολείου» *Σύγχρονη Εκπαίδευση 20,* pp. 40-48, 1985.

[8] K. e. al., «Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel» 1975. [Ηλεκτρονικό]. Available: https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1055&context=istlibrary. [Πρόσβαση 14 02 2024].

[9] W. Dubay, «The Principles of Readability» 2004.

[10] «Wikipedia: Gunning fog index» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Gunning_fog_index. [Πρόσβαση 14 02 2024].

[11] «Πιστοποίηση ελληνομάθειας» Κέντρο Ελληνικής Γλώσσας, 2014. [Ηλεκτρονικό]. Available: https://www.greek-language.gr/certification/readability/about.html. [Πρόσβαση 14 02 2024].

[12] K. Collins-Thompson και J. P. Callan, «A Language Modeling Approach to Predicting Reading Difficulty» σε *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston,

Massachusetts, USA, 2004.

[13] S. Schwarm και M. Ostendorf, «Reading Level Assessment Using Support Vector Machines and Statistical Language Models» σε *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 523–530*, Michigan, 2005.

[14] E. Pitler και A. Nenkova, «Revisiting Readability: A Unified Framework for Predicting Text Quality» σε *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 186–195*, Honolulu, Hawaii, 2008.

[15] Mohammadi και Khasteh, «Text as Environment: A Deep Reinforcement Learning Text Readability Assessment Model» 2023, v4.

[16] M. Azpiazu και M. S. Pera, «Multiattentive Recurrent Neural Network Architecture for Multilingual Readability Assessment» *Transactions of the Association for Computational Linguistics, Volume 7,* p. 421–436, 2019.

[17] M. Grandini, E. Bagli και G. Visani, «Metrics for Multi-Class Classification: an Overview, arXiv:2008.05756,» 2020. [Ηλεκτρονικό]. Available: https://arxiv.org/pdf/2008.05756.pdf. [Πρόσβαση 14 02 2024].

[18] Z. Vujovic, «Classification Model Evaluation Metrics» *International Journal of Advanced Computer Science and Applications,* pp. Volume 12. 599-606, 2021.

[19] Q. Li, H. Peng, L. JianXin και C. Xia, «A Survey on Text Classification: From Traditional to Deep Learning» *ACM Transactions on Intelligent Systems and Technology,* pp. 1-41, 2022.

[20] M. a. M. N. S. Hossin, «A review on evaluation metrics for data classification evaluations» *International journal of data mining & knowledge management process 5.2* , 2015.

[21] I. Michailidis, K. Diamantaras, S. Vasileiadis και Y. Frère, «Greek Named Entity Recognition using Support Vector Machines, Maximum Entropy and Onetime» σε *In Proceedings of the Fifth International Conference on Language Resources*, 2006.

[22] M. K. Dasari, «Machine Learning NLP Text Classification Algorithms and Models» 24 01 2022. [Ηλεκτρονικό]. Available: https://www.linkedin.com/pulse/machine-learning-nlp-text-classification-algorithms-models-dasari. [Πρόσβαση 14 02 2024].

[23] R. Gandhi, «Support Vector Machine — Introduction to Machine Learning Algorithms,» Towards Data Science, 2018. [Ηλεκτρονικό]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Πρόσβαση 14 02 2024].

[24] J. Brownlee, «Support Vector Machines for Machine Learning» 2016. [Ηλεκτρονικό]. Available: https://machinelearningmastery.com/support-vector-machines-for-machine-learning/. [Πρόσβαση 14 02 2024].

[25] S. Gunn, «Support Vector Machines for Classification and Regression» ISIS Technical Report, 1998. [Ηλεκτρονικό]. Available: https://svms.org/tutorials/Gunn1998.pdf. [Πρόσβαση 2024].

[26] P. Vadapalli, «What is an Ensemble Method» upGrad, [Ηλεκτρονικό]. Available: https://www.upgrad.com/blog/bagging-vs-boosting/. [Πρόσβαση 14 02 2024].

[27] T. Chen και C. Guestrin, «Xgboost: A scalable tree boosting system» σε *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).*, 2016.

[28] x. developers, «XGBoost Documentation» 2022. [Ηλεκτρονικό]. Available: https://xgboost.readthedocs.io/en/stable/tutorials/model.html. [Πρόσβαση 14 02 2024].

[29] M. Giatsoglou, . M. G. Vozalis και K. Diamantaras, «Sentiment analysis leveraging emotions and word embeddings» *Expert Systems with Applications,* pp. vol. 69, pp. 214-224, 2017.

[30] B. Chiu και S. Baker, «Word embeddings for biomedical natural language processing: A survey» *Language and Linguistics Compass,* 2020.

[31] «Continuous bag of words (CBOW) in NLP» GeeksforGeeks, [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/continuous-bag-of-words-cbow-in-nlp/.

[32] X. Rong, «word2vec parameter learning explained» *arXiv preprint, arXiv:1411.2738,* 2014.

[33] C. Maklin, «Word2Vec — Skip-Gram» Medium, 2022. [Ηλεκτρονικό]. Available: https://medium.com/@corymaklin/word2vec-skip-gram-904775613b4c. [Πρόσβαση 14 02 2024].

[34] «Encoders-Decoders, Sequence to Sequence Architecture» Medium, 2021. [Ηλεκτρονικό]. Available: https://medium.com/analytics-vidhya/encoders-decoders-sequence-to-sequence-architecture-5644efbb3392. [Πρόσβαση 2024].

[35] Δ. Παπαδόπουλος, «Γνωσιακές μηχανές με χρήση μεθόδων μηχανικής μάθησης» Εθνικό αρχείο διδακτορικών διατριβών, 2022. [Ηλεκτρονικό]. Available: https://freader.ekt.gr/eadd/index.php?doc=52039&lang=el. [Πρόσβαση 14 02 2024].

[36] D. Jurafsky και J. H. Martin , «Machine Translation and Encoder-Decoder Models» σε *Speech and Language Processing; drsft,* 2022.

[37] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & U, «Attention Is All You Need» σε *Advances in Neural Information Processing Systems, pp. 5998-6008.*, 2017.

[38] «Attention (machine learning)» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Attention_%28machine_learning%29. [Πρόσβαση 14 02 2024].

[39] J. Alammar, «The Illustrated Transformer» [Ηλεκτρονικό]. Available: https://jalammar.github.io/illustrated-transformer/. [Πρόσβαση 14 02 2024].

[40] J. e. a. Devlin, «Bert: Pre-training of deep bidirectional transformers for language understanding» *arXiv preprint arXiv:1810.04805,* 2018.

[41] J. Koutsikakis, . I. Chalkidis, P. Malakasiotis και I. Androutsopoulos, «GREEK-BERT: The Greeks visiting Sesame Street» σε *11th Hellenic Conference on Artificial Intelligence, Greece pp. 1-8.*, 2020.

[42] D. Jacob, i. C. Ming-We, L. Kenton και T. Kristina, «Bert model (uncased)» Hugging Face, [Ηλεκτρονικό]. Available: https://huggingface.co/bert-base-uncased. [Πρόσβαση 14 02 2024].

[43] I. Chalkidis, «GreekBert» Hugging Face, 2020. [Ηλεκτρονικό]. Available: https://huggingface.co/nlpaueb/bert-base-greek-uncased-v1. [Πρόσβαση 14 02 2024].

[44] «FastText: Library for efficient text classification and representation learning» Facebook Inc., [Ηλεκτρονικό]. Available: https://fasttext.cc/docs/en/crawl-vectors.html. [Πρόσβαση 14 02 2024].

[45] J. M. Imperial, «BERT Embeddings for Automatic Readability Assessment» σε *In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pages 611–618,* 2021.

[46] G. K. Mikros και R. Voskaki, «A Modern Greek readability tool: Development of evaluation methods» *Language and Text,* pp. pp.164-175, 2021.

Παράρτημα

Εύκολες λέξεις

ακόμη	έχει	μητέρας	ότι
αλλά	έχουμε	μαύρος	ούτε
άλλος	έχετε	μαύρου	όχι
άλλον	έχουν	μαύρε	παιδί
άλλου	ή	μαύροι	παιδιού
άλλους	ήμουν	μαύρων	παιδιά
άλλη	ήσουν	μαύρους	παιδιών
άλλης	ήταν	μαύρη	πάλι
άλλες	ήμασταν	μαύρης	πάω
άλλο	ήσασταν	μαύρες	πας
άλλα	θα	μαύρα	πάει
άλλων	θέλω	με	πάμε
αμέσως	θέλεις	μέρα	πάτε
αν	θέλει	μέρας	πάνε
άντρας	θέλουμε	μέρες	παν
άντρα	θέλετε	μερών	πηγαίνω
άντρες	θέλουν	ημέρα	πηγαίνεις
αντρών	ίσια	ημέρας	πηγαίνει
από	κάθε	ημέρες	πηγαίνουμε
αργά	και	ημερών	πηγαίνετε
αρέσει	καλημέρα	μερικοί	πηγαίνουν
αριστερά	καληνύχτα	μερικών	πήγαινε
άσπρος	καλησπέρα	μερικούς	πάντα
άσπρου	καλοκαίρι	μερικές	πάνω

άσπροι	καλοκαίρια	μερικά	παππούς
άσπρους	καλός	μεγάλος	παππού
άσπρη	καλού	μεγάλου	πάρα
άσπρης	καλό	μεγάλο	παρασκευή
άσπρες	καλέ	μεγάλοι	παρασκευής
άσπρο	καλοί	μεγάλων	πέμπτη
άσπρα	καλούς	μεγάλους	πέμπτης
άσπρων	καλή	μεγάλη	πέρσι
αύριο	καλής	μεγάλης	πιο
αυτός	καλές	μεγάλες	πίσω
αυτού	καλά	μεγάλα	ποιος
αυτόν	καλών	μέσα	ποιου
αυτοί	κανείς	μεσημέρι	ποιο
αυτούς	κανένας	μεσημέρια	ποιοι
αυτή	καμία	μετά	ποιων
αυτής	καμιά	μέχρι	ποιους
αυτές	καμίας	μη	ποια
αυτό	καμιάς	μην	ποιας
αυτά	κανένα	μικρός	ποιες
αυτών	κάνω	μικρού	πολλά
βράδυ	κάνεις	μικρό	πολύ
γεια	κάνει	μικροί	πόσο
για	κάνουμε	μικρών	πότε
γιαγιά	κάνετε	μικρούς	ποτέ
γιατί	κάνουν	μικρή	που
γιος	κάνε	μικρής	πού

γυναίκα	έκανα	μικρές	πράσινος
γυναίκας	έκανες	μικρά	πράσινου
γυναίκες	έκανε	μόλις	πράσινο
δασκάλα	κάναμε	μόνο	πράσινοι
δασκάλας	κάνατε	μπαμπάς	πράσινων
δασκάλες	έκαναν	μπαμπά	πράσινους
δεν	κάτι	πατέρας	πράσινη
δε	κάτω	πατέρα	πράσινης
δεξιά	κίτρινος	μπορώ	πράσινες
δευτέρα	κίτρινου	μπορείς	πράσινα
δευτέρας	κίτρινο	μπορεί	πρέπει
δευτέρες	κίτρινε	μπορούμε	πριν
δύο	κίτρινοι	μπορείτε	πρωί
εγώ	κίτρινους	μπορούν	πρώτα
εμένα	κίτρινη	μπόρεσα	πώς
με	κίτρινης	μπόρεσες	σάββατο
εδώ	κίτρινες	μπόρεσε	σαββάτου
είμαι	κίτρινα	μπορέσαμε	σάββατα
είσαι	κίτρινων	μπορέσατε	σε
είναι	κόκκινος	μπόρεσαν	σήμερα
είμαστε	κόκκινου	μπορέσω	σιγά
είστε	κόκκινο	μπορέσεις	σπίτι
εκεί	κόκκινε	μπορέσει	σπιτιού
εκείνος	κόκκινοι	μπορέσουμε	σπίτια
εκείνου	κόκκινη	μπορέσετε	στη
εκείνοι	κόκκινης	μπορέσουν	στην

εκείνους	κόκκινες	μπροστά	στο
εκείνη	κόκκινα	να	στον
εκείνης	κόκκινων	ναι	στα
εκείνες	κοντά	νύχτα	σχολείο
εκείνο	κόρη	νύχτας	σχολείου
εκείνα	κόρης	νύχτες	σχολεία
εκείνων	κόρες	νωρίς	σχολείων
εμείς	κυριακή	ξανά	τετάρτη
εμάς	κυριακής	ο	τετάρτης
μας	κυριακές	του	τι
ένας	κυρία	τον	τίποτα
ενός	κυρίας	οι	τίποτε
έναν	κυρίες	των	τότε
ένα	κύριος	τους	τρίτη
μία	κυρίου	η	τρίτης
μια	κύριοι	της	τώρα
μιας	λέω	την	υπάρχω
μίας	λες	τις	υπάρχεις
ένα	λέει	το	υπάρχει
εντάξει	λέμε	τα	υπάρχουμε
έξω	λέτε	όλος	υπάρχετε
εσύ	λένε	όλου	υπάρχουν
εσένα	λεν	όλο	φέτος
σε	λέγε	όλοι	χθες
σου	λίγο	όλων	χωρίς
εσείς	μαζί	όλους	ώρα

εσάς	μακριά	όλη	ώρας
έτσι	μαμά	όλης	ώρες
ευχαριστώ	μαμάς	όλες	ωραία
έχω	μητέρα	όλα	δυο
έχεις		όταν	

Ο προσδιορισμός των εύκολων λέξεων του κειμένου στη διπλωματική βασίστηκε στον παραπάνω πίνακα έτσι όπως είχε διαμορφωθεί το 2012 από το ΚΕΓ. Σε αυτό το σημείο να τονιστεί ότι αυτός ο πίνακας πιθανόν να έχει καταστεί παρωχημένος, δεδομένης της δυναμικής της γλώσσας να εισάγει νεολογισμούς και να παραγκωνίζει στο πεδίο της λήθης λέξεις ή σημασίες λέξεων. Αν θεωρήσουμε τη λέξη *γράμμα* οικεία, εξακολουθεί να είναι οικεία και κατανοητή στη φράση το *γράμμα του νόμου;* [3]

Προθήματα (Prefixes)

ανα	εισ	κάτ	συμ
ανά	είσ	κάθ	συγ
ανε	εκ	κυρα	συλ
ανέ	εξ	κερα	συρ
ανη	έκ	μαστρο	συσ
ανή	έξ	μετα	συ
Αϊ	εμπορο	μετ	σύν
αμφι	επαν	μεθ	σύμ
αμφί	επι	μετά	σύγ
αντι	επ	μέτ	σύλ
αντί	εφ	μέθ	σύρ
αντ	επί	μπαρμπα	σύσ
ανθ	έπ	ξε	σύ
απο	έφ	ξέ	υπερ
από	ευ	παρα	υπέρ
γρια	εύ	παρ	υπο
δια	θεια	παρά	υπ
διά	κατα	πάρ	υφ
δι	κατ	περι	υπό
δυσ	καθ	περί	ύπ
δύσ	κατά	συν	ύφ
			χατζη

Επιθήματα (Postfixes)

άδα	εία	ιανών	λικιών	τζίδικους
άδας	είας	ιανός	λόι	τζού
άδες	είες	ιανοί	λογιού	τζούς
άδων	ειδές	ιανούς	λόγια	τζούδες
αδάκι	ειδούς	ιάρα	λογιών	τζούδων
αδάκια	ειδή	ιάρας	λού	τήρας
άδι	ειδών	ιάρες	λούς	τήρα
αδιού	ειδής	ιαρών	λούδες	τήρες
άδια	ειδείς	ιάρης	λούδων	τήρων
αδιών	ειο	ιάρη	μάνι	τήρι
άδικη	ειου	ιάρηδες	μάρα	τηριού
άδικης	εια	ιάρηδων	μάρας	τήρια
άδικες	ειων	ιάρικο	μάρες	τηριών
άδικων	ειό	ιάρικου	μαρών	τήρια
άδικο	ειού	ιάρικα	μμα	τήριο
άδικου	ειά	ιάρικων	μματος	τηρίου
άδικα	ειών	ιάρισσα	μματα	τηρίων
άδικων	είο	ιάρισσας	μματων	τήριος
άδικος	είου	ιάρισσες	ξιμη	τήριου
άδικου	εία	ιαρισσών	ξιμης	τηρίου
άδικο	είων	ίας	ξιμες	τήριοι
άδικε	ειος	ία	ξιμων	τήριους
άδικοι	είου	ίες	ξιμο	τηρίους
άδικων	ειον	ιών	ξιμου	τική
άδικους	ειοι	ιάτικα	ξιμα	τικής
αδόρος	ειους	ιάτικη	ξιμος	τικές
αδόρου	έλι	ιάτικης	ξιμοι	τικών
αδόρε	ελιού	ιάτικες	ξιμους	τικιά
αδόροι	έλια	ιάτικων	όζικο	τικιάς
αδόρων	ελιών	ιάτικο	όζικου	τικιές
αδόρους	εμα	ιάτικου	όζικα	τικιών

αίικο	έματος	ιάτικα	όζικων	τικό
αίικου	έματα	ιάτικος	όζος	τικού
αίικα	εμάτων	ιάτικοι	όζου	τικά
αίικων	εμός	ιάτικους	όζοι	τικός
αινα	εμού	ιδα	όζους	τικοί
αινας	εμοί	ιδας	όλη	τικούς
αινες	εμών	ιδες	όλης	τορας
αινων	εμούς	ιδων	όλες	τορα
αίνω	ένια	ίδα	ολών	τορες
αίνεις	ένιας	ίδας	οπούλα	τορων
αίνει	ένιες	ίδες	οπούλας	τρα
αίνουμε	ενιών	ίδων	οπούλες	τρας
αίνετε	ένιο	ιδερή	οπουλων	τρες
αίνουν	ένιου	ιδερής	όπουλο	τρων
αινα	ένια	ιδερές	όπουλου	τρια
αινες	ένιος	ιδερών	όπουλα	τριας
αινε	ένιου	ιδερό	οστή	τρίας
αίναμε	ένιοι	ιδερού	οστής	τριες
αίνατε	ένιους	ιδερά	οστές	τριών
αιναν	ερή	ιδερός	οστών	τρο
άκης	ερής	ιδεροί	οστό	τρου
άκη	ερές	ιδερούς	οστού	τρα
άκηδες	ερών	ίδι	οστά	τσής
άκηδων	έρης	ιδιού	οστός	τσή
άκι	έρη	ιδίου	οστοί	τσήδες
άκια	έρηδες	ίδια	οστούς	τσήδων
άκιας	έρηδων	ιδίων	οσύνη	τσού
άκια	ερί	ιδιών	οσύνης	τσούς
άκλα	εριού	ίδικη	οσύνες	τσούδες
άκλας	εριά	ίδικης	ότατη	τσούδων
άκλες	εριών	ίδικες	οτάτη	ύδριο
άκλων	ερία	ίδικων	ότατης	υδρίου

ακλας	ερίας	ίδικο	οτάτης	ύδρια
ακλα	ερίες	ίδικου	ότατες	υδρίων
ακλαδες	εριών	ίδικα	ότατων	ύλλιο
ακλαδων	έρνω	ίδικος	οτάτων	υλλίου
άλα	έρνεις	ίδικοι	ότατο	ύλλια
άλας	έρνει	ίδικους	ότατου	υλλίων
άλες	έρνουμε	ιδίκων	οτάτου	ύλος
αλάκι	έρνετε	ιδίκους	ότατα	ύλου
αλάκια	έρνουν	ίδιο	ότατος	ύλοι
αλάς	ερνα	ιδίου	ότατοι	ύλων
αλά	ερνες	ίδια	ότατους	ύλους
αλάδες	ερνε	ιδίων	οτάτους	ύνομαι
αλάδων	έρναμε	ιέρα	ότερα	ύνεσαι
άλας	έρνατε	ιέρας	ότερη	ύνεται
άλα	ερναν	ιέρες	οτέρα	υνόμαστε
άλες	ερό	ιερών	ότερης	υνόμεθα
άλων	ερού	ιέρης	οτέρας	ύνεστε
αλέα	ερά	ιέρη	ότερες	ύνεσθε
αλέας	ερών	ιέρηδες	ότερων	ύνονται
αλέες	ερός	ιέρηδων	οτέρων	υνόμουν
αλέων	ερού	ίζω	ότερο	υνόσουν
αλέο	εροί	ίζεις	ότερου	υνόταν
αλέου	ερών	ίζει	οτέρου	υνόμασταν
αλέα	ερούς	ίζουμε	ότερα	υνόσασταν
αλέων	έσα	ίζετε	ότερος	ύνονταν
αλέος	έσας	ίζουν	ότεροι	ύνω
αλέου	έσες	ιζα	ότερους	ύνεις
αλέοι	εσών	ιζες	οτέρους	ύνει
αλέους	έστατα	ιζε	ότητα	ύνουμε
αλέων	έστατη	ίζαμε	ότητας	ύνετε
αλής	έστατης	ίζατε	ότητες	ύνουν
αλή	εστάτης	ιζαν	οτήτων	υνα

αλήδες	έστατες	ίσω	ούδα	υνες
αλήδων	έστατων	ίσεις	ούδας	υνε
αλού	εστάτων	ίσει	ούδες	ύναμε
αλούς	έστατο	ίσουμε	ούδων	ύνατε
αλούδες	έστατου	ίσετε	ουδάκι	υναν
αλούδων	εστάτου	ίσουν	ουδάκια	ύτατα
αμα	έστατα	ισα	ουδέλι	ύτατη
άματος	έστατος	ισες	ουδέλια	ύτατης
άματα	έστατον	ισε	ούδι	υτάτης
αμάτων	έστατοι	ίσαμε	ούδια	ύτατες
αμάρα	έστατους	ίσατε	ούδικο	ύτατων
αμάρας	εστάτους	ισαν	ούδικου	υτάτων
αμάρες	έστερα	ικη	ούδικα	ύτατο
αμός	έστερη	ικης	ούδικων	ύτατου
αμού	έστερης	ικες	ούκλα	υτάτου
αμοί	εστέρας	ικων	ούκλας	ύτατος
αμών	έστερες	ική	ούκλες	ύτατοι
αμούς	έστερων	ικής	ούκλας	ύτατους
άνα	εστέρων	ικές	ούλα	υτάτους
άνας	έστερο	ικών	ούλας	ύτερα
άνες	έστερου	ίκι	ούλες	ύτερη
ανών	εστέρου	ικιού	ουλών	υτέρα
ανή	έστερα	ικίου	ουλάκι	ύτερης
ανής	έστερος	ίκια	ουλάκια	υτέρας
ανές	έστερον	ικίων	ουλας	ύτερες
ανών	έστεροι	ικιά	ουλα	ύτερων
άνη	έστερους	ικιάς	ουλες	υτέρων
άνης	εστέρους	ικιές	ουλων	ύτερο
άνες	έτα	ικιών	ουλάς	ύτερου
ανών	έτας	ικο	ουλά	υτέρου
ανίδα	έτες	ικου	ουλάδες	ύτερος
ανίζω	ετών	ικα	ουλάδων	ύτεροι

ανίζεις	ετή	ικων	ουλή	ύτερους
ανίζει	ετής	ικό	ουλής	υτέρους
ανίζουμε	ετές	ικού	ουλές	ύτητα
ανίζετε	ετών	ικά	ουλών	ύτητας
ανίζουν	ετής	ικών	ούλης	ύτητες
άνιζα	ετούς	ικος	ούλη	υτήτων
άνιζες	ετείς	ικου	ούληδες	φτή
άνιζε	ετών	ικοι	ούληδων	φτής
ανίζαμε	ετό	ικων	ούλης	φτές
ανίζατε	ετού	ικους	ούλι	φτών
άνιζαν	ετά	ικός	ούλια	φτης
ανίσω	ετός	ικού	ούλι	φτη
ανίσεις	ετοί	ικοί	ούλιακας	φτες
ανίσει	ετούς	ικών	ούλιακα	φτων
ανίσουμε	εύομαι	ικούς	ούλιακες	φτό
ανίσετε	εύεσαι	ίλα	ουλίζω	φτού
ανίσουν	εύεται	ίλας	ουλίζεις	φτά
άνισα	ευόμαστε	ίλες	ουλίζει	φτών
άνισες	εύεστε	ιλίκι	ουλίζουμε	φτός
άνισε	εύονται	ιλίκικου	ουλίζετε	φτοί
ανίσαμε	ευόμουν	ιλικίου	ουλίζουν	φτούς
ανίσατε	ευόσουν	ιλίκια	ούλιζα	χτή
άνισαν	ευόταν	ιλικιών	ούλιζες	χτής
άντζα	ευόμασταν	ιλικίων	ούλιζε	χτές
άντζας	ευόσταστε	ιμη	ουλίζαμε	χτών
άντζες	ευόσασταν	ιμης	ουλίζατε	χτης
αντζών	ευόντουσαν	ιμες	ούλιζαν	χτη
άρα	εύω	ιμων	ουλίσω	χτες
άρας	εύεις	ιμο	ουλίσεις	χτής
άρες	εύει	ιμου	ουλίσει	χτή
αρών	εύουμε	ιμα	ουλίσουμε	χτές
αράδικο	εύετε	ιμων	ουλίσετε	χτό

αράδικου	εύουν	ιμος	ουλίσουν	χτού
αράδικα	ευα	ιμοι	ούλισα	χτά
αράδικων	ευες	ιμους	ούλισες	χτός
αράκι	ευε	ίνα	ούλισε	χτοί
αράκια	εύαμε	ίνας	ουλίσαμε	χτούς
αράκος	εύατε	ίνες	ουλίσατε	ψη
αράκου	ευαν	ινών	ούλισαν	ψης
αράκο	εύσω	ινη	ούλικα	ψεως
αράκοι	εύσεις	ινης	ούλικη	ψεις
αράκων	εύσει	ινες	ούλικης	ψεων
αράκους	εύσουμε	ινων	ούλικες	ψιά
αράς	εύσετε	ινή	ούλικων	ψιάς
αρά	εύσουν	ινής	ούλικια	ψιές
αράδες	ευσα	ινές	ούλικιας	ψιών
αράδων	ευσες	ινών	ούλικιες	ψία
άρας	ευσε	ίνη	ούλικιων	ψίας
άρα	εύσαμε	ίνης	ούλικο	ψίες
άρες	εύσατε	ίνες	ούλικου	ψιμη
αρέλι	ευσαν	ινών	ούλικα	ψιμης
αρέλια	έψω	ινιά	ούλικο	ψιμες
αρή	έψεις	ινιάς	ούλικος	ψιμων
άρης	έψει	ινιές	ούλικοι	ψιμο
άρη	έψουμε	ινιών	ούλικους	ψίματος
άρηδες	έψετε	ινο	ουλό	ψίματα
άρηδων	έψουν	ινου	ουλού	ψιμάτων
άρι	εψα	ινα	ουλά	ψιμος
αριού	εψες	ινων	ουλών	ψιμου
άρια	εψε	ινό	ουλός	ψιμοι
αριών	έψαμε	ινού	ουλοί	ψιμους
αριά	έψατε	ινά	ουλούς	ώα
αριάς	εψαν	ινών	ούρα	ώας
αριές	εώνας	ινος	ούρας	ώες

αριών	εώνα	ινοι	ούρες	ώων
αρία	εώνες	ινους	ουρών	ώδες
αρίας	εώνων	ινός	ουριά	ώδους
αρίες	ηδόν	ινοί	ουριάς	ώδη
αριών	ήθρα	ινούς	ουριές	ωδών
αρίζω	ημα	ιση	ουριών	ώδης
αρίζεις	ήματος	ισης	ουρίζω	ώδεις
αρίζει	ήματα	ισες	ουρίζεις	ώδικη
αρίζουμε	ημάτων	ισων	ουρίζει	ώδικης
αρίζετε	ημός	ίσια	ουρίζουμε	ώδικες
αρίζουν	ημού	ίσιας	ουρίζετε	ώδικων
άριζα	ημοί	ίσιες	ουρίζουν	ώδικο
άριζες	ημών	ισιων	ούριζα	ώδικου
άριζε	ημούς	ίσιο	ούριζες	ώδικα
αρίζαμε	ηρή	ίσιου	ούριζε	ώδικος
αρίζατε	ηρής	ίσια	ουρίζαμε	ώδικοι
άριζαν	ηρές	ίσιων	ουρίζατε	ώδικους
αρίσω	ηρών	ίσιος	ούριζαν	ωμα
αρίσεις	ηρό	ίσιοι	ουρίσω	ώματος
αρίσει	ηρού	ίσιους	ουρίσεις	ώματα
αρίσουμε	ηρά	ισίους	ουρίσει	ωμάτων
αρίσετε	ηρός	ίσκος	ουρίσουμε	ωμάρα
αρίσουν	ηροί	ίσκου	ουρίσετε	ωμάρας
άρισα	ηρούς	ίσκο	ουρίσουν	ωμάρες
άρισες	ηση	ίσκον	ούρισα	ωμένη
άρισε	ησης	ίσκοι	ούρισες	ωμένης
αρίσαμε	ήσεις	ίσκους	ούρισε	ωμένες
αρίσατε	ήσεων	ίσκων	ουρίσαμε	ωμένων
άρισαν	ήσια	ισμα	ουρίσατε	ωμένο
άρικο	ήσιας	ίσματος	ούρισαν	ωμένου
άρικου	ησίας	ίσματα	ούτσικα	ωμένα
άρικα	ήσιες	ισμάτων	ούτσικη	ωμένος

άρικων	ήσιων	ισμός	ούτσικης	ωμένοι
αριό	ησίων	ισμού	ούτσικες	ωμένους
αριού	ήσιος	ισμοί	ούτσικων	ωμός
αριά	ήσιου	ισμών	ούτσικια	ωμού
αριών	ησίου	ισμούς	ούτσικιας	ωμοί
άριο	ήσιοι	ιστη	ούτσικιες	ωμούς
αρίου	ήσιων	ιστης	ούτσικιων	ώνας
άρια	ησίων	ιστες	ούτσικο	ώνα
αρίων	ητη	ίστων	ούτσικος	ώνες
άρισσα	ητης	ιστή	ούτσικοι	ώνων
άρισσας	ητες	ιστής	ούτσικους	ώνομαι
άρισσες	ητων	ιστές	πλάσια	ώνεσαι
αρισσών	ητή	ιστών	πλάσιας	ώνεται
αρό	ητής	ιστής	πλάσιες	ωνόμαστε
αρού	ητές	ιστή	πλάσιων	ωνόμεθα
αρά	ητών	ιστού	πλασίων	ώνεστε
αρών	ητής	ιστή	πλάσιο	ώνεσθε
άρομαι	ητή	ιστές	πλάσιου	ώνονται
άρεσαι	ητές	ιστούς	πλασίου	ωνόμουν
άρεται	ητών	ιστική	πλάσιος	ωνόσουν
αρόμαστε	ήτης	ιστικής	πλάσιοι	ωνόταν
άρεστε	ήτη	ιστικές	πλάσιους	ωνόμασταν
άρονται	ήτες	ιστικών	πλασίους	ωνόσασταν
αρόμουν	ητών	ιστικιά	πλή	ώνονταν
αρόσουν	ητική	ιστικιάς	πλής	ώνω
αρόταν	ητικής	ιστικιές	πλές	ώνεις
αρόμασταν	ητικές	ιστικιών	πλών	ώνει
αρόσασταν	ητικών	ιστικό	πλό	ώνουμε
αρόντουσαν	ητικιά	ιστικού	πλού	ώνετε
αρός	ητικιάς	ιστικά	πλά	ώνουν
αρού	ητικιές	ιστικών	πλός	ωνα
αροί	ητικιών	ιστικός	πλοί	ωνες

αρών	ητικό	ιστικοί	πλούς	ωνε
αρού	ητικού	ιστικούς	πτης	ώναμε
αρούς	ητικά	ιστο	πτη	ώνατε
αρούδες	ητικών	ιστου	πτες	ωναν
αρούδι	ητικός	ιστα	πτων	ώσω
αρούδια	ητικοί	ιστων	ρίζω	ώσεις
αρούδικο	ητικούς	ιστό	ρίζω	ώσει
αρούδικου	ητο	ιστού	ρίζεις	ώσουμε
αρούδικα	ήτου	ιστά	ρίζει	ώσετε
αρούδικων	ητα	ιστών	ρίζουμε	ώσουν
άρω	ήτων	ιστος	ρίζετε	ωσα
άρεις	ητό	ιστοι	ρίζουν	ωσες
άρει	ητού	ιστους	ριζα	ωσε
άρουμε	ητά	ιστός	ριζες	ώσαμε
άρετε	ητών	ιστοί	ριζε	ώσατε
άρουν	ητος	ιστούς	ρίζαμε	ωσαν
αρα	ητου	ίστρα	ρίζατε	ώο
αρες	ήτου	ίστρας	ριζαν	ώου
αρε	ητοι	ίστρες	ρίσω	ώα
αραμε	ητων	ιστρών	ρίσεις	ώων
αρατε	ήτων	ιτζής	ρίσει	ώος
αραν	ητός	ιτζή	ρίσουμε	ώοι
ατζής	ητού	ιτζήδες	ρίσετε	ώους
ατζή	ητοί	ιτζήδων	ρίσουν	ωπή
ατζήδες	ητών	ιτζού	ρισα	ωπής
ατζήδων	ητούς	ιτζούς	ρισες	ωπές
ατζίδικη	ιάζομαι	ιτζούδες	ρισε	ωπών
ατζίδικης	ιάζεσαι	ιτζούδων	ρίσαμε	ωπό
ατζίδικες	ιάζεται	ίτιδα	ρίσατε	ωπού
ατζίδικων	ιαζόμαστε	ίτιδας	ρισαν	ωπά
ατζίδικο	ιάζεστε	ίτιδες	στη	ωπός
ατζίδικου	ιάζονται	ίτιδων	στης	ωποί

ατζίδικα	ιαζόμουν	ίτικη	στες	ωπούς
ατζίδικος	ιαζόσουν	ίτικης	στων	ως
ατζίδικοι	ιαζόταν	ίτικες	στης	ώς
ατζίδικους	ιαζόμασταν	ίτικων	στη	ωση
ατζού	ιαζόσασταν	ίτικο	στής	ωσις
ατζούς	ιάζονταν	ίτικου	στή	ωσης
ατζούδες	ιάζω	ίτικα	στές	ώσεως
ατζούδων	ιάζεις	ίτικος	στών	ώσεις
άτικη	ιάζει	ίτικοι	στική	ώσεων
άτικης	ιάζουμε	ίτικους	στικής	ωση
άτικες	ιάζετε	ίτσα	στικές	ωσιά
άτικων	ιάζουν	ίτσας	στικών	ωσιάς
άτικο	ίαζα	ίτσες	στικιά	ωσιές
άτικου	ίαζες	ιτσών	στικιάς	ωσιών
άτικα	ίαζε	ιώνας	στικιές	ωτη
άτικος	ιάζαμε	ιώνα	στικιών	ωτης
άτικοι	ιάζατε	ιώνες	στικό	ωτες
άτικους	ίαζαν	ιώνων	στικού	ωτων
άτισσα	ιάσω	ιώτης	στικά	ωτή
άτισσας	ιάσεις	ιώτη	στικός	ωτής
άτισσες	ιάσει	ιώτες	στικοί	ωτές
ατισσών	ιάσουμε	ιωτών	στικούς	ωτών
άτο	ιάσετε	ιώτικο	στό	ωτής
άτου	ιάσουν	ιώτικου	στός	ωτή
άτα	ίασα	ιώτικα	στού	ώτης
άτων	ίασες	ιώτικων	στοί	ώτη
άτορας	ίασε	ιώτισσα	στούς	ώτες
άτορα	ιάσαμε	ιώτισσας	στών	ωτών
άτορες	ιάσατε	ιώτισσες	τέα	ωτική
ατόρων	ίασαν	ιώτισσων	τέας	ωτικής
άτος	ιαία	καιρη	τέες	ωτικές
άτου	ιαίας	καιρης	τέων	ωτικών

άτοι	ιαίες	καιρες	τέο	ωτικιά
άτων	ιαίων	καιρων	τέου	ωτικιάς
άτους	ιαίο	καιρο	τέος	ωτικιές
γμα	ιαίου	καιρου	τέοι	ωτικιών
γματος	ιαία	καιρα	τέους	ωτικό
γματα	ιαίος	καιρος	τερή	ωτικού
γματων	ιαίοι	καιροι	τερής	ωτικά
γμός	ιαίους	καιρους	τερές	ώτικο
γμού	ιακή	κοπη	τερών	ώτικου
γμοί	ιακής	κοπης	τερό	ώτικα
γμών	ιακές	κοπες	τερού	ώτικων
γμούς	ιακών	κοπων	τερά	ωτικός
γονος	ιακό	κοπο	τερός	ωτικοί
γονου	ιακού	κοπου	τεροί	ωτικούς
γονοι	ιακά	κοπα	τερούς	ώτισσα
γονων	ιακός	κοπος	τζής	ώτισσας
γονους	ιακοί	κοποι	τζή	ώτισσες
δήποτε	ιακούς	κοπους	τζήδες	ωτισσών
έας	ιάνα	κτης	τζήδων	ωτο
έα	ιάνας	κτη	τζής	ωτου
είς	ιάνες	κτες	τζίδικη	ωτα
έων	ιανών	κτων	τζίδικης	ωτων
εια	ιανή	λής	τζίδικες	ωτό
ειας	ιανής	λή	τζίδικων	ωτού
ειες	ιανές	λήδες	τζίδικο	ωτά
ειών	ιανών	λήδων	τζίδικου	ωτών
ειά	ιανό	λίκι	τζίδικα	ωτος
ειάς	ιανού	λικιού	τζίδικος	ωτοι
ειές	ιανά	λίκια	τζίδικοι	ωτους
				ωτός
				ωτοί
				ωτούς

Σύνδεσμοι

με τον τρόπο που	και	όποιες	όπου
ακόμη κι αν	καθώς	οποιεσδήποτε	οπουδήποτε
κι ας μην	εφόσον	οποίο	όπως
έτσι και	επειδή	όποιο	όσα
έστω κι αν	ενώ	οποιοδήποτε	όσες
κι αν	εάν	οποίοι	όση
για να	διότι	όποιοι	όσης
αν και	γιατί	οποιοιδήποτε	όσο
μια και	αφού	οποίον	όσος
πάνω που	αν	όποιον	όσου
παρ' όλο	άμα	οποιονδήποτε	όσους
παρ' όλο που	αλλά	οποίος	όσων
παρόλο που	μήπως	όποιος	όταν
που να	μόλις	οποιοσδήποτε	ότι
πριν να	μολονότι	οποίου	ό,τι
προτού να	να	όποιου	οτιδήποτε
σαν να	όμως	οποιουδήποτε	ο,τιδήποτε
σε περίπτωση που	οποία	οποίους	παρόλο
τη στιγμή που	όποια	όποιους	που
ώστε να	οποιαδήποτε	οποιουσδήποτε	πριν
μα	οποίας	οποίων	προτού
λοιπόν	όποιας	όποιων	πως
κι	οποιασδήποτε	οποιωνδήποτε	ώστε
	οποίες	όποτε	ωστόσο

Επιθήματα Μετοχών

έντος	οντα	αντα	μένο	μενου
έντα	οντες	αντες	μένοι	μενο
έντες	όντων	άντων	μένων	μενοι
έντων	ουσα	άσης	μένους	μένων
είσα	ούσης	άσας	μένη	μενους
είσης	ούσας	ασαι	μένης	μενη
είσας	ουσαι	ασες	μένες	μενης
είσες	ούσες	ασών	μένων	μενες
εισών	ουσών	αντος	μένο	μενων
έντος	οντος	αντα	μένου	μενο
έντα	οντα	άντων	μένα	μενου
έντων	όντων	μένος	μένων	μενα
οντος	αντος	μένου	μενος	μενων