



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
Τμήμα Πληροφορικής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Αναγνώριση προσβλητικού λόγου σε Ελληνικά
tweets, με χρήση τεχνικών Βαθιάς Μάθησης**

**Offensive speech detection in Greek tweets, using
Deep Learning techniques**

Χρήστος Ζηγγόπης

A.M. Π2019027

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ

Δρ. Κάτια Λήδα Κερμανίδου

ΕΠΙΤΡΟΠΗ

Δρ. Αριστείδης Βραχάτης

Δρ. Ανδρέας Καναβός

ΚΕΡΚΥΡΑ, 2024

Περίληψη

Η συχνή εμφάνιση εχθρικού περιεχομένου στα μέσα κοινωνικής δικτύωσης, έχει δημιουργήσει την ανάγκη για αυτόματη αναγνώριση αυτών των αναρτήσεων, καθώς οι διαχειριστές δεν έχουν το χρόνο να επεξεργαστούν και να τις απορρίψουν. Ο κλάδος της πληροφορικής που ασχολείται με την αναγνώριση του νοήματος σε σώματα κειμένων είναι η τεχνητή νοημοσύνη και πιο συγκεκριμένα ο τομέας Επεξεργασίας Φυσικής Γλώσσας. Οι πρόσφατες εξελίξεις, που βασίζονται στις αρχιτεκτονικές βαθιάς μάθησης όπως οι μετασχηματιστές - transformers, έχουν δώσει τεράστια ώθηση στην κατανόηση της φυσικής γλώσσας. Στην εργασία μου αυτή, χρησιμοποιώ και προσαρμόζω ένα προεκπαιδευμένο μοντέλο μετασχηματιστή, το GreekBERT που με τη σειρά του είναι προσαρμογή του αρχικού BERT σε Ελληνικά κείμενα. Για την εκπαίδευση / προσαρμογή του προεκπαιδευμένου μοντέλου, χρησιμοποιώ ένα επισημασμένο dataset που αποτελείται από αναρτήσεις - tweets στην Ελληνική γλώσσα. Μετά την προεπεξεργασία του dataset, το μοντέλο εκπαιδεύεται με διάφορους συνδυασμούς υπερ παραμέτρων, ώστε να επιτευχθεί το καλύτερο αποτέλεσμα. Το τελικό μοντέλο, το οποίο είναι ελεύθερο προς χρήση για όλους, έχει δημοσιευθεί στην πλατφόρμα Hugging Face.

Λέξεις Κλειδιά

Ταξινόμηση Κειμένου, Αναγνώριση Εχθρικού Λόγου, Βαθιά Μάθηση, Τεχνητή Νοημοσύνη, Επεξεργασία Φυσικής Γλώσσας, BERT, Νευρωνικά Δίκτυα, Μετασχηματιστές

Abstract

The frequent appearance of hateful content on social media has created the need for automatic identification of these posts, as moderators do not have the time to process and reject them. The field of computer science that deals with the recognition of meaning in text corpora is artificial intelligence and more specifically the subfield of Natural Language Processing (NLP). Recent developments, based on deep learning architectures such as transformers, have given a huge boost to the understanding of natural language. In this work, I use and adapt a pre-trained transformer model, GreekBERT, which is in turn an adaptation of the original BERT to Greek texts. For the training/adaptation of the pre-trained model, I use a labeled dataset consisting of tweets written in Greek. After preprocessing the dataset, the model is trained with different combinations of hyperparameters in order to achieve the best results. The final model, which is free for everyone to use, has been published on the Hugging Face platform.

Keywords

Text Classification, Offensive Speech Detection, Deep Learning, Artificial Intelligence, Natural Language Processing, BERT, Neural Networks, Transformers

Περιεχόμενα

Περίληψη.....	2
Λέξεις Κλειδιά.....	2
Abstract.....	2
Keywords.....	3
Λίστα εικόνων.....	6
1. Εισαγωγή.....	10
1.1 Αναγνώριση Εχθρικού Λόγου.....	10
1.2 Εφαρμογές Αναγνώρισης Εχθρικού Λόγου.....	11
2. Θεωρητικό Υπόβαθρο.....	12
2.1 Μηχανική Μάθηση.....	12
2.2 Επεξεργασία Φυσικής Γλώσσας.....	13
2.3 Νευρωνικά δίκτυα και Βαθιά μάθηση - Deep learning.....	15
2.3.1 Τεχνητά νευρωνικά δίκτυα.....	15
Ο Νευρώνας Perceptron.....	15
Perceptron ενός επιπέδου και πολλών επιπέδων.....	18
2.3.2 Βαθιά μάθηση.....	19
Convolutional Neural Networks.....	20
Αρχιτεκτονική ενός CNN.....	20
Convolution Layer.....	20
Pooling Layer.....	21
Fully connected layers.....	22
Recurrent Neural Network (RNN).....	22
Long short-term memory.....	23
Μετασχηματιστές (Transformers).....	24
Ο κωδικοποιητής (encoder).....	25
Ο αποκωδικοποιητής (decoder).....	25
Λειτουργία των μετασχηματιστών (transformers).....	26
2.4 Προεπεξεργασία σώματος κειμένων.....	27
2.5 Εξαγωγή Χαρακτηριστικών.....	28
2.5.1 Bag of Words (Μοντέλο σακούλας λέξεων).....	30
2.5.2 N-grams (Ακολουθίες n λέξεων).....	30
2.5.3 TF-IDF.....	31
2.5.4 Vector space models (μοντέλα διανυσματικού χώρου).....	31
Word2Vec.....	32
CBOW.....	33
Skip Gram.....	34
2.6 Μοντέλα ΕΦΓ που βασίζονται στους μετασχηματιστές.....	34
2.6.1 BERT.....	34
Λειτουργία του BERT.....	35
Μηχανισμός αυτο-προσοχής (self-attention).....	36
2.6.2 Άλλα μοντέλα που βασίζονται σε Transformer encoders.....	37
RoBERTa.....	37
ALBERT.....	37

2.6.3 Μοντέλα που βασίζονται σε Transformer decoders.....	38
3. Μεθοδολογία υλοποίησης.....	40
3.1 Περιγραφή Συνόλου Δεδομένων (Dataset).....	40
3.2 Περιγραφή προεπεξεργασίας δεδομένων, στο dataset που χρησιμοποιήθηκε.....	41
3.2.1 Μετατροπή σε πεζά.....	42
3.2.2 Αφαίρεση τόνων και διακριτικών.....	42
3.2.3 Άλλες αλλαγές και αποφάσεις.....	42
3.3 Περιγραφή Εργαλείων και βιβλιοθηκών.....	43
3.3.1 Google Colaboratory (Colab).....	43
3.3.2 Hugging Face.....	44
3.3.3 Hugging Face Transformers.....	44
3.3.4 Άλλα εργαλεία.....	44
3.3.5 Συνδυασμένη χρήση εργαλείων.....	44
3.4 Αναλυτική περιγραφή αρχιτεκτονικής βαθιάς μάθησης.....	45
3.4.1 Ανάλυση σε σύμβολα.....	47
3.4.2 Εξαγωγή χαρακτηριστικών.....	49
3.4.3 Προσαρμογή (fine tuning) του μοντέλου.....	52
3.5 Παρουσίαση αποτελεσμάτων πειραμάτων για διάφορους συνδυασμούς υπερ παραμέτρων.....	56
3.5.1 Μη ισορροπημένο dataset, με σταθερό learning rate.....	56
3.5.2 Ισορροπημένο dataset, με σταθερό learning rate.....	57
3.5.3 Μη ισορροπημένο dataset, με σταθερό μεγάλο learning rate.....	59
3.5.4 Ισορροπημένο dataset, με σταθερό μεγάλο learning rate.....	60
3.5.5 Μη ισορροπημένο dataset, με learning rate που μειώνεται γραμμικά.....	61
3.5.6 Ισορροπημένο dataset, με learning rate που μειώνεται γραμμικά.....	62
3.5.7 Ισορροπημένο dataset, χωρίς τα hashtags, με learning rate που μειώνεται γραμμικά.....	63
3.6 Σύγκριση αποτελεσμάτων.....	65
3.6.1 Διερεύνηση διαφορών ταξινόμησης, στο test dataset.....	66
3.6.2 Σύγκριση με αντίστοιχες εργασίες.....	69
4. Σύνοψη.....	70
4.1 Συμπεράσματα.....	70
4.2 Βελτιώσεις - μελλοντικές επεκτάσεις.....	71
5. Κώδικας υλοποίησης και προσαρμοσμένο μοντέλο.....	72
6. Βιβλιογραφία.....	72

Λίστα εικόνων

Εικόνα 2.1 - Νευρώνας Perceptron

Πηγή: <https://datascientest.com/en/perceptron-definition-and-use-cases>

Εικόνα 2.2 - Sigmoid function

Πηγή: https://en.wikipedia.org/wiki/Activation_function

Εικόνα 2.3 - ReLU

Πηγή: https://en.wikipedia.org/wiki/Activation_function

Εικόνα 2.4 - Softplus

Πηγή: https://en.wikipedia.org/wiki/Activation_function

Εικόνα 2.5 - Tanh

Πηγή: https://en.wikipedia.org/wiki/Activation_function

Εικόνα 2.6 - Feed forward neural network

Πηγή: Βιβλίο “Τεχνητή νοημοσύνη: Μια σύγχρονη προσέγγιση”

Εικόνα 2.7 - Feed forward neural network

Πηγή: <http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks>

Εικόνα 2.8 - Convolutional neural network

Πηγή: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn>

Εικόνα 2.9 - Pooling layer

Πηγή: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn>

Εικόνα 2.10 - Recurrent Neural Network

Πηγή: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types>

Εικόνα 2.11 - Long-short term memory

Πηγή: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types>

Εικόνα 2.12 - Transformer model

Πηγή: <https://machinelearningmastery.com/the-transformer-model>

Εικόνα 2.13 - Bag of Words

Πηγή: <https://www.kdnuggets.com/2019/11/text-encoding-review.html>

Εικόνα 2.14 - N-grams

Πηγή: <https://www.kdnuggets.com/2022/06/ngram-language-modeling-natural-language-processing.html>

Εικόνα 2.15 - TF-IDF

Πηγή: <https://web.stanford.edu/~jurafsky/slp3/14.pdf>

Εικόνα 2.16 - Contextual word meaning

Πηγή: <https://towardsdatascience.com/vector-space-models-48b42a15d86d>

Εικόνα 2.17 - Cosine similarity

Πηγή: <https://www.exxactcorp.com/blog/Deep-Learning/text-preprocessing-methods-for-deep-learning>

Εικόνα 2.18 - CBOW vs Skip-gram

Πηγή: <https://www.researchgate.net/profile/Daniel-Braun-6/publication/326588219/figure/fig1/AS:652185784295425@1532504616288/Continuous-Bag-of-words-CBOW-CB-and-Skip-gram-SG-training-model-illustrations.png>

Εικόνα 2.19 - BERT self-attention

Πηγή: <https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model>

Εικόνα 2.20 - Decoder-only models

Πηγή: <https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-qi6vc/>

Εικόνα 2.21 - Decoder-only architecture

Πηγή: <https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-qi6vc/>

Εικόνα 3.1 - Απόσπασμα από τα tweets Dataset (πριν την επεξεργασία)

Εικόνα 3.2 - Αρχιτεκτονική υλοποίησης

Εικόνα 3.3 - Απόσπασμα Ελληνικών tweets που φορτώθηκε στο Colab

Εικόνα 3.4 - Το dataset μετά την προεπεξεργασία

Εικόνα 3.5 - Το dataset μετά την αφαίρεση των περιττών στηλών

Εικόνα 3.6 - Παράδειγμα tokenization Ελληνικής πρότασης

Εικόνα 3.7 - Παράδειγμα tokenization Ελληνικής πρότασης που περιέχει hashtags

Εικόνα 3.8 - Παράδειγμα μετατροπής πρότασης σε λίστα από token ids

Εικόνα 3.9 - Δημιουργία token ids, token type ids και attention mask, μιας πρότασης

Εικόνα 3.10 - Tokenization, padding, truncation, λίστας προτάσεων

Εικόνα 3.11 - Embeddings που παράγει το BERT για τη λίστα προτάσεων

Εικόνα 3.12 - Αρχικοποίηση του μοντέλου με βάση το GreekBERT

Εικόνα 3.13 - Αρχιτεκτονική του μοντέλου για προσαρμογή (fine tuning)

Εικόνα 3.14 - Προτροπή του HuggingFace για προσαρμογή του μοντέλου

Εικόνα 3.15 - Χαρακτηριστικά του classifier επιπέδου

Εικόνα 3.16 - Χαρακτηριστικά του dropout επιπέδου

Εικόνα 3.17 - Χαρακτηριστικά του BERT επιπέδου

*Εικόνα 3.18 - Προσαρμογή - **fit** (fine tuning) του μοντέλου με τα Ελληνικά tweets*

Εικόνα 3.19 - Γράφημα των metrics για μη ισορροπημένο dataset, με σταθερό learning rate

Εικόνα 3.20 - Γράφημα των metrics για ισορροπημένο dataset, με σταθερό learning rate

Εικόνα 3.21 - Γράφημα των metrics για μη ισορροπημένο dataset, με σταθερό (μεγάλο) learning rate

Εικόνα 3.22 - Γράφημα των metrics για ισορροπημένο dataset, με σταθερό (μεγάλο) learning rate

Εικόνα 3.23 - Γράφημα των metrics για μη ισορροπημένο dataset, με γραμμικά μειούμενο learning rate

Εικόνα 3.24 - Γράφημα των metrics για ισορροπημένο dataset, με γραμμικά μειούμενο learning rate

*Εικόνα 3.25 - Γράφημα των metrics για ισορροπημένο dataset, **χωρίς hashtags**, με γραμμικά μειούμενο learning rate*

Εικόνα 3.26 - Σύγκριση των metrics για τις διάφορες κατηγορίες πειραμάτων

Εικόνα 3.27 - Tweets που ταξινομήθηκαν διαφορετικά από το μοντέλο σε σχέση με την επισημάνση

Εικόνα 3.28 - Αποτελέσματα ταξινόμησης Tweets, για πέντε γλώσσες

1. Εισαγωγή

1.1 Αναγνώριση Εχθρικού Λόγου

Το διαδίκτυο έχει αλλάξει ριζικά τον τρόπο επικοινωνίας των ανθρώπων, αλλά ταυτόχρονα έχει δώσει τη δυνατότητα για την ταχεία εξάπλωση του προσβλητικού λόγου. Από τα απαξιωτικά σχόλια μέχρι τον διαδικτυακό εκφοβισμό, αυτός ο λόγος μπορεί να έχει πραγματικό αντίκτυπο, δημιουργώντας εχθρικά περιβάλλοντα και προωθώντας το μίσος εναντίων συγκεκριμένων ανθρώπων ή ομάδων. Για το λόγο αυτό, γίνονται προσπάθειες για τον εντοπισμό προσβλητικού λόγου με αυτοματοποιημένο τρόπο. Πρόκειται για μια τεχνολογική προσέγγιση η οποία χρησιμοποιώντας σύνθετους αλγορίθμους και μηχανική μάθηση, αναλύει κείμενο, ήχο και ακόμη και οπτικό υλικό, επισημαίνοντας πιθανές εκφάνσεις εχθρικού περιεχομένου. Στόχος είναι η δημιουργία φίλτρων που θα μπορούν να εντοπίσουν και να περιορίσουν την εξάπλωση τέτοιου περιεχομένου.

Υπάρχουν διάφορες μέθοδοι εντοπισμού εχθρικού λόγου όπως:

- Αναγνώριση προσβλητικών λέξεων: Τα συστήματα εκπαιδεύονται σε εκτεταμένα σύνολα δεδομένων που περιέχουν εχθρική γλώσσα, μαθαίνοντας να αναγνωρίζουν ύβρεις, προσβολές και όρους μίσους.
- Κατανόηση του περιεχομένου: Πολλές φορές δεν αρκούν μόνο οι ίδιες οι λέξεις. Τα συστήματα ανίχνευσης αναλύουν το πλαίσιο στο οποίο χρησιμοποιούνται. Η ειρωνεία, ο σαρκασμός και οι πολιτιστικές αναφορές μπορούν να επηρεάσουν το νόημα.
- Εντοπισμός συνηθισμένων ομάδων/στόχων: Ο εχθρικός λόγος συχνά στοχεύει σε συγκεκριμένες ομάδες με βάση τη φυλή, τη θρησκεία, το φύλο ή τον σεξουαλικό προσανατολισμό. Τα συστήματα ανίχνευσης μπορούν να εντοπίσουν γλώσσα που επιτίθεται σε αυτά τα προστατευόμενα χαρακτηριστικά (protected characteristics).

Ωστόσο, παρά την μεγάλη εξέλιξη στον τομέα της μηχανικής μάθησης και της σημαντικής αύξησης δυνατοτήτων των σύγχρονων υπολογιστικών συστημάτων, εξακολουθούν να υπάρχουν προκλήσεις, όπως:

- Η γλώσσα μπορεί να είναι περίπλοκη και με υπονοούμενα. Ο σαρκασμός και οι πολιτιστικές αναφορές θέτουν προκλήσεις στα συστήματα ανίχνευσης.

- Εξέλιξη της γλώσσας. Νέοι και απαξιωτικοί όροι εμφανίζονται συνέχεια, απαιτώντας συνεχείς ενημερώσεις στα μοντέλα αντίχνευσης
- Κανένα σύστημα δεν είναι τέλειο. Η αντίχνευση μπορεί να επισημάνει εσφαλμένα ακίνδυνο περιεχόμενο (ψευδώς θετικά) ή να μην αντιληφθεί την πραγματικά προσβλητική γλώσσα (ψευδώς αρνητικά)

Για τους παραπάνω λόγους, οι ερευνητές εργάζονται συνεχώς για τη βελτίωση αυτών των συστημάτων, εστιάζοντας σε βελτιώσεις όπως η κατανόηση της πρόθεσης δηλαδή από τις ίδιες τις λέξεις για να αναλύσουμε την πρόθεση πίσω από τη γλώσσα, η πολυγλωσσική αντίχνευση και η ανάπτυξη προσαρμοστικών μοντέλων που μπορούν να μάθουν νέες μορφές προσβλητικού λόγου.

1.2 Εφαρμογές Αναγνώρισης Εχθρικού Λόγου

Ο εντοπισμός εχθρικού λόγου είναι ένα ισχυρό εργαλείο για την προώθηση ασφαλέστερων διαδικτυακών χώρων. Μπορεί να προσφέρει:

- Βελτιωμένη εμπειρία χρήστη. Ο προσβλητικός λόγος δημιουργεί ένα εχθρικό περιβάλλον, απομακρύνοντας τους χρήστες από τις διαδικτυακές πλατφόρμες. Η αντίχνευση μπορεί να βοηθήσει στην απομάκρυνση του αρνητικού περιεχομένου, προωθώντας μια πιο θετική εμπειρία στους χρήστες.
- Ενίσχυση της φήμης μιας επωνυμίας (brand name). Οι πλατφόρμες που καταπολεμούν ενεργά το προσβλητικό περιεχόμενο αποδεικνύουν τη δέσμευσή τους στην υπεύθυνη επικοινωνία. Αυτό μπορεί να ενισχύσει τη φήμη της επωνυμίας και να προσελκύσει χρήστες που εκτιμούν τις πλατφόρμες οι οποίες προωθούν το σεβασμό.
- Παρακίνηση των χρηστών για αναφορά αρνητικού περιεχομένου. Ο εντοπισμός προσβλητικού λόγου δίνει κίνητρο στους επισκέπτες του ιστότοπου να αναφέρουν ευκολότερα το επιβλαβές περιεχόμενο. Η γνώση ότι το σύστημα μπορεί να επισημάνει πιθανές προσβολές μπορεί να ενθαρρύνει τους χρήστες να μιλήσουν ανοιχτά κατά του αρνητικού περιεχομένου.
- Τα συστήματα αντίχνευσης μπορούν να αναλύσουν πρότυπα στον προσβλητικό λόγο, παρέχοντας πολύτιμες πληροφορίες, οι οποίες μπορούν να χρησιμοποιηθούν για τον εντοπισμό αναδυόμενων τάσεων στον εχθρικό λόγο ή στις τακτικές εκφοβισμού, επιτρέποντας στοχευμένες παρεμβάσεις.

- Αποτελεσματικότερη διαχείριση περιεχομένου. Αυτά τα συστήματα μπορούν να μειώσουν σημαντικά τον φόρτο εργασίας των διαχειριστών περιεχομένου. Με τον αυτόματο εντοπισμό πιθανών προσβολών, οι διαχειριστές μπορούν να επικεντρωθούν στην αντιμετώπιση πιο σοβαρών περιπτώσεων και στην ανάπτυξη αποτελεσματικών στρατηγικών διαχείρισης περιεχομένου.
- Τα προσβλητικά σχόλια σε έναν ιστότοπο πωλήσεων, ίσως να αποτελούν ένδειξη ότι υπάρχει σοβαρή ενόχληση μεταξύ κάποιων πελατών και να κινητοποιήσουν το τμήμα εξυπηρέτησης πελατών για να αποκλιμακώσουν την κατάσταση πριν ξεφύγει από τον έλεγχο.
- Προώθηση του κοινωνικού καλού. Ο προσβλητικός λόγος μπορεί να υποκινήσει διακρίσεις και βία. Περιορίζοντας την εξάπλωσή του, αυτά τα συστήματα μπορούν να συμβάλλουν στην προώθηση του κοινού καλού, ενθαρρύνοντας διαδικτυακές κοινότητες χτισμένες πάνω στον σεβασμό και την κατανόηση.

2. Θεωρητικό Υπόβαθρο

2.1 Μηχανική Μάθηση

Η μηχανική μάθηση είναι ένας κλάδος της επιστήμης των υπολογιστών που ασχολείται με την ανάπτυξη αλγορίθμων που μπορούν να μάθουν από δεδομένα χωρίς να έχουν προγραμματιστεί ρητά. Είναι στενά συνδεδεμένη με την υπολογιστική στατιστική, η οποία ασχολείται με την ανάλυση δεδομένων και την εξαγωγή συμπερασμάτων.

Υπάρχουν τρεις βασικές κατηγορίες μηχανικής μάθησης:

- Επιβλεπόμενη μάθηση: Στην επιβλεπόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται σε ένα σύνολο δεδομένων που περιλαμβάνει παραδείγματα του προβλήματος που πρέπει να λυθεί. Για παράδειγμα, ένας αλγόριθμος επιβλεπόμενης μάθησης μπορεί να εκπαιδευτεί σε ένα σύνολο δεδομένων φωτογραφιών προσώπων με ετικέτες, ώστε να μάθει να αναγνωρίζει πρόσωπα.
- Μη επιβλεπόμενη μάθηση: Στη μη επιβλεπόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται σε ένα σύνολο δεδομένων που δεν περιλαμβάνει παραδείγματα του προβλήματος που πρέπει να λυθεί. Για παράδειγμα, ένας αλγόριθμος μη επιβλεπόμενης μάθησης μπορεί να εκπαιδευτεί σε ένα σύνολο δεδομένων αριθμών, ώστε να μάθει να ομαδοποιεί τους αριθμούς σε ομάδες με βάση τα κοινά τους χαρακτηριστικά.

- **Ημι-επιβλεπόμενη μάθηση:** Στην ημι-επιβλεπόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται σε ένα σύνολο δεδομένων που περιλαμβάνει ένα μικρό ποσοστό δεδομένων με ετικέτες και ένα μεγάλο ποσοστό δεδομένων χωρίς ετικέτες. Για παράδειγμα, ένας αλγόριθμος ημι-επιβλεπόμενης μάθησης μπορεί να εκπαιδευτεί σε ένα σύνολο δεδομένων φωτογραφιών προσώπων με λίγες ετικέτες, ώστε να μάθει να αναγνωρίζει πρόσωπα με μεγαλύτερη ακρίβεια από έναν αλγόριθμο μη επιβλεπόμενης μάθησης.

Η μηχανική μάθηση έχει ευρεία εφαρμογή σε πολλούς τομείς, όπως:

- Αναγνώριση εικόνων και βίντεο
- Πρόγνωση μελλοντικών καταστάσεων με βάση τα ιστορικά δεδομένα
- Ρομποτική
- Κατανόηση και ταξινόμηση κειμένου

Όπως γίνεται αντιληπτό, η μηχανική μάθηση είναι ένα δυναμικά αναπτυσσόμενο πεδίο με πολλές δυνατότητες, και πολλά υποσχόμενο μέλλον.

2.2 Επεξεργασία Φυσικής Γλώσσας

Η επεξεργασία φυσικής γλώσσας (ΕΦΓ) είναι ο κλάδος της επιστήμης των υπολογιστών που ασχολείται με την κατανόηση και επεξεργασία της ανθρώπινης γλώσσας από υπολογιστικές μηχανές.

Με την εφαρμογή τεχνικών ΕΦΓ, οι υπολογιστές μπορούν να αναγνωρίσουν τη γραμματική, το συντακτικό και το νόημα του κειμένου.

Οι εφαρμογές της είναι ευρέως διαδεδομένες και περιλαμβάνουν:

- **Κατηγοριοποίηση ολόκληρων προτάσεων:** Υπάρχουν διάφορες κατηγορίες στις οποίες μπορεί να ταξινομηθεί μια πρόταση όπως, προσδιορισμός του συναισθήματος (sentiment) μιας κριτικής, ανίχνευση εάν ένα email είναι spam, προσδιορισμός εάν μια πρόταση είναι γραμματικά σωστή ή εάν δύο προτάσεις συνδέονται λογικά μεταξύ τους

- Απάντηση ερωτήσεων: Η παροχή απαντήσεων σε ερωτήσεις διατυπωμένες σε φυσική γλώσσα.
- Αναγνώριση επωνύμων οντοτήτων (named entity recognition): Η αναγνώριση και η κατηγοριοποίηση ονομάτων προσώπων, τοποθεσιών, οργανισμών κ.λπ.
- Μηχανική μετάφραση: Η μετάφραση κειμένου από μία γλώσσα σε άλλη.
- Εξαγωγή πληροφοριών: Η ανάκτηση πληροφοριών από κείμενα.

Η βασική μεθοδολογία για την υλοποίηση εφαρμογών ΕΦΓ περιλαμβάνει τα εξής βήματα:

1. Συλλογή δεδομένων: Η συλλογή ενός σώματος κειμένου που αντιπροσωπεύει την κατάλληλη πληροφορία για τον εκάστοτε σκοπό.
2. Επισήμανση κειμένου: Η επισήμανση κειμένου είναι η διαδικασία προσθήκης πληροφοριών σε ένα κείμενο για να διευκολύνεται η ανάλυση και η κατανόησή του από μηχανές. Αυτή η διαδικασία μπορεί να περιλαμβάνει την προσθήκη τεκμηρίων, σημειώσεων, επισημάνσεων ή άλλων σχετικών πληροφοριών.
3. Προεπεξεργασία δεδομένων: Είναι η διαδικασία καθαρισμού, μορφοποίησης και μετατροπής ακατέργαστου κειμένου δεδομένων σε μορφή κατάλληλη για εργασίες ΕΦΓ. Αυτό είναι απαραίτητο επειδή το ακατέργαστο κείμενο δεδομένων μπορεί να είναι άμορφο και μη δομημένο, περιέχοντας θόρυβο, λάθη και ασυνέπειες. Η προεπεξεργασία κειμένου βοηθά στη βελτίωση της ποιότητας και της ακρίβειας των μοντέλων ΕΦΓ, αφαιρώντας τις άσχετες και περιττές πληροφορίες και προετοιμάζοντας το κείμενο για περαιτέρω ανάλυση.
4. Αριθμητική αναπαράσταση δεδομένων: Η μετατροπή του σώματος κειμένου σε αριθμητική μορφή για την ανάλυση από υπολογιστικές μηχανές.
5. Εκπαίδευση γλωσσικού μοντέλου: Η εκπαίδευση ενός γλωσσικού μοντέλου που μπορεί να κατανοήσει τη φυσική γλώσσα.
6. Αξιολόγηση γλωσσικού μοντέλου: Η αξιολόγηση της απόδοσης του γλωσσικού μοντέλου.

Η ΕΦΓ είναι ένας δυναμικά αναπτυσσόμενος τομέας της επιστήμης των υπολογιστών με ευρεία εφαρμογή σε διάφορους τομείς και έχει τη δυνατότητα να βελτιώσει σημαντικά τον τρόπο με τον οποίο αλληλεπιδρούμε με τους υπολογιστές, καθιστώντας την επικοινωνία μας με αυτούς πιο φυσική και αποτελεσματική.

2.3 Νευρωνικά δίκτυα και Βαθιά μάθηση - Deep learning

2.3.1 Τεχνητά νευρωνικά δίκτυα

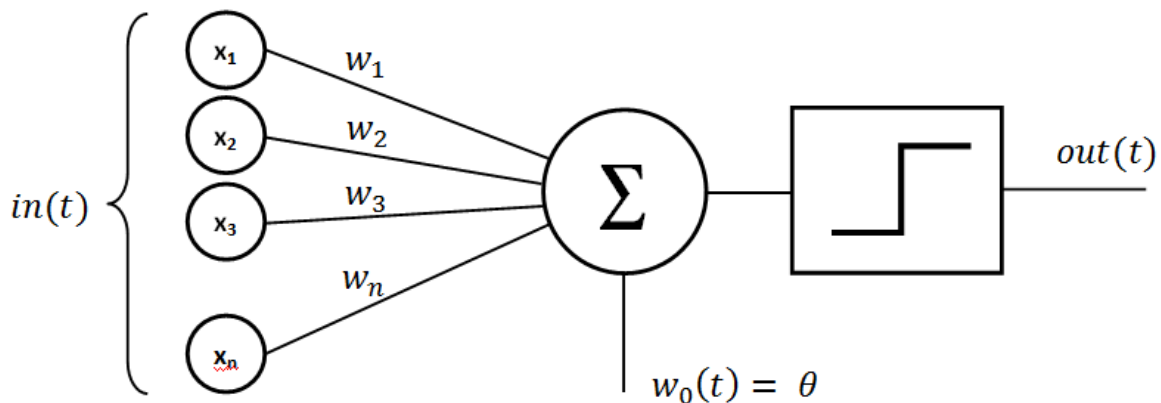
Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks - ANNs) είναι μοντέλα τα οποία έχουν τη δυνατότητα να μαθαίνουν μέσω εκπαίδευσης πολύπλοκα μη γραμμικά όρια αποφάσεων, σε διάφορες εφαρμογές. Έχουν δείξει επανειλημμένα ότι ξεπερνούν σε απόδοση άλλα μοντέλα επεξεργασία γλώσσας. Η σχεδιάσή τους μιμείται τον ανθρώπινο εγκέφαλο ο οποίος αποτελείται από νευρώνες που συνδέονται με άλλους μέσω ινών που λέγονται νευρίτες. Όταν διεγείρεται ένας νευρώνας ανταποκρινόμενος σε ένα ερέθισμα, μεταδίδει νευρικές διεγέρσεις μέσω των νευριτών σε άλλους νευρώνες. Οι νευρώνες-παραλήπτες, λαμβάνουν αυτές τις διεγέρσεις χρησιμοποιώντας δομές, οι οποίες ονομάζονται δενδρίτες, οι οποίοι είναι προεκτάσεις από το σώμα των κυττάρων του νευρώνα. Η ισχύς του σημείου επαφής ανάμεσα σε έναν δενδρίτη και σε έναν νευρίτη, γνωστή ως σύναψη, καθορίζει τη συνεκτικότητα ανάμεσα στους νευρώνες. Οι νευρολόγοι έχουν ανακαλύψει ότι ο ανθρώπινος εγκέφαλος εκπαιδεύεται, αλλάζοντας την ισχύ της συναπτικής σύνδεσης μεταξύ των νευρώνων κατά την επαναλαμβανόμενη διέγερση από το ίδιο ερέθισμα.[1] Η σύνθεση αυτών των απλών λειτουργιών, συνολικά μπορεί εκφράζει πιο πολύπλοκες λειτουργίες. Αυτή η ιδέα αποτέλεσε τη βάση κατασκευής των τεχνητών νευρωνικών δικτύων. Με τρόπο ανάλογο του ανθρώπινου εγκεφάλου, ένα τεχνητό νευρωνικό δίκτυο αποτελείται από ένα σύνολο μονάδων επεξεργασίας, που ονομάζονται κόμβοι και συνδέονται μεταξύ τους μέσω προσανατολισμένων συνδέσμων. Οι κόμβοι αντιστοιχούν σε νευρώνες που εκτελούν τις βασικές μονάδες των υπολογισμών, ενώ οι προσανατολισμένοι σύνδεσμοι αντιστοιχούν σε συνδέσεις μεταξύ νευρώνων, αποτελούμενες από νευρίτες και δενδρίτες. Επιπλέον, το βάρος ενός προσανατολισμένου συνδέσμου που βρίσκεται ανάμεσα σε δύο νευρώνες αναπαριστά την ισχύ της συναπτικής σύνδεσης μεταξύ των νευρώνων. Όπως συμβαίνει στα βιολογικά νευρωνικά συστήματα, ο βασικός στόχος των ANN είναι να προσαρμόζει τα βάρη των συνδέσμων, έως ότου να ταιριάζουν με τις σχέσεις εισόδων-εξόδων των υποκειμένων δεδομένων.

Ο Νευρώνας Perceptron

Ένας νευρώνας Perceptron είναι ένας στοιχειώδης τύπος μοντέλου ANN, ο οποίος περιέχει δύο τύπους κόμβων, τους κόμβους εισόδου, που αναπαριστούν τα χαρακτηριστικά εισόδου και έναν κόμβο εξόδου, ο οποίος χρησιμοποιείται για να αναπαριστά την έξοδο του

μοντέλου. Το παρακάτω σχήμα δείχνει τη βασική αρχιτεκτονική ενός νευρώνα, ο οποίος δέχεται n χαρακτηριστικά εισόδου ($x_1, x_2, \dots, x_n$) και παράγει μία δυαδική έξοδο [1]

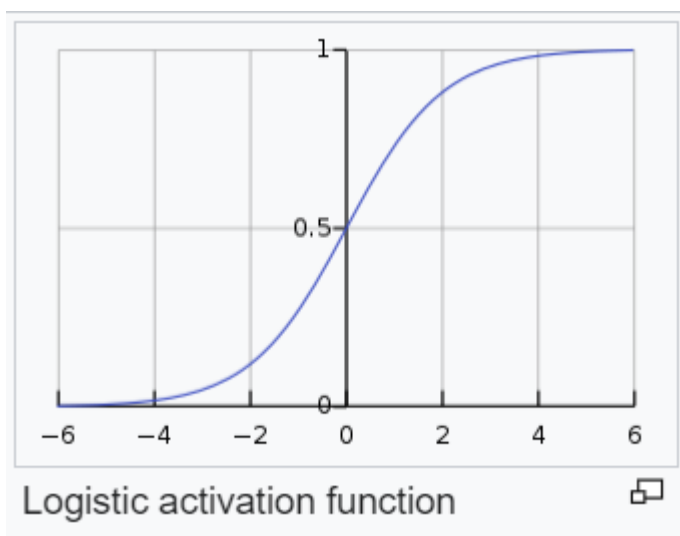
Η έξοδος $out(t) = 1$, αν $w^T \cdot x + b > 0$, αλλιώς $= 0$. Όταν η έξοδος είναι 1, το αποτέλεσμα θεωρείται θετικό ενώ το 0 αρνητικό. Η έξοδος του νευρώνα γράφεται και ως $out(x) = \text{sign}(w^T \cdot x)$ [1], όπου η συνάρτηση sign ονομάζεται συνάρτηση ενεργοποίησης. [4]



Εικόνα 2.1 - Νευρώνας Perceptron

Πηγή: <https://datascientest.com/en/perceptron-definition-and-use-cases>

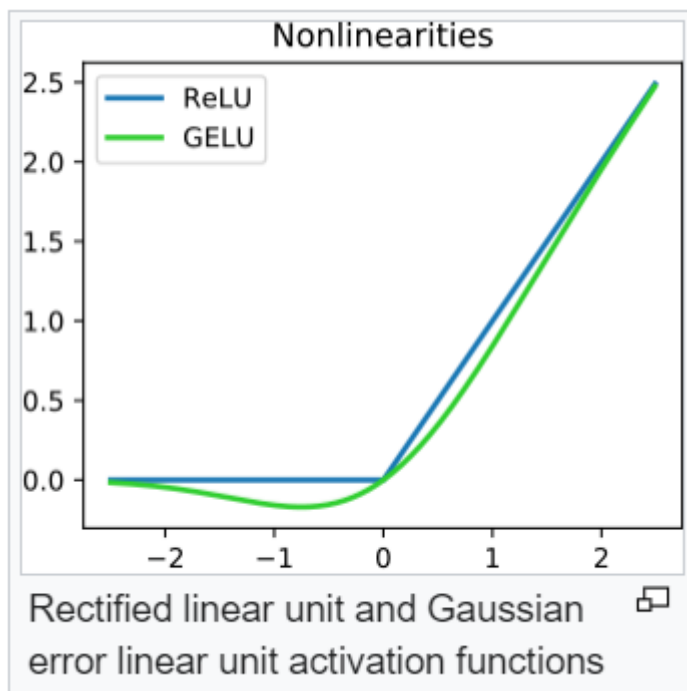
Συνηθισμένες συναρτήσεις ενεργοποίησης είναι οι παρακάτω:



Εικόνα 2.2 - sigmoid function

Πηγή: https://en.wikipedia.org/wiki/Activation_function

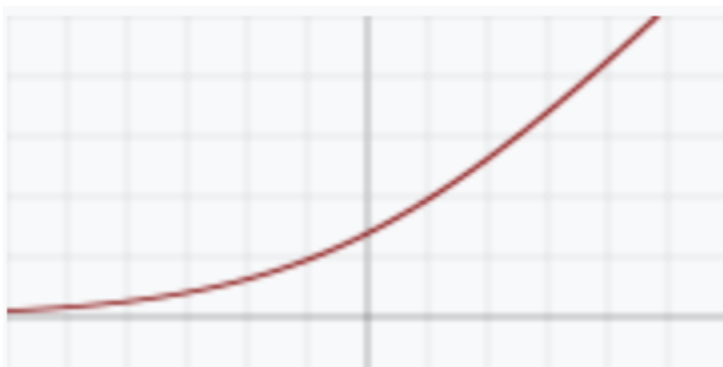
- Λογιστική ή σιγμοειδής (sigmoid): $\sigma(x) = 1/(1+e^{-x})$



Εικόνα 2.3 - ReLU

Πηγή: https://en.wikipedia.org/wiki/Activation_function

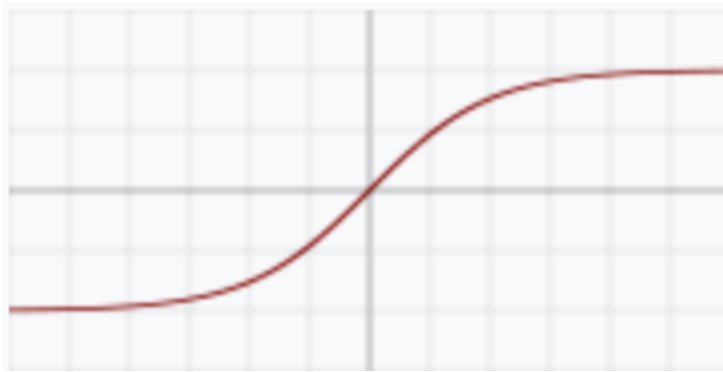
- ReLU (rectified linear unit): $\text{ReLU}(x) = \max(0, x)$



Εικόνα 2.4 - Softplus

Πηγή: https://en.wikipedia.org/wiki/Activation_function

- $\text{softplus}(x) = \ln(1 + e^x)$



Εικόνα 2.5 - Tanh

Πηγή: https://en.wikipedia.org/wiki/Activation_function

- $\tanh(x) = (e^{2x}-1)/(e^{2x}+1)$

Perceptron ενός επιπέδου και πολλών επιπέδων

Ένα Perceptron ενός επιπέδου μπορεί να εκπαιδευτεί να διαχωρίζει μόνο γραμμικές συναρτήσεις. Αντίθετα ένα Perceptron πολλών επιπέδων, που ονομάζεται και δίκτυο

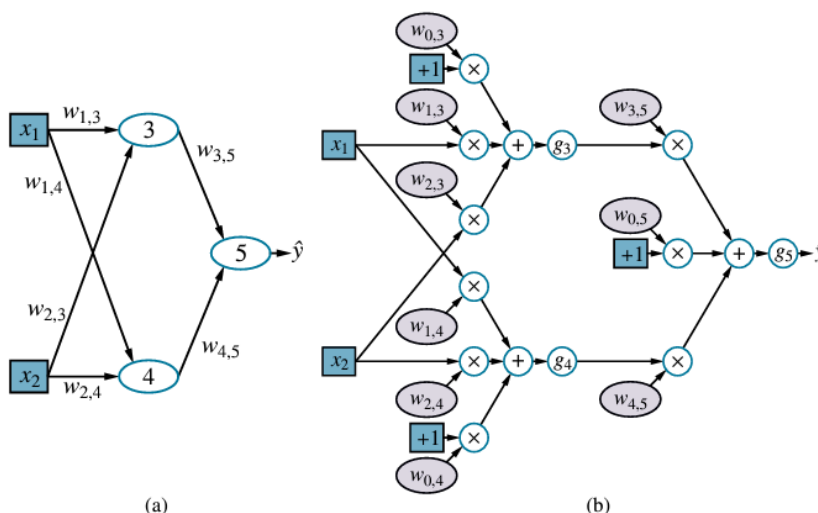


Figure 21.3 (a) A neural network with two inputs, one hidden layer of two units, and one output unit. Not shown are the dummy inputs and their associated weights. (b) The network in (a) unpacked into its full computation graph.

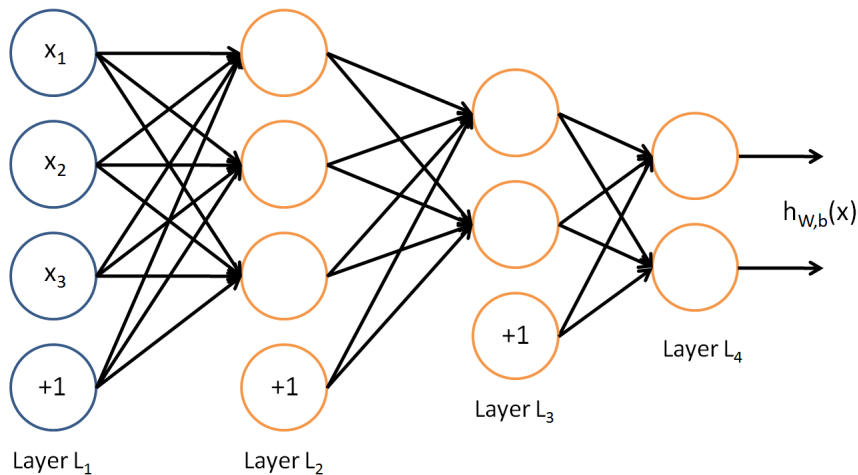
Εικόνα 2.6 - Feed forward neural network

Πηγή: Βιβλίο “Τεχνητή νοημοσύνη: Μια σύγχρονη προσέγγιση”

πρόσθιας τροφοδότησης (feed-forward neural network), ξεπερνάει τον περιορισμό των γραμμικά διαχωρίσιμων συναρτήσεων [4]. Ένα δίκτυο πρόσθιας τροφοδότησης (feedforward network), όπως υποδηλώνει και το όνομά του, έχει συνδέσεις μόνο προς μία κατεύθυνση –δηλαδή, σχηματίζει έναν κατευθυνόμενο ακυκλικό γράφο με κόμβους εισόδου και εξόδου. Κάθε κόμβος υπολογίζει μια συνάρτηση των εισόδων του και μεταβιβάζει το αποτέλεσμα

στους διαδόχους του στο δίκτυο. Οι πληροφορίες ρέουν μέσω του δικτύου από τους κόμβους εισόδου στους κόμβους εξόδου, και δεν υπάρχουν βρόχοι.[2]

Παρακάτω βλέπουμε ένα δίκτυο πρόσθιας τροφοδότησης πολλαπλών επιπέδων [5]



Εικόνα 2.7 - Feed forward neural network

Πηγή: <http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks>

2.3.2 Βαθιά μάθηση

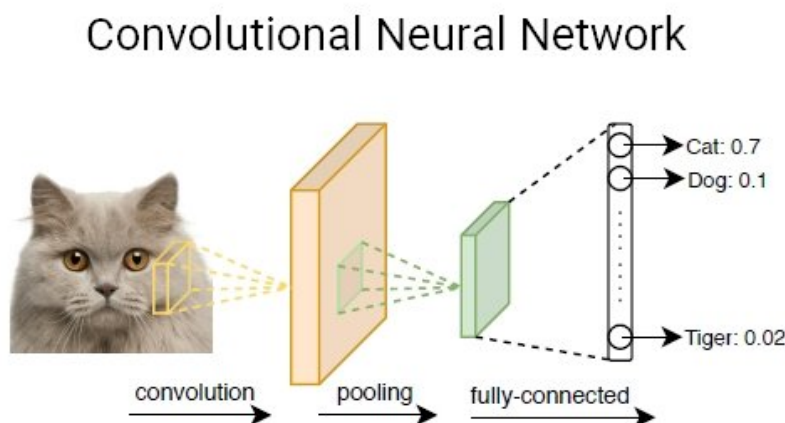
Η βαθιά μάθηση είναι ένα υποσύνολο της μηχανικής μάθησης που χρησιμοποιεί πολυεπίπεδα νευρωνικά δίκτυα, που ονομάζονται βαθιά νευρωνικά δίκτυα, για να προσομοιώσει την ικανότητα σύνθετης λήψης αποφάσεων του ανθρώπινου εγκεφάλου. Με βάση τον ορισμό, ένα βαθύ νευρωνικό δίκτυο ή DNN, είναι ένα νευρωνικό δίκτυο με τρία ή περισσότερα επίπεδα. Στην πράξη, τα περισσότερα DNN έχουν πολύ περισσότερα επίπεδα. Τα DNN εκπαιδεύονται σε μεγάλες ποσότητες δεδομένων για να αναγνωρίζουν και να ταξινομούν φαινόμενα, να αναγνωρίζουν πρότυπα και σχέσεις, να αξιολογούν πιθανότητες και να λαμβάνουν προβλέψεις και αποφάσεις. Τα βαθιά νευρωνικά δίκτυα αποτελούνται από πολλαπλά στρώματα διασυνδεδεμένων κόμβων, όπου το καθένα βασίζεται στο προηγούμενο επίπεδο για να βελτιώσει το αποτέλεσμα. Αυτή η εξέλιξη των υπολογισμών μέσω του δικτύου ονομάζεται διάδοση προς τα εμπρός (forward propagation). Τα στρώματα εισόδου και εξόδου ενός βαθιού νευρωνικού δικτύου ονομάζονται ορατά στρώματα. Το επίπεδο εισόδου είναι εκείνο, όπου το μοντέλο βαθιάς μάθησης δέχεται τα δεδομένα για επεξεργασία και το επίπεδο εξόδου είναι εκείνο, όπου γίνεται η τελική πρόβλεψη ή ταξινόμηση.

Μια άλλη διαδικασία, που ονομάζεται backpropagation, χρησιμοποιεί αλγόριθμους, όπως η gradient descent, για τον υπολογισμό των σφαλμάτων στις προβλέψεις και στη συνέχεια προσαρμόζει τα βάρη και τα biases της συνάρτησης προχωρώντας προς τα πίσω επίπεδα, σε μια προσπάθεια να εκπαιδεύσει το μοντέλο. Μαζί, η προς τα εμπρός διάδοση και η αντίστροφη διάδοση επιτρέπουν σε ένα νευρωνικό δίκτυο να κάνει προβλέψεις και να διορθώνει ανάλογα τυχόν σφάλματα. Με την πάροδο του χρόνου, ο αλγόριθμος γίνεται σταδιακά πιο ακριβής. [6]

Convolutional Neural Networks

Ένα Συνελικτικό Νευρωνικό Δίκτυο, γνωστό και ως CNN ή ConvNet, είναι μια κατηγορία νευρωνικών δικτύων που ειδικεύονται στην επεξεργασία δεδομένων που έχουν τοπολογία πλέγματος, όπως μια εικόνα. Μια ψηφιακή εικόνα είναι μια δυαδική αναπαράσταση οπτικών δεδομένων. Περιέχει μια σειρά από pixel διατεταγμένα σε μορφή πλέγματος που περιέχει τιμές pixel για να υποδηλώσει πόσο φωτεινό και τι χρώμα πρέπει να είναι κάθε pixel.

Αρχιτεκτονική ενός CNN



Εικόνα 2.8 - Convolutional neural network

Πηγή: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn>

Convolution Layer

Αυτό είναι το πρώτο δομικό στοιχείο ενός CNN. Η κύρια μαθηματική εργασία που εκτελείται ονομάζεται συνέλιξη, η οποία είναι η εφαρμογή μιας συνάρτησης συρόμενου παραθύρου σε μια μήτρα pixel που αντιπροσωπεύει μια εικόνα. Η συνάρτηση ολίσθησης που

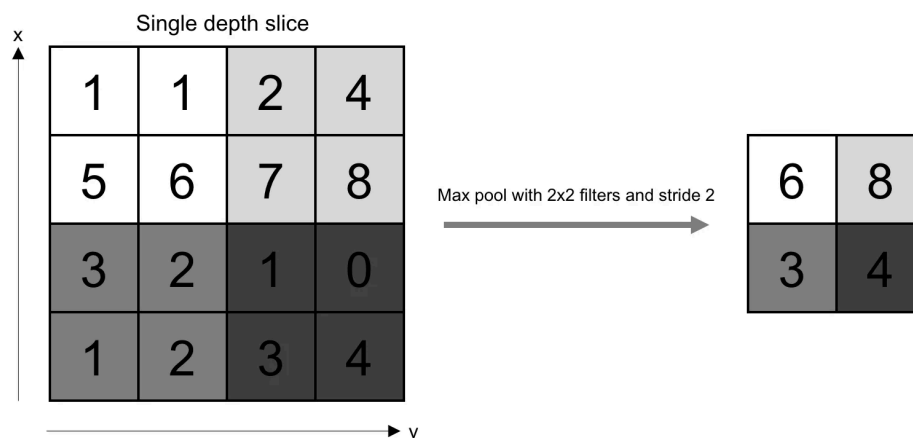
εφαρμόζεται στη μήτρα ονομάζεται πυρήνας ή φίλτρο, όπου και τα δύο μπορούν να χρησιμοποιηθούν εναλλακτικά.

Στο στρώμα συνέλιξης, εφαρμόζονται πολλά φίλτρα ίσου μεγέθους και κάθε φίλτρο χρησιμοποιείται για την αναγνώριση ενός συγκεκριμένου μοτίβου από την εικόνα, όπως η καμπυλότητα των ψηφίων, οι άκρες, ολόκληρο το σχήμα των ψηφίων και πολλά άλλα.

Με απλά λόγια, στο επίπεδο συνέλιξης, χρησιμοποιούμε μικρά πλέγματα (που ονομάζονται φίλτρα ή πυρήνες) που κινούνται πάνω από την εικόνα. Κάθε μικρό πλέγμα είναι σαν ένας μίνι μεγεθυντικός φακός που αναζητά συγκεκριμένα μοτίβα στη φωτογραφία, όπως γραμμές, καμπύλες ή σχήματα. Καθώς κινείται κατά μήκος της φωτογραφίας, δημιουργεί ένα νέο πλέγμα, όπου επισημαίνεται πού βρέθηκαν αυτά τα μοτίβα. [9]

Pooling Layer

Ο στόχος του στρώματος συγκέντρωσης (pooling layer), είναι να αντλήσει τα πιο σημαντικά χαρακτηριστικά από τον convoluted matrix (συνελιγμένος πίνακας) του προηγούμενου επιπέδου. Αυτό γίνεται με την εφαρμογή ορισμένων λειτουργιών συνάθροισης, οι οποίες μειώνουν τη διάσταση του χάρτη χαρακτηριστικών (συνελιγμένος πίνακας), μειώνοντας έτσι τη μνήμη που χρησιμοποιείται κατά την εκπαίδευση του δικτύου. Η συγκέντρωση είναι επίσης σημαντική για τον μετριασμό της υπερ προσαρμογής. [8]



Εικόνα 2.9 - Pooling layer

Πηγή: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn>

Οι πιο συνηθισμένες συναρτήσεις συγκέντρωσης που μπορούν να εφαρμοστούν είναι:

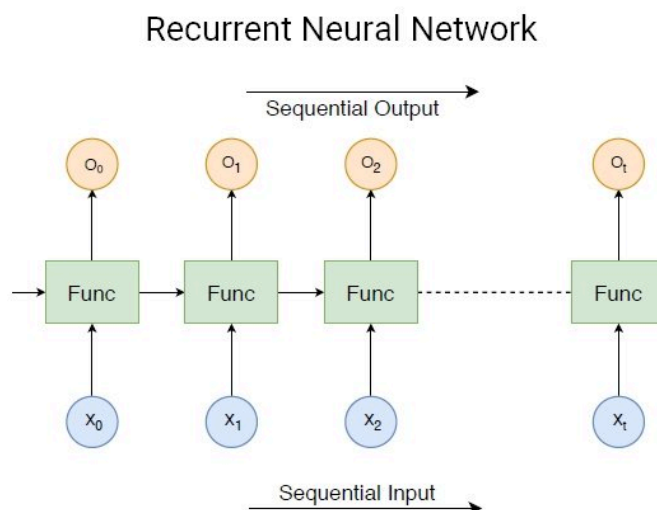
- Μέγιστη συγκέντρωση (max pooling), η οποία είναι η μέγιστη τιμή του χάρτη χαρακτηριστικών
- Η συγκέντρωση αθροίσματος (sum pooling) αντιστοιχεί στο άθροισμα όλων των τιμών του χάρτη χαρακτηριστικών
- Η μέση συγκέντρωση (average pooling) είναι ο μέσος όρος όλων των τιμών.

Fully connected layers

Αυτά τα στρώματα βρίσκονται στο τελευταίο τμήμα του συνελκτικού νευρωνικού δικτύου και οι είσοδοι τους αντιστοιχούν στον πεπλατυσμένο (flattened) μονοδιάστατο πίνακα που δημιουργείται από το τελευταίο στρώμα συγκέντρωσης (pooling layer). Οι συναρτήσεις ενεργοποίησης ReLU εφαρμόζονται σε αυτές για μη γραμμικότητα. Τέλος, ένα επίπεδο πρόβλεψης softmax χρησιμοποιείται για τη δημιουργία τιμών πιθανότητας για καθεμία από τις πιθανές ετικέτες εξόδου και η τελική προβλεπόμενη ετικέτα είναι αυτή με την υψηλότερη πιθανότητα.

Recurrent Neural Network (RNN)

Ένα επαναλαμβανόμενο νευρωνικό δίκτυο (RNN) είναι μια άλλη κατηγορία τεχνητών νευρωνικών δικτύων που χρησιμοποιούν διαδοχική τροφοδοσία δεδομένων. Τα RNN έχουν αναπτυχθεί για να αντιμετωπίσουν το πρόβλημα χρονοσειρών των διαδοχικών δεδομένων



Εικόνα 2.10 - Recurrent Neural Network

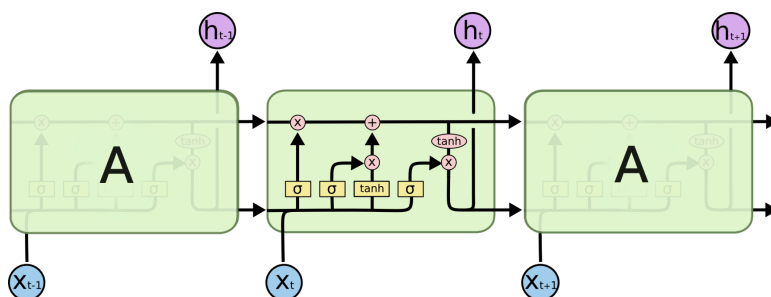
Πηγή: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types>

εισόδου. Η είσοδος του RNN αποτελείται από την τρέχουσα είσοδο και τα προηγούμενα δείγματα. Επομένως, οι συνδέσεις μεταξύ των κόμβων σχηματίζουν ένα κατευθυνόμενο γράφημα κατά μήκος μιας χρονικής ακολουθίας. Επιπλέον, κάθε νευρώνας σε ένα RNN διαθέτει μια εσωτερική μνήμη που διατηρεί τις πληροφορίες του υπολογισμού από τα προηγούμενα δείγματα.[12]

Τα μοντέλα RNN χρησιμοποιούνται ευρέως στην Επεξεργασία Φυσικής Γλώσσας (NLP) λόγω της ικανότητάς τους για επεξεργασία δεδομένων με μήκος εισόδου που δεν είναι σταθερό. Στα RNN, κάθε επόμενο επίπεδο είναι μια συλλογή μη γραμμικών συναρτήσεων σταθμισμένων αθροισμάτων εξόδων και της προηγούμενης κατάστασης. Έτσι, η βασική μονάδα του RNN ονομάζεται «κελί» και κάθε κύτταρο αποτελείται από στρώματα και μια σειρά κυψελών που επιτρέπει τη διαδοχική επεξεργασία επαναλαμβανόμενων μοντέλων νευρωνικών δικτύων.

Long short-term memory

Ένα δίκτυο μακροπρόθεσμης βραχυπρόθεσμης μνήμης (LSTM) είναι ένα επαναλαμβανόμενο νευρωνικό δίκτυο (RNN), που στοχεύει στο να αντιμετωπίσει το πρόβλημα της εξαφάνισης της κλίσης (vanishing gradient problem) [12] που υπάρχει στα παραδοσιακά RNN. Η σχετική του έλλειψη ευαισθησίας είναι το πλεονέκτημά του σε σχέση με άλλα RNN, κρυφά μοντέλα Markov και άλλες μεθόδους εκμάθησης ακολουθίας. Στόχος του είναι να παρέχει μια βραχυπρόθεσμη μνήμη για το RNN που μπορεί να διαρκέσει χιλιάδες χρονικά βήματα, επομένως «μακροπρόθεσμη μνήμη» (long short-term memory). Τα δίκτυα αυτά χρησιμοποιούνται για προβλήματα ταξινόμησης, επεξεργασίας και πρόβλεψης, βάσει χρονοσειρών, όπως σε χειρόγραφα κείμενα, αναγνώριση ομιλίας, μηχανική μετάφραση, ανίχνευση δραστηριότητας ομιλίας, έλεγχο ρομπότ, βιντεοπαιχνίδια, και στην υγεία. [11], [13]

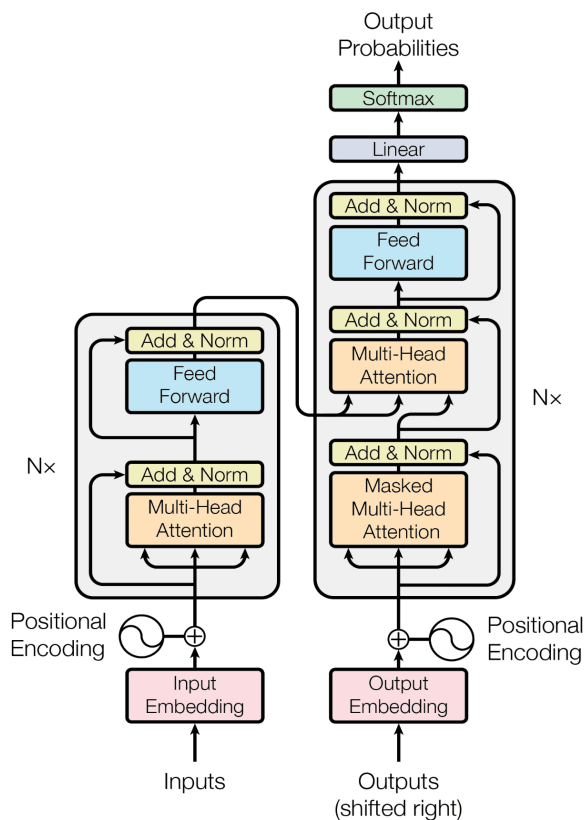


Εικόνα 2.11 - Long-short term memory

Πηγή: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types>

Μετασχηματιστές (Transformers)

Η αρχιτεκτονική του μετασχηματιστή (transformer) ακολουθεί μια δομή



κωδικοποιητή-αποκωδικοποιητή (encoder-decoder), αλλά δεν βασίζεται σε επανάληψη και συνελίξεις για να δημιουργήσει μια έξοδο.

Εικόνα 2.12 - Transformer model

Πηγή: <https://machinelearningmastery.com/the-transformer-model/>

Με λίγα λόγια, το καθήκον του κωδικοποιητή, στο αριστερό μισό της αρχιτεκτονικής του Transformer, είναι να αντιστοιχίσει μια ακολουθία εισόδου σε μια ακολουθία συνεχών αναπαραστάσεων, η οποία στη συνέχεια τροφοδοτείται σε έναν αποκωδικοποιητή.

Ο αποκωδικοποιητής, στο δεξί μισό της αρχιτεκτονικής, λαμβάνει την έξοδο του κωδικοποιητή μαζί με την έξοδο του αποκωδικοποιητή στο προηγούμενο χρονικό βήμα για να δημιουργήσει μια ακολουθία εξόδου. Σε κάθε βήμα το μοντέλο είναι autoregressive, καταναλώνοντας τα σύμβολα που δημιουργήθηκαν προηγουμένως ως πρόσθετη είσοδο κατά τη δημιουργία του επόμενου. [14]

Ο κωδικοποιητής (encoder)

Ο κωδικοποιητής αποτελείται από μια στοίβα έξι όμοιων στρωμάτων (layers), όπου το καθένα αποτελείται από δύο υποστρώματα:

1. Το πρώτο υποστρώμα εφαρμόζει έναν μηχανισμό αυτο-προσοχής (self attention) πολλαπλών κεφαλών.
2. Το δεύτερο υποστρώμα είναι ένα πλήρως συνδεδεμένο δίκτυο (fully connected) που αποτελείται από δύο γραμμικούς μετασχηματισμούς με ενεργοποίηση Rectified Linear Unit (ReLU)

Τα έξι επίπεδα του κωδικοποιητή Transformer εφαρμόζουν τους ίδιους γραμμικούς μετασχηματισμούς σε όλες τις λέξεις της ακολουθίας εισόδου, αλλά κάθε επίπεδο χρησιμοποιεί διαφορετικές παραμέτρους βάρους και bias για να το κάνει.

Κάθε υποστρώμα διαδέχεται επίσης ένα επίπεδο κανονικοποίησης, το οποίο κανονικοποιεί το άθροισμα που υπολογίζεται μεταξύ της εισόδου του υποστρώματος και της εξόδου που δημιουργείται από το ίδιο το υποστρώμα.

Ένα σημαντικό στοιχείο που πρέπει να έχουμε κατά νου είναι ότι η αρχιτεκτονική του Transformer δεν μπορεί να συλλάβει καμία πληροφορία σχετικά με τις σχετικές θέσεις των λέξεων στην ακολουθία, καθώς δεν κάνει χρήση της επανάληψης. Αυτές οι πληροφορίες πρέπει να εισαχθούν με την κωδικοποίηση θέσης στις ενσωματώσεις (embeddings) εισόδου.

Τα διανύσματα κωδικοποίησης θέσης έχουν την ίδια διάσταση με τις ενσωματώσεις εισόδου. Στη συνέχεια, απλώς αθροίζονται στις ενσωματώσεις εισόδου προκειμένου να εισαχθούν οι πληροφορίες θέσης.

Ο αποκωδικοποιητής (decoder)

Ο αποκωδικοποιητής αποτελείται επίσης από μια στοίβα έξι όμοιων στρωμάτων, που το καθένα αποτελείται από τρία υποστρώματα:

1. Το πρώτο υποστρώμα λαμβάνει την έξοδο της προηγούμενης στοίβας του αποκωδικοποιητή, της προσθέτει πληροφορίες θέσης και εφαρμόζει την αυτό-προσοχή πολλών κεφαλών (multi-head self-attention). Ενώ ο κωδικοποιητής έχει σχεδιαστεί για να εξετάζει (attend) όλες τις λέξεις στην ακολουθία εισαγωγής, ανεξάρτητα από τη θέση τους, ο αποκωδικοποιητής εξετάζει μόνο τις προηγούμενες λέξεις.

2. Το δεύτερο επίπεδο υλοποιεί έναν μηχανισμό αυτοπροσοχής πολλαπλών κεφαλών παρόμοιο με αυτόν που εφαρμόζεται στο πρώτο υποστρώμα του κωδικοποιητή. Στην πλευρά του αποκωδικοποιητή, αυτός ο μηχανισμός πολλαπλών κεφαλών λαμβάνει τα ερωτήματα από το προηγούμενο υποστρώμα αποκωδικοποιητή, τα κλειδιά και τις τιμές από την έξοδο του κωδικοποιητή. Αυτό επιτρέπει στον αποκωδικοποιητή να παρακολουθεί όλες τις λέξεις στην ακολουθία εισόδου.
3. Το τρίτο επίπεδο υλοποιεί ένα πλήρως συνδεδεμένο δίκτυο πρόσθιας τροφοδοσίας (feed forward), παρόμοιο με αυτό που υλοποιείται στο δεύτερο υποστρώμα του κωδικοποιητή.

Τα τρία υποστρώματα του αποκωδικοποιητή τα διαδέχεται ένα στρώμα κανονικοποίησης. Επίσης, οι κωδικοποιήσεις θέσης προστίθενται στις ενσωματώσεις εισόδου του αποκωδικοποιητή με τον ίδιο τρόπο που γίνεται στον κωδικοποιητή.

Λειτουργία των μετασχηματιστών (transformers)

Η όλη διαδικασία λειτουργίας του μετασχηματιστή περιγράφεται παρακάτω:

1. Κάθε λέξη της ακολουθίας εισόδου, μετατρέπεται σε ένα διάνυσμα ενσωμάτωσης (embedding) με διάσταση N , που εξαρτάται από το μοντέλο.
2. Το διάνυσμα ενσωμάτωσης που αντιπροσωπεύει τη λέξη εισόδου, αθροίζεται με το διάνυσμα κωδικοποίησης θέσης.
3. Τα επαυξημένα διανύσματα ενσωμάτωσης τροφοδοτούνται στο μπλοκ κωδικοποιητή που αποτελείται από τα δύο υποστρώματα που εξηγήθηκαν παραπάνω. Εφόσον ο κωδικοποιητής εξετάζει όλες τις λέξεις στην ακολουθία εισόδου, ανεξάρτητα από το αν προηγούνται ή διαδέχονται την τρέχουσα λέξη, συμπεραίνουμε ότι ο κωδικοποιητής (encoder) είναι αμφίδρομος.
4. Ο αποκωδικοποιητής λαμβάνει ως είσοδο τη λέξη που είχε προβλέψει ο ίδιος προηγουμένως.
5. Η είσοδος του αποκωδικοποιητή εμπλουτίζεται με κωδικοποίηση θέσης με τον ίδιο τρόπο που γίνεται στην πλευρά του κωδικοποιητή.
6. Η εμπλουτισμένη είσοδος του αποκωδικοποιητή τροφοδοτείται στα τρία υποστρώματα που περιγράφηκαν παραπάνω. Η κάλυψη (masking) εφαρμόζεται στο πρώτο υποστρώμα προκειμένου να σταματήσει ο αποκωδικοποιητής να παρακολουθεί τις επόμενες λέξεις. Στο δεύτερο υποστρώμα, ο αποκωδικοποιητής λαμβάνει επίσης την έξοδο του κωδικοποιητή, ο οποίος τώρα επιτρέπει στον αποκωδικοποιητή να παρακολουθεί όλες τις λέξεις της ακολουθίας εισόδου.

7. Η έξοδος του αποκωδικοποιητή τελικά περνά μέσα από ένα πλήρως συνδεδεμένο στρώμα, ακολουθούμενο από ένα στρώμα softmax, για να δημιουργήσει μια πρόβλεψη για την επόμενη λέξη της ακολουθίας εξόδου.

2.4 Προεπεξεργασία σώματος κειμένων

Όπως αναφέρθηκε και προηγουμένως, η προεπεξεργασία κειμένου είναι η διαδικασία καθαρισμού, μορφοποίησης και μετατροπής ακατέργαστου κειμένου δεδομένων σε μορφή κατάλληλη για εργασίες ΕΦΓ.

Συνήθη βήματα προεπεξεργασίας κειμένου

- Ανάλυση σε σύμβολα (tokenization): Η ανάλυση σε σύμβολα διασπά το κείμενο σε μικρότερες μονάδες όπως λέξεις, σημεία στίξης ή άλλα σημαντικά στοιχεία.
- Κανονικοποίηση: Η κανονικοποίηση τυποποιεί το κείμενο μετατρέποντας όλους τους χαρακτήρες σε πεζά, αφαιρώντας τόνους και ειδικούς χαρακτήρες και διορθώνοντας κοινά ορθογραφικά λάθη.
- Αφαίρεση λέξεων stopwords: Οι λέξεις stopwords είναι κοινές λέξεις όπως "the", "a", και "an" που παρέχουν λίγες σημασιολογικές πληροφορίες. Αυτές οι λέξεις συχνά αφαιρούνται για να μειωθεί ο θόρυβος και να επικεντρωθεί στο πιο ουσιαστικό περιεχόμενο του κειμένου.
- Stemming και lemmatization: Το stemming (εντοπισμός θέματος) και το lemmatization (εντοπισμός λήμματος) μειώνουν τις λέξεις στις ρίζες τους, βοηθώντας στη συσχέτιση σχετικών λέξεων και στη βελτίωση της ακρίβειας των εργασιών ΕΦΓ.
- Αντιστοίχιση σε μέρος του λόγου (POS tagging): Το POS tagging αποδίδει σε κάθε λέξη μιας πρότασης μια γραμματική κατηγορία, όπως ουσιαστικό, ρήμα, επίθετο ή επίρρημα. Αυτές οι πληροφορίες μπορούν να είναι χρήσιμες για εργασίες όπως η αναγνώριση ονομάτων (NER) και η ανάλυση συναισθήματος.
- Επιδιόρθωση ορθογραφίας: Τα εργαλεία διόρθωσης ορθογραφίας μπορούν να χρησιμοποιηθούν για τον εντοπισμό και τη διόρθωση των λέξεων που έχουν γραφτεί λάθος στο κείμενο. Αυτό βοηθά στη βελτίωση της ακρίβειας περαιτέρω ανάλυσης.
- Καθαρισμός δεδομένων: Ο καθαρισμός δεδομένων περιλαμβάνει την αφαίρεση άσχετων ή διπλότυπων δεδομένων, τη διαχείριση ελλειπουσών τιμών και την ανίχνευση και διόρθωση ασυνεπειών στο κείμενο.

Οφέλη της προεπεξεργασίας κειμένου

- Βελτιωμένη απόδοση μοντέλου: Καθαρίζοντας και προετοιμάζοντας τα δεδομένα κειμένου, η προεπεξεργασία κειμένου μπορεί να βελτιώσει την ακρίβεια και τη γενικευσιμότητά των μοντέλων ΕΦΓ.
- Μειωμένα υπολογιστικά κόστη: Το προ επεξεργασμένο κείμενο είναι ευκολότερο να επεξεργαστούν τα μοντέλα ΕΦΓ, γεγονός που μπορεί να οδηγήσει σε πιο γρήγορη και αποδοτική εκπαίδευση και εκτέλεση.
- Βελτίωση ερμηνευσιμότητας: Το προ επεξεργασμένο κείμενο είναι συχνά πιο ευανάγνωστο και κατανοητό, καθιστώντας ευκολότερο να ερμηνευθούν τα αποτελέσματα των μοντέλων ΕΦΓ.

Εργαλεία για την προεπεξεργασία κειμένου

- NLTK (Natural Language Toolkit): Μια ευρέως χρησιμοποιούμενη Python βιβλιοθήκη για εργασίες επεξεργασίας κειμένου.
- SpaCy: Μια άλλη δημοφιλή βιβλιοθήκη Python για NLP, συμπεριλαμβανομένων εργαλείων προεπεξεργασίας κειμένου.
- OpenRefine: Ένα web-based εργαλείο

2.5 Εξαγωγή Χαρακτηριστικών

Στην Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ) μετατρέπουμε το κείμενο σε αριθμούς για δύο κύριους λόγους:

1. Οι υπολογιστές καταλαβαίνουν αριθμούς, όχι κείμενο: Οι αλγόριθμοι μηχανικής μάθησης, που αποτελούν την κινητήρια δύναμη της ΕΦΓ, βασίζονται σε μαθηματικές πράξεις. Το κείμενο, από την άλλη πλευρά, είναι συμβολικό και μη δομημένο. Μετατρέποντας το κείμενο σε αριθμούς, μπορούμε να το τροφοδοτήσουμε σε αυτούς τους αλγόριθμους και να τους κάνουμε να κατανοήσουν τις πληροφορίες που περιέχει.
2. Εύρεση μοτίβων και σχέσεων: Οι αριθμοί μας επιτρέπουν να ποσοτικοποιήσουμε και να συγκρίνουμε διαφορετικές πτυχές του κειμένου. Αυτό είναι ζωτικής σημασίας για εργασίες όπως η ανάλυση συναισθημάτων (εντοπισμός θετικών ή αρνητικών

απόψεων) ή η μοντελοποίηση θεμάτων (ανακάλυψη υποκείμενων θεμάτων σε ένα έγγραφο). Αναπαριστώντας λέξεις ή φράσεις με αριθμητικό τρόπο, μπορούμε να βρούμε μοτίβα και σχέσεις μεταξύ τους, τα οποία θα ήταν δύσκολο ή αδύνατο να εντοπιστούν σε ακατέργαστο κείμενο.

Η επιλογή της κατάλληλης τεχνικής εξαρτάται από την συγκεκριμένη εργασία ΕΦΓ που θέλουμε να εκτελέσουμε. Παράγοντες που επηρεάζουν την επιλογή είναι το μέγεθος του λεξιλογίου, το μήκος των κειμένων και το επιθυμητό επίπεδο λεπτομέρειας.

Η εξαγωγή χαρακτηριστικών αποτελεί σημαντικό βήμα στην ΕΦΓ καθώς βελτιώνει την απόδοση των αλγορίθμων μηχανικής μάθησης. Μετατρέποντας τα ακατέργαστα δεδομένα κειμένου σε πιο δομημένη μορφή, η εξαγωγή χαρακτηριστικών βοηθάει τους αλγορίθμους να μάθουν και να κάνουν πιο ακριβείς προβλέψεις.

Οφέλη της εξαγωγής χαρακτηριστικών στην ΕΦΓ:

- Βελτιωμένη απόδοση: Συμβάλλει στην καλύτερη εκμάθηση και πιο ακριβείς προβλέψεις των αλγορίθμων μηχανικής μάθησης.
- Μειωμένη πολυπλοκότητα: Απλοποιεί τα δεδομένα κειμένου σε πιο δομημένη μορφή, διευκολύνοντας τους αλγορίθμους.
- Αυξημένη ερμηνευσιμότητα: Παρέχει πληροφορίες για τα χαρακτηριστικά που χρησιμοποιούνται στις προβλέψεις, κάνοντας τους αλγορίθμους πιο κατανοητούς.

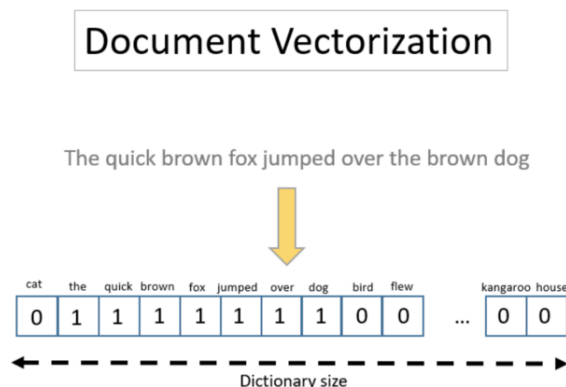
Προκλήσεις της εξαγωγής χαρακτηριστικών στην ΕΦΓ:

- Αραιότητα δεδομένων: Όταν το λεξιλόγιο είναι μεγάλο και τα κείμενα λίγα, μπορεί να δυσκολέψει την εκμάθηση των αλγορίθμων.
- Επιλογή χαρακτηριστικών: Η διαδικασία επιλογής των πιο σημαντικών χαρακτηριστικών από τα δεδομένα μπορεί να είναι δύσκολη καθώς δεν υπάρχει πάντα μοναδικός τρόπος.
- Μηχανική χαρακτηριστικών (Feature engineering): Η διαδικασία μετασχηματισμού των χαρακτηριστικών στα δεδομένα απαιτεί γνώσεις του πεδίου και εφευρετικότητα, καθιστώντας την απαιτητική.

Στην Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ) υπάρχουν διάφορες τεχνικές εξαγωγής χαρακτηριστικών που μπορούμε να αξιοποιήσουμε. Μερικές από τις πιο συνηθισμένες είναι οι παρακάτω:

2.5.1 Bag of Words (Μοντέλο σακούλας λέξεων)

Απλή αλλά αποτελεσματική τεχνική που μετράει πόσες φορές εμφανίζεται κάθε λέξη μέσα σε ένα κείμενο. Με αυτόν τον τρόπο δημιουργούμε ένα διάνυσμα χαρακτηριστικών που αντιπροσωπεύει το κείμενο. Η τεχνική δεν κωδικοποιεί τη σειρά των λέξεων αλλά μόνο τη συχνότητα εμφάνισης [16]. Στο παράδειγμα που ακολουθεί βλέπουμε πως η φράση “The quick brown ...” είναι μονοδιάστατο διάνυσμα όπου για κάθε λέξη υπάρχει το πλήθος των εμφανίσεων [16].

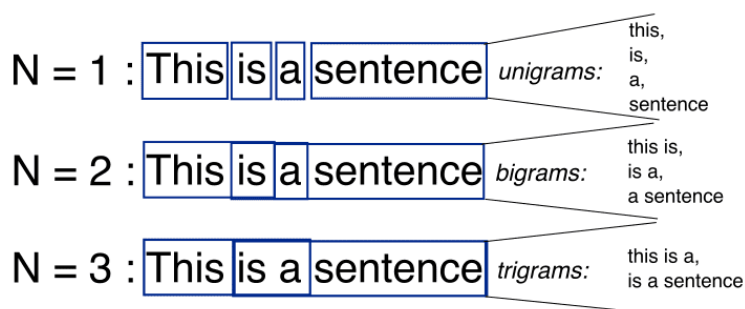


Εικόνα 2.13 - Bag of Words

Πηγή: <https://www.kdnuggets.com/2019/11/text-encoding-review.html>

2.5.2 N-grams (Ακολουθίες n λέξεων)

Τα n-grams μπορούν να μας δώσουν διανύσματα χαρακτηριστικών που αποτυπώνουν τη σειρά εμφάνισης των λέξεων μέσα στο κείμενο. Στο παράδειγμα που ακολουθεί βλέπουμε τα n grams για τιμές του $n = 1, 2, 3$ [17].



Εικόνα 2.14 - N-grams

Πηγή: <https://www.kdnuggets.com/2022/06/ngram-language-modeling-natural-language-processing.html>

Με τα n grams, μειώνεται το μέγεθος του λεξικού σε σχέση με την παραπάνω απλή εφαρμογή BoW, και επιτυγχάνεται βαθύτερη κατανόηση του κειμένου, μέσω της συσχέτισης των λέξεων που παρουσιάζονται μαζί στο κείμενο. Έτσι, ένα μοντέλο 2-gram μπορεί να έχει

καλύτερη απόδοση από την απλή εφαρμογή της συχνότητας εμφάνισης μοναδικών λέξεων (BoW)

2.5.3 TF-IDF

Συντομογραφία του “term frequency-inverse document frequency” (συχνότητα όρου - αντίστροφη συχνότητα εγγράφου). Το TF-IDF είναι μια μέθοδος στάθμισης που αξιολογεί πόσο σημαντική είναι μια λέξη μέσα σε ένα κείμενο, λαμβάνοντας υπόψη ότι μερικές λέξεις εμφανίζονται συχνότερα. [18] π.χ. η συχνότητα df (document frequency) και η idf (inverse document frequency) για κάποιες λέξεις των 37 έργων του Shakespeare είναι οι ακόλουθες [19]

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

Εικόνα 2.15 - TF-IDF

Πηγή: <https://web.stanford.edu/~jurafsky/slp3/14.pdf>

Εδώ βλέπουμε ότι λέξεις “Romeo”, “Falstaff”, και “salad” εμφανίζονται σε πολύ λίγα έργα, οπότε βλέποντας μία από τις λέξεις, μπορεί κάποιος να συμπεράνει με μεγάλη βεβαιότητα σε ποιο έργο αναφέρεται. Αντίθετα, οι λέξεις “good” και “sweet” εμφανίζονται σε κάθε έργο και δεν βοηθούν να συμπαιράνουμε σε ποιο έργο αναφέρονται.

2.5.4 Vector space models (μοντέλα διανυσματικού χώρου)

Τα μοντέλα διανυσματικού χώρου είναι αλγεβρικά μοντέλα που χρησιμοποιούνται για την αναπαράσταση κειμένου (αν και μπορούν να αναπαραστήσουν οποιοδήποτε αντικείμενο) ως διάνυσμα αναγνωριστικών στοιχείων. Με αυτά τα μοντέλα, μπορούμε να διαπιστώσουμε εάν διάφορα κείμενα έχουν παρόμοιο νόημα, ανεξάρτητα από το αν μοιράζονται τις ίδιες λέξεις. [21]

Στην παρακάτω εικόνα (2.16) βλέπουμε, στο αριστερό σχήμα, δύο προτάσεις που ξεκινούν με την ίδια φράση αλλά έχουν διαφορετικό νόημα, ενώ στο δεξιό σχήμα, οι προτάσεις περιέχουν διαφορετικές λέξεις, που όμως έχουν το ίδιο νόημα [21]



Εικόνα 2.16 - Contextual word meaning

Πηγή: <https://towardsdatascience.com/vector-space-models-48b42a15d86d>

Όταν λοιπόν οι προτάσεις του αριστερού σχήματος αναπαρασταθούν με μοντέλο διανυσματικού χώρου, η ομοιότητα συνημιτόνου (cosine similarity) θα είναι μικρή ενώ στη δεξιά θα είναι μεγάλη [22]. Στην παρακάτω εικόνα (2.17) φαίνεται οπτικά αυτή η ιδέα



Εικόνα 2.17 - Cosine similarity

Πηγή: <https://www.exxactcorp.com/blog/Deep-Learning/text-preprocessing-methods-for-deep-learning>

Οι λέξεις “Good”, “Awesome” είναι παρόμοιες και έχουν μικρή γωνία, άρα μεγάλη ομοιότητα συνημιτόνου, το ίδιο και το ζευγάρι “Bad” και “Worst”. Από την άλλη οι λέξεις “Good” και “Bad” που είναι ανόμοιες, έχουν μεγάλη γωνία άρα και μικρή ομοιότητα συνημιτόνου

Word2Vec

Ο αλγόριθμος Word2Vec, ανήκει στα μοντέλα διανυσματικού χώρου και αποσκοπεί στη μετατροπή των λέξεων σε διανύσματα, τα οποία αποτυπώνουν πληροφορίες για την έννοια

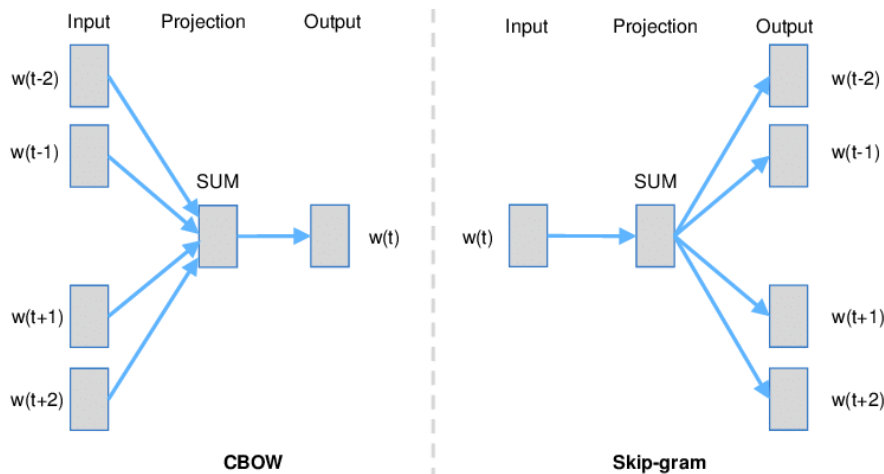
της λέξης, σε σχέση με τις λέξεις που την περικλείουν, στο κείμενο. Ο αλγόριθμος Word2vec υπολογίζει αυτές τις απεικονίσεις μέσω της μοντελοποίησης κειμένου σε ένα μεγάλο σώμα κειμένων (corpus). Ένα εκπαιδευμένο μοντέλο Word2vec μπορεί να εντοπίσει συνώνυμα ή να προτείνει πρόσθετες λέξεις για μια μερική πρόταση. Αναπτύχθηκε από τον Tomáš Mikolov και τους συνεργάτες του στην Google το 2013. Το Word2vec αναπαριστά μια λέξη ως ένα διάνυσμα πολλών διαστάσεων, που αποτυπώνει τις σχέσεις μεταξύ των λέξεων. Συγκεκριμένα, λέξεις που εμφανίζονται σε παρόμοια συμφραζόμενα αντιστοιχούνται σε διανύσματα που βρίσκονται κοντά μεταξύ τους, όπως μετράται από την ομοιότητα συνημιτόνου. Αυτό υποδεικνύει το επίπεδο σημασιολογικής ομοιότητας μεταξύ των λέξεων. Έτσι, για παράδειγμα, τα διανύσματα για τις λέξεις "περπάτημα" και "τρέξιμο" είναι κοντά, όπως είναι και τα διανύσματα για τις λέξεις "αλλά" και "όμως" ή "Βερολίνο" και "Γερμανία". Το Word2vec δεν είναι ένας μεμονωμένος αλγόριθμος, αλλά μια ομάδα συγγενικών μοντέλων που χρησιμοποιούνται για να δημιουργήσουν word embeddings. Αυτά τα μοντέλα είναι απλά νευρωνικά δίκτυα δύο επιπέδων, τα οποία εκπαιδεύονται να ανακατασκευάζουν τα γλωσσικά συμφραζόμενα (context) κάθε λέξης. Το Word2vec δέχεται ως δεδομένα εισόδου ένα μεγάλο σώμα κειμένου (corpus) και παράγει έναν διανυσματικό χώρο, συνήθως με αρκετές εκατοντάδες διαστάσεις. Σε κάθε μοναδική λέξη μέσα στο αρχικό κείμενο αντιστοιχεί ένα διάνυσμα μέσα σε αυτόν τον χώρο.

Το Word2vec μπορεί να αξιοποιήσει δύο διαφορετικές αρχιτεκτονικές μοντέλων για να παράγει αυτές τις κατανεμημένες αναπαραστάσεις λέξεων: το μοντέλο συνεχούς σακούλας λέξεων (CBOW, Continuous Bag-Of-Words) ή το μοντέλο συνεχούς ολισθαίνοντος skip-gram. Και στις δύο αρχιτεκτονικές, το word2vec λαμβάνει υπόψη του τόσο μεμονωμένες λέξεις όσο και ένα ολισθαίνον παράθυρο λέξεων (sliding window), καθώς επαναλαμβάνεται σε όλο το σώμα κειμένου (corpus). [20]

CBOW

Το μοντέλο CBOW μπορεί να θεωρηθεί ως μια εργασία “συμπλήρωσης κενών”, όπου το word embedding μιας λέξης αντιπροσωπεύει τον τρόπο με τον οποίο η λέξη αυτή επηρεάζει τις σχετικές πιθανότητες εμφάνισης άλλων λέξεων μέσα στο ολισθαίνον παράθυρο. Λέξεις που είναι σημασιολογικά παρόμοιες θα πρέπει να επηρεάζουν αυτές τις πιθανότητες με παρόμοιο τρόπο, καθώς σημασιολογικά παρόμοιες λέξεις χρησιμοποιούνται συνήθως σε παρόμοια συμφραζόμενα. Αξίζει να τονιστεί πως η σειρά των λέξεων μέσα στο ολισθαίνον παράθυρο, δεν επηρεάζει την πρόβλεψη (ακολουθώντας την υπόθεση της σακούλας λέξεων -

bag-of-words). Στο σχήμα φαίνεται πως το CBOW προσπαθεί να προβλέψει τη λέξη $w[t]$ με βάση τις συμφοραζόμενες $w[t-2]$, $w[t-1]$, $w[t+1]$, $w[t+2]$. [23]



Εικόνα 2.18 - CBOW vs Skip-gram

Πηγή: <https://www.researchgate.net/profile/Daniel-Braun-6/publication/326588219/figure/fig1/AS:652185784295425@1532504616288/Continuous-Bag-of-words-CBOW-CB-and-Skip-gram-SG-training-model-illustrations.png>

Skip Gram

Στην αρχιτεκτονική skip-gram συνεχούς ροής (continuous skip-gram), το μοντέλο χρησιμοποιεί την τρέχουσα λέξη για να προβλέψει τις συμφοραζόμενες λέξεις στο ολισθαίνον παράθυρο. Αυτή η αρχιτεκτονική δίνει μεγαλύτερη βαρύτητα στις λέξεις του άμεσου περιβάλλοντος σε σχέση με τις πιο απομακρυσμένες λέξεις. Σύμφωνα με τους δημιουργούς του αλγορίθμου, το μοντέλο CBOW είναι ταχύτερο στην εκπαίδευση, ενώ το skip-gram αποδίδει καλύτερα για λέξεις που εμφανίζονται σπάνια στο σώμα κειμένου. [23]

2.6 Μοντέλα ΕΦΓ που βασίζονται στους μετασχηματιστές

2.6.1 BERT

Το μοντέλο γλώσσας BERT είναι ένα framework ανοιχτού λογισμικού, για την επεξεργασία φυσικής γλώσσας (NLP). Το BERT είναι σχεδιασμένο να βοηθά τους υπολογιστές να κατανοούν την έννοια της φυσικής γλώσσας, χρησιμοποιώντας το συμφοραζόμενο κείμενο για να καθορίσει το νόημα των λέξεων. Το BERT εκπαιδεύτηκε χρησιμοποιώντας κείμενα από τη Wikipedia και μπορεί να βελτιωθεί περαιτέρω με δεδομένα ερωτήσεων και απαντήσεων. Προηγουμένως, τα μοντέλα γλώσσας μπορούσαν να διαβάσουν το κείμενο εισόδου μόνο σειριακά - είτε από αριστερά προς τα δεξιά είτε από δεξιά προς τα αριστερά - αλλά δεν μπορούσαν να κάνουν και τα δύο ταυτόχρονα. Το BERT είναι διαφορετικό επειδή έχει σχεδιαστεί να διαβάζει αμφίδρομα και στις δύο κατευθύνσεις. Χρησιμοποιώντας την

αμφίδρομη επεξεργασία κειμένου, το BERT εκπαιδεύτηκε εκ των προτέρων σε δύο διαφορετικά αλλά σχετικά έργα ΕΦΓ, το γλωσσικό μοντέλο απόκρυψης (masked language modeling - MLM) και την πρόβλεψη επόμενης πρότασης (next sentence prediction - NSP).

Ο στόχος της εκπαίδευσης MLM είναι να κρύψει μια λέξη μέσα σε μια πρόταση και στη συνέχεια το μοντέλο να προβλέψει ποια λέξη έχει αποκρυφτεί, βασιζόμενο στα συμφραζόμενα (context) της κρυμμένης λέξης. Ο στόχος της εκπαίδευσης NSP είναι να κάνει το μοντέλο να προβλέψει εάν δύο δεδομένες προτάσεις έχουν μια λογική, διαδοχική σύνδεση ή εάν η σχέση τους είναι απλώς τυχαία [24].

Λειτουργία του BERT

Ο στόχος κάθε τεχνικής Επεξεργασίας Φυσικής Γλώσσας (ΕΦΓ) είναι να κατανοήσει την ανθρώπινη γλώσσα. Για να γίνει αυτό, τα μοντέλα εκπαιδεύονται χρησιμοποιώντας ένα μεγάλο όγκο δεδομένων, στα οποία γλωσσολόγοι επισημαίνουν χειροκίνητα τα δεδομένα. Το BERT, ωστόσο, εκπαιδεύτηκε χρησιμοποιώντας ένα σώμα μη επισημασμένου κειμένου, που περιελάμβανε την Αγγλική Wikipedia και το Brown Corpus. Η διαδικασία αυτή συνεχίζεται από τη Google ώστε το BERT να μαθαίνει μέσω μη επιβλεπόμενης μάθησης από μη επισημασμένο κείμενο και βελτιώνεται καθώς χρησιμοποιείται σε εφαρμογές όπως η αναζήτηση Google. Η προεκπαίδευση του BERT λειτουργεί ως βασικό επίπεδο γνώσης. Από εκεί και πέρα, το BERT μπορεί να προσαρμοστεί σε εξειδικευμένο σώμα κειμένου και μπορεί να βελτιωθεί περαιτέρω σύμφωνα με τις απαιτήσεις του χρήστη. Αυτή η διαδικασία είναι γνωστή ως μεταφορά μάθησης (transfer learning).

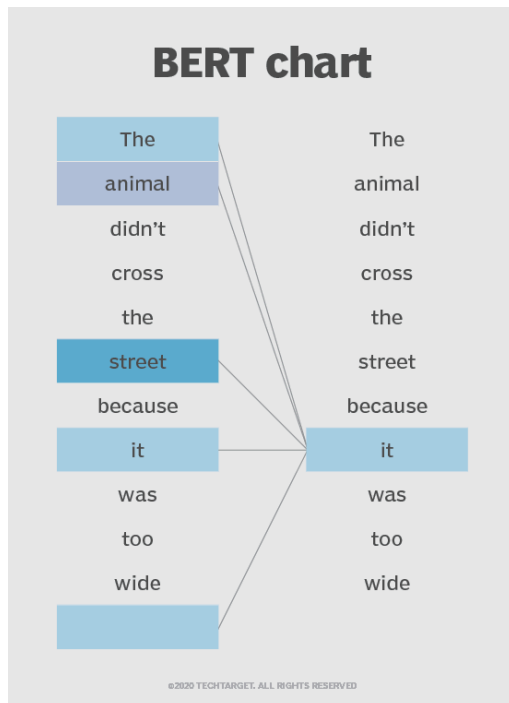
Η δημιουργία του BERT κατέστη δυνατή με την εργασία της Google πάνω στους μετασχηματιστές (transformers). Ο μετασχηματιστής είναι το τμήμα του μοντέλου που δίνει στο BERT την αυξημένη του ικανότητα να κατανοεί τα συμφραζόμενα και την αμφισημία στη γλώσσα. Ο μετασχηματιστής επεξεργάζεται κάθε λέξη σε σχέση με όλες τις άλλες λέξεις σε μια πρόταση, αντί να τις επεξεργάζεται μία προς μία. Εξετάζοντας όλες τις γύρω λέξεις, ο μετασχηματιστής επιτρέπει στο BERT να κατανοήσει το πλήρες πλαίσιο (συμφραζόμενα) της λέξης [24]. Το Word2Vec παράγει ενσωματώσεις (embeddings) που είναι ανεξάρτητες από τα συμφραζόμενα, επομένως κάθε λέξη αναπαρίσταται ως ένα μόνο διάνυσμα. Από την άλλη, η αμφίδρομη κωδικοποίηση του BERT, του επιτρέπει να λαμβάνει υπόψη τη θέση κάθε λέξης σε μια ακολουθία και να την περιλαμβάνει στην ενσωμάτωση αυτής της λέξης, κάτι που το Word2Vec δεν μπορεί να κάνει. [25].

Από την αρχιτεκτονική των μετασχηματιστών, το BERT χρησιμοποιεί μόνο το τμήμα του κωδικοποιητή (encoder), καθώς αυτό λειτουργεί καλύτερα σε εργασίες που απαιτούν την κατανόηση ολόκληρης της πρότασης. Το BERT χρησιμοποιείται σε εργασίες όπως:

- Απάντηση ερωτήσεων
- Σύνοψη κειμένου
- Πρόβλεψη επόμενης πρότασης
- Άρση αμφισημίας λέξης
- Ταξινόμηση κειμένου, κ.α

Μηχανισμός αυτο-προσοχής (self-attention)

Το BERT βασίζεται επίσης σε μηχανισμό αυτό-προσοχής (self-attention) που συλλαμβάνει και κατανοεί τις σχέσεις μεταξύ των λέξεων σε μια πρόταση, χάρη στην αμφίδρομη ικανότητα των μετασχηματιστών. Αυτό είναι σημαντικό, επειδή συχνά μια λέξη μπορεί να αλλάξει νόημα καθώς αναπτύσσεται μια πρόταση. Κάθε λέξη που προστίθεται, αυξάνει το συνολικό νόημα της λέξης στην οποία εστιάζει ο αλγόριθμος. Όσο περισσότερες λέξεις υπάρχουν σε κάθε πρόταση, τόσο πιο αμφίσημη γίνεται η εξεταζόμενη λέξη. Το BERT, διαβάζοντας αμφίδρομα, λαμβάνει υπόψη την επίδραση όλων των άλλων λέξεων σε μια πρόταση, στη λέξη εστίασης.



Εικόνα 2.19 - BERT self-attention

Πηγή: <https://www.techtargget.com/searchenterpriseai/definition/BERT-language-model/>

Στην εικόνα 2.19, παραπάνω, το BERT προσπαθεί να καταλάβει σε ποια λέξη αναφέρεται το “it” και εφαρμόζοντας την “αυτο-προσοχή” να ζυγίσει τις επιλογές. Η λέξη με το μεγαλύτερο υπολογισμένο βαθμό θεωρείται η επικρατέστερη. Σε αυτό το παράδειγμα το BERT θα επιλέξει σωστά τη λέξη “animal” αντί της λέξης “street” [24].

2.6.2 Άλλα μοντέλα που βασίζονται σε Transformer encoders

RoBERTa

Η ερευνητική ομάδα του Facebook AI διαπίστωσε ότι το BERT ήταν σημαντικά υπο εκπαιδευμένο και πρότεινε μια βελτιωμένη έκδοση, που ονομάστηκε RoBERTa. Για τη βελτίωση στην εκπαίδευση:

1. Χρησιμοποιήθηκαν 160GB κειμένου αντί του αρχικού συνόλου των 16GB που χρησιμοποιήθηκαν για την εκπαίδευση του BERT.
2. Αυξήθηκε ο αριθμός των επαναλήψεων από 100 χιλιάδες σε 300 χιλιάδες και στη συνέχεια σε 500 χιλιάδες.
3. Χρησιμοποιήθηκαν μεγαλύτερες παρτίδες (batches) 8 χιλιάδες αντί για 256 στο αρχικό βασικό μοντέλο BERT.
4. Μεγαλύτερο λεξιλόγιο BPE σε επίπεδο byte με 50 χιλιάδες υπομονάδες λέξεων αντί για λεξιλόγιο BPE σε επίπεδο χαρακτήρα μεγέθους 30 χιλιάδων.
5. Αφαιρέθηκε ο στόχος πρόβλεψης επόμενης πρότασης (next sentence prediction).
6. Εφαρμόστηκε δυναμική επιλογή του μοτίβου απόκρυψης λέξης, στα δεδομένα εκπαίδευσης.

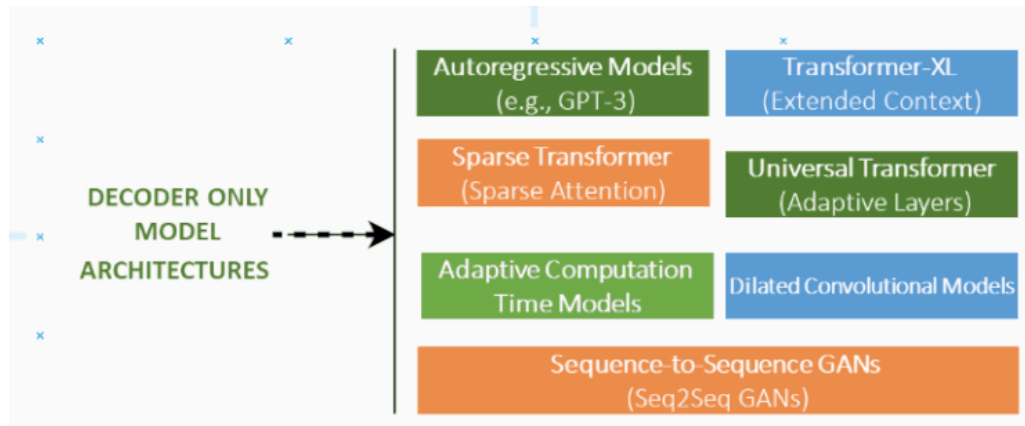
Το αποτέλεσμα ήταν το μοντέλο RoBERTa να ξεπερνάει το BERT σε όλα τα GLUE benchmarks [27].

ALBERT

Για να αντιμετωπίζει το πρόβλημα του συνεχώς αυξανόμενου μεγέθους των προεκπαιδευμένων μοντέλων γλώσσας, το οποίο οδηγεί σε περιορισμούς μνήμης, μεγαλύτερο χρόνο εκπαίδευσης και μερικές φορές σε απροσδόκητα μειωμένη απόδοση η ερευνητική ομάδα της Google εισήγαγε την αρχιτεκτονική ALBERT (A Lite BERT), που περιέχει τις τεχνικές μείωσης παραμέτρων “factorized embedding parameterization” και “cross-layer parameter sharing”. Το αποτέλεσμα ήταν ένα μοντέλο με 18x λιγότερες παραμέτρους και 1,7x ταχύτερο χρόνο εκπαίδευσης σε σχέση με το BERT-large, διατηρώντας όμως σχεδόν ίδια απόδοση στα GLUE benchmarks [27].

2.6.3 Μοντέλα που βασίζονται σε Transformer decoders

Σε αντίθεση με τα μοντέλα που βασίζονται σε encoders (π.χ BERT) και είναι πολύ καλά στην κατανόηση του κειμένου, τα μοντέλα που βασίζονται σε decoders είναι ικανά να δημιουργούν νέο κείμενο.



Εικόνα 2.20 - Decoder-only models

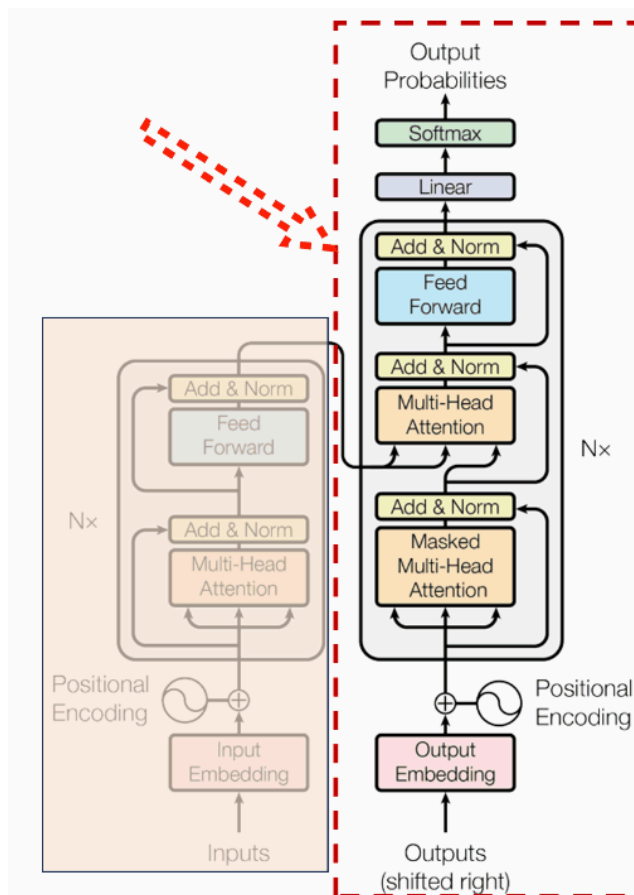
Πηγή: <https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-qi6vc/>

Μερικά από τα πιο γνωστά μοντέλα της κατηγορίας είναι τα GPT-3, ChatGPT, GPT-4, PaLM, LaMDa και Falcon. Η ισχύς των μοντέλων αποκωδικοποίησης (decoder-only) έγκειται στην ικανότητά τους όχι μόνο να μιμούνται κείμενο που μοιάζει ανθρώπινο, αλλά και να είναι δημιουργικά στις απαντήσεις τους. Μπορούν να δημιουργούν ιστορίες, να απαντούν σε ερωτήσεις και ακόμη και να συμμετέχουν σε διαλόγους που ακούγονται φυσικοί και ευχάριστοι. Αυτή η ικανότητα τα καθιστά εξαιρετικά χρήσιμα σε ένα ευρύ φάσμα εφαρμογών, από chatbots και ψηφιακούς βοηθούς έως τη δημιουργία περιεχομένου, την περίληψη περιεκτικού χαρακτήρα και την αφήγηση ιστοριών.

Η βασική αρχιτεκτονική ενός μοντέλου αποκωδικοποίησης (decoder-only model) είναι σχετικά απλή (εικόνα 2.21). Περιλαμβάνει συνήθως [28]:

- Το Token Input Layer. Εδώ το μοντέλο λαμβάνει την ακολουθία συμβόλων (token) εισόδου.
- Το Decoder Block Layer. Αυτό το επίπεδο αποτελεί τον πυρήνα του μοντέλου και περιέχει διάφορα στοιχεία:
 - Επίπεδο αυτο-προσοχής με πολλαπλές κεφαλές και μάσκα (Masked Multi-Head Self-Attention Layer): Αυτό είναι απαραίτητο για την κατανόηση της ακολουθίας εισόδου. Η "μάσκα" διασφαλίζει ότι η πρόβλεψη για ένα διακριτικό δεν λαμβάνει υπόψη μελλοντικά διακριτικά, διατηρώντας την αυτοαναδρομική φύση του μοντέλου.

- Πλήρως συνδεδεμένα στρώματα (Fully Connected Layers): Αυτά τα στρώματα επεξεργάζονται περαιτέρω τις πληροφορίες.
- Κωδικοποίηση θέσης (Positional Encoding): Αυτό προσθέτει πληροφορίες σχετικά με τη θέση κάθε διακριτικού στην ακολουθία, κάτι που είναι απαραίτητο για την κατανόηση της σειράς των λέξεων.
- Τελευταίο επίπεδο (Last Layer): Ένα πλήρως συνδεδεμένο στρώμα ακολουθούμενο από μια συνάρτηση SoftMax χρησιμοποιείται σε ολόκληρο το λεξιλόγιο για την πρόβλεψη της επόμενης λέξης στην ακολουθία.



Εικόνα 2.21 - Decoder-only architecture

Πηγή: <https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-qi6vc/>

Με βάση τις απαιτήσεις, ένα μοντέλο αποκωδικοποίησης (decoder-only model) μπορεί να έχει πολλά μπλοκ αποκωδικοποίησης (Decoder Block) στοιβαγμένα το ένα πάνω στο άλλο. Η πραγματική κινητήρια δύναμη πίσω από αυτά τα μοντέλα είναι ο μηχανισμός αυτο-προσοχής με μάσκα (Masked Self-Attention). Αυτός ο μηχανισμός επιτρέπει στο μοντέλο να εστιάζει σε διαφορετικά μέρη της ακολουθίας εισόδου όταν προβλέπει κάθε διακριτικό, διευκολύνοντας την παραγωγή κειμένου με συνάφεια στο περιεχόμενο. Τα μοντέλα αποκωδικοποίησης προ εκπαιδεύονται σε ένα τεράστιο σώμα κειμένου, που είναι διαθέσιμο στο διαδίκτυο. Ο κύριος στόχος κατά τη διάρκεια αυτής της φάσης

προεκπαίδευσης είναι η πρόβλεψη της επόμενης λέξης σε κάθε ακολουθία κειμένου. Αυτή η εκτεταμένη εκπαίδευση επιτρέπει στο μοντέλο να κατανοήσει και να δημιουργήσει κείμενο που μοιάζει ανθρώπινο. Στη συνέχεια, αυτά τα μοντέλα μπορούν να βελτιωθούν περαιτέρω για συγκεκριμένες εργασίες. Αυτή η περαιτέρω βελτίωση, προσαρμόζει το μοντέλο για εφαρμογές όπως συστήματα απάντησης σε ερωτήσεις, εικονικούς βοηθούς ή συστήματα βασισμένα σε διαλόγους.

3. Μεθοδολογία υλοποίησης

3.1 Περιγραφή Συνόλου Δεδομένων (Dataset)

Μετά από ενδελεχή αναζήτηση, επέλεξα ως σώμα δεδομένων ένα αρχείο που περιέχει επισημειωμένα Tweets, τα οποία έχουν αναρτηθεί από Έλληνες χρήστες της πλατφόρμας κοινωνικής δικτύωσης X, πρώην Twitter. Το αρχείο αυτό δημιουργήθηκε στα πλαίσια μεταπτυχιακής εργασίας [1] τον Ιούνιο του 2019. Με τη βοήθεια βιβλιοθηκών της Python ο συγγραφέας επέλεξε να κατεβάσει tweets για δημοφιλή θέματα της περιόδου εκείνης, με κυριότερο τις προσεχείς δημοτικές και βουλευτικές εκλογές. Επίσης, για να αυξήσει το ποσοστό των tweets με προσβλητικό περιεχόμενο, πραγματοποίησε αναζητήσεις με βάση υβριστικές λέξεις που χρησιμοποιούνται συχνά από τους Έλληνες χρήστες των κοινωνικών δικτύων. Το αρχικό σύνολο αποτελούνταν από 49.154 tweets. Στη συνέχεια πραγματοποιήθηκε αρχική εκκαθάριση του κειμένου, απ' όπου αφαιρέθηκαν τα URLs, τα Emojis και τα Emoticons. Επίσης τα ονόματα χρηστών και οι αναφορές σε αυτούς (mentions) αντικαταστάθηκαν από τη λέξη *@User*. Σύμφωνα με τον δημιουργό του dataset, συνεχόμενες εμφανίσεις συμβόλων, όπως ερωτηματικά, θαυμαστικά, κλπ αντικαταστάθηκαν από ένα σύμβολο, π.χ τα !!! γίνονται !. Όμως μετά από έλεγχο του αρχείου, παρατήρησα ότι κάποιοι επαναλαμβανόμενοι χαρακτήρες δεν είχαν αντικατασταθεί από ένα και έτσι το υλοποίησα εγώ. Στη συνέχεια επιλέχθηκαν τυχαία 5.000 tweets τα οποία επισημείωσαν Ελληνόφωνοι εθελοντές. Για κάθε tweet υπήρχαν τρεις επισημειώσεις και ως αποτέλεσμα λήφθηκε η πλειοψηφία.

ID	Tweet	Label
2517	Τατουάζ = λέπρα τάδε έφη κύριος Αραπαντζής #giatinparea	Offensive
3225	Οι γυναίκες που δεν κάνουν στοματικό έρωτα κινδυνεύουν από κατάθλιψη!	Not Offensive
3672	Μας τα επηρέξες όμως με το μου έχει πει η Βίκυ το ένα, η Βίκυ το άλλο. #shoppingstar	Not Offensive
3673	Έλαβε το μήνυμα λέει ο Τσίπρας. Νταξ πριν καλά καλά του το στείλουν, αυτός το έχει λάβει, το έχει διαβάσει, το έχει κατανοήσει και έχει καταστρώσει και το ανάλογο σχέδιο για να κερδίσει. Πουτάνα όλα #ekloges2019	Offensive
3676	@USER Όχι δε βρίζεις ούτε είσαι τραμπούκος! Γιατί τα έσβησες αυτά; Το παίζεις συζητήσιμος μέχρι που δεν έχεις απάντηση γυρνάς σε βρισιδί. Του είπα ότι είναι μισογύνες όταν είπε ότι οι γυναίκες ρίχνουν παντόφλα γι αυτό είναι καλύτερες στο στόχο αυτή ήταν η απάντηση. #survivorgr	Not Offensive
3697	Καλησπερα και καλη εβδομαδα τουιτερο-στυλιστριες!Να και μια φορα που καταφερα ναμια κι εγω μαζί σας! Τι βλέπουμε; Ποιο είναι αυτο το νευρωσικο παρτσάκιο; #shoppingstar	Offensive
3703	Απόλυτη κατάρρευση για τον ΣΥΡΙΖΑ – Κάνει τα πάντα για ψήφους ο Τσίπρας! Ρομά, πρόσφυγες όσοι «δεν έχουν σταθερό» οι ψηφοφόροι του	Not Offensive
3727	Το σκατουλίνι ήθελε βραχιόλι και αυτό κιόλας; Αυτή δεν ήθελε πριν να πάρει σκουλαρίκια; Πόσο διπρόσωπη ρε φίλε; #shoppingstar	Offensive
3823	Ο καινούριος βρωμαει απο χιλιομετρα μπουρτζοβλαχος καγκουρομαγκας χαζοναρκισσος. Αλλα ευτυχως μας ξεκαθαρισε σιι ολες τον θελουν κι αυτος τις αφηνει και γλυτωσαμε την καψουρα του #PowerOfLoveGR	Offensive
4004	Αλιμονο η Σπυριδουλα εχει να διαχειριστεί και μια δημοσιότητα τωρα! περιζήτητη παντου! που καιρος για γαμους αυτα είναι για τον κοσμάκη! δεν προλαβαίνει η Σταρουμπα να ραφτεί στη VERA WANG! αει μαρη! #masterchefgr_#eftiles #giatinparea #survivorgr #starkoukou #MasterChefGR	Not Offensive
4066	Μόνο για άντρες Όταν συζητάτε με γυναίκα που σας αρέσει(όχι σχέση), τη σκέφτεστε σεξουαλικά;	Not Offensive
4453	ρε μαλακα ΜΟΝΟ η δαλακα ακουγεται να σχολιαζει αρνητικά! ΕΙΣΑΙ Η ΜΟΝΑΔΙΚΗ ΑΘΛΗΤΡΙΑ ΕΝ ΕΝΕΡΓΕΙΑ ΕΚΕΙ ΜΕΣΑ! και εισαι η μονη που ακουγεσαι να σχολιαζει αρνητικά! #survivorGR	Not Offensive
4813	Τρέιλερ #MasterChefGR ,μετά το φαγητό...άμα βλέπεις την πατσαβούρα,δεν το γλιτώνεις το ξερατο..#giatinparea #ShoppingStar #poweroflovegr	Offensive
2	Πάντως επειδή θυμάμαι πολυ καλα αυτα που ειχε πει η Κώνσταντινα μόνο αν ήσουν βλάκας (η παρουσιάζεις εκπομπή) δεν θα καταλάβαινες πως όντως έκανε πλάκα η κοπέλα και αυτοτρολιαριζοταν αλλα ως γνωστόν απ την μίμνα βίνκι στα κανάλια #GiatiNparea	Offensive

Εικόνα 3.1 - Απόσπασμα από τα tweets Dataset (πριν την επεξεργασία)

Το σύνολο δεδομένων αποτελείται από τρεις στήλες, έναν αύξοντα αριθμό (ID), το κείμενο του tweet (Tweet) και η επισημείωση (Label). Οι τιμές της στήλης Label είναι είτε “Offensive” ή “Not Offensive”. Το σύνολο αυτό δεν είναι ισορροπημένο (imbalanced), διότι τα tweets που χαρακτηρίζονται προσβλητικά είναι 1.401 ενώ τα μη προσβλητικά 3.378. Όπως φαίνεται τα tweets είναι συνολικά 4.779, καθώς μερικά από τα αρχικά απορρίφθηκαν ως spam. Στο παραπάνω δείγμα αξίζει να παρατηρήσουμε ότι τα hashtags (#κείμενο) δεν έχουν αφαιρεθεί, καθώς ο συγγραφέας θεωρεί, ορθά, ότι παίζουν ρόλο στην ταξινόμηση.

3.2 Περιγραφή προεπεξεργασίας δεδομένων, στο dataset που χρησιμοποιήθηκε

Το αρχικό σύνολο δεδομένων είναι σε κωδικοποίηση utf-8 την οποία δεν τροποποίησα, καθώς το κείμενο περιέχει κυρίως Ελληνικές λέξεις, και ο tokenizer που χρησιμοποίησα μπορεί να χειριστεί κείμενο αυτής της μορφής. Οι αλλαγές που έκανα περιγράφονται αναλυτικότερα στη συνέχεια.

3.2.1 Μετατροπή σε πεζά

Αρχικά μετέτρεψα τα κεφαλαία σε πεζά. Με αυτό τον τρόπο το λεξικό που παράγεται είναι μικρότερο και το νόημα δεν αλλάζει σημαντικά, καθώς η χρήση των κεφαλαίων στην Ελληνική γλώσσα γίνεται στην αρχή μιας νέας πρότασης και σε ονόματα. Η αρχή μιας πρότασης όμως συμπεραίνεται και από σύμβολα που προηγούνται, όπως η τελεία, το θαυμαστικό και το ερωτηματικό, τα οποία δεν έχω αφαιρέσει από το αρχικό κείμενο. Από την άλλη, χρησιμοποιώντας μοντέλα που βασίζονται στο BERT, οι λέξεις που αναφέρονται σε ονόματα γίνονται κατανοητές ως τέτοιες και χωρίς τη χρήση κεφαλαίων στη θέση του πρώτου χαρακτήρα. Ο σημαντικότερος λόγος όμως είναι ότι τα κείμενα στα οποία θα γίνει προσαρμογή (fine tuning) προέρχονται από το X (πρώην twitter) και σε αυτές τις πλατφόρμες οι χρήστες δεν προσέχουν ιδιαίτερα τη σωστή γραμματική, και σύνταξη της γλώσσας και συχνά ολόκληρες προτάσεις είναι γραμμένες με κεφαλαία, ή στη μέση μιας λέξης εμφανίζονται ένα ή περισσότερα κεφαλαία γράμματα. Η υλοποίηση γίνεται με τη χρήση της ενσωματωμένης συνάρτησης `lower()`, της γλώσσας Python.

3.2.2 Αφαίρεση τόνων και διακριτικών

Στη συνέχεια αφαιρώ τους τόνους και τα διακριτικά. Στη σύγχρονη Ελληνική από τα διακριτικά χρησιμοποιούνται τα διαλυτικά, χωρίς να αποκλείεται όμως κάποιος χρήστης να έχει βάλει οξείες, δασείες κλπ. Το σκεπτικό είναι παρόμοιο με τη μετατροπή σε πεζά, δηλαδή το λεξικό που παράγεται να είναι μικρότερου μεγέθους, χωρίς να επηρεάζεται το νόημα των προτάσεων. Και εδώ η αφαίρεση των τόνων και των διακριτικών, δεν αλλάζει τη σημασία των λέξεων. Όπως επισημάνθηκε και προηγουμένως οι χρήστες σε αυτές τις πλατφόρμες δεν δίνουν ιδιαίτερη σημασία στη σωστή γραμματική. Για την υλοποίηση χρησιμοποίησα τις συναρτήσεις `normalize` και `translate`, της βιβλιοθήκης `unicodedata`, περνώντας τις κατάλληλες παραμέτρους.

3.2.3 Άλλες αλλαγές και αποφάσεις

Τα κείμενα των κοινωνικών δικτύων περιέχουν και ασυνήθιστους συνδυασμούς χαρακτήρων όπως τα emoticons, π.χ η ακολουθία :) συμβολίζει ένα χαμογελαστό πρόσωπο για το οποίο υπάρχει και σύμβολο που ονομάζεται emoji, με περιγραφή “grinning face” και κωδικό 1F60, σε δεκαεξαδικό σύστημα. Αυτά έχουν ήδη αφαιρεθεί από το αρχικό dataset, οπότε δε χρειάστηκε να κάνω καθαρισμό τέτοιου είδους. Επίσης, στο αρχικό dataset έχουν αντικατασταθεί όλες οι αναφορές σε χρήστες με τη φράση @User, την οποία αποφάσισα να αφαιρέσω, καθώς δε φαινόταν να αλλάζει κάτι στα αποτελέσματα της προσαρμογής. Τα

URLs και οι χαρακτήρες retweet είχαν και αυτά αφαιρεθεί. Αποφάσισα να κάνω διαφορετικές δοκιμές, σε κάποιες αφήνοντας τα hashtags και σε κάποιες αφαιρώντας τα. Η λογική πίσω από αυτή την επιλογή, είναι ότι θεωρώ πολύ πιθανόν να επηρεάζουν την ταξινόμηση των προτάσεων, όπως φαίνεται από τα παραδείγματα #MasterChefGR_Xeftiles, #ΣΥΡΙΖΑ_ξεφτιλες, #πρεζεμπορας, #ερχεται_ΚΑΡΑΠΙΟΥ***ΚΛΑΡΑ κ.α. Ακόμη, δεν αφαίρεσα τα σημεία στίξης, όπως κόμματα, ερωτηματικά, θαυμαστικά κλπ γιατί αυτά επηρεάζουν το νόημα των προτάσεων, π.χ το ! δίνει έμφαση στην πρόταση. Τέλος, αποφάσισα να μην αφαιρέσω τα stopwords γιατί χρησιμοποιώ μοντέλο βασισμένο στο BERT, το οποίο δημιουργεί τα embeddings με τέτοιο τρόπο που λαμβάνει υπόψη τα συμφραζόμενα των λέξεων. Συνεπώς, αν τα αφαιρούσα, θα άλλαζα ουσιαστικά το νόημα των προτάσεων, π.χ στα Ελληνικά η λέξη “μη” / “μην” δημιουργεί άρνηση και αφαιρώντας την, η πρόταση αποκτά διαφορετικό νόημα.

3.3 Περιγραφή Εργαλείων και βιβλιοθηκών

Η υλοποίηση έγινε με τη χρήση της γλώσσας Python, με τη δημιουργία jupyter notebook, στο περιβάλλον Google Collaboratory το οποίο συχνά αναφέρεται ως Colab. Αναλυτικότερα τα εργαλεία είναι τα εξής:

3.3.1 Google Colaboratory (Colab)

Είναι μια υπηρεσία cloud που παρέχεται από την Google Research. Ουσιαστικά είναι ένα γραφικό περιβάλλον (IDE) που επιτρέπει στον χρήστη να γράφει τον κώδικα σε Jupyter Notebook και να τον εκτελεί μέσα από το πρόγραμμα περιήγησης (browser). Συγκεκριμένα, υποστηρίζει τη γλώσσα προγραμματισμού Python, και συνήθως χρησιμοποιείται σε εργασίες μηχανικής μάθησης, ανάλυσης δεδομένων, κ.λπ. Αυτή η υπηρεσία φιλοξενείται δωρεάν με τον λογαριασμό GMail, και δεν απαιτεί κάποια εγκατάσταση στον υπολογιστή μας. Δίνει τη δυνατότητα εκτέλεσης του κώδικα σε GPU/TPU της Google, που είναι απαραίτητοι για εργασίες εκπαίδευσης μοντέλων μηχανικής μάθησης και ιδιαίτερα βαθιάς μάθησης (deep learning). Η δωρεάν έκδοση δίνει περιορισμένους πόρους, αλλά αρκετούς ώστε να δοκιμάσει κάποιος π.χ φοιτητής, να εκπαιδεύσει κάποια μικρότερα μοντέλα.

(όπως HuggingFace, Google Colab, βασικές βιβλιοθήκες python όπως η Tensorflow, κλπ)

3.3.2 Hugging Face

Το Hugging Face είναι μια πλατφόρμα για εφαρμογές επιστήμης των δεδομένων (data science) και τεχνητής νοημοσύνης (machine learning - ML). Παρέχει χώρο στο νέφος, όπου οι χρήστες μπορούν να αναζητήσουν προεκπαιδευμένα μοντέλα τεχνητής νοημοσύνης αλλά και πλήθος συνόλων δεδομένων (data sets) τα οποία έχουν ανεβάσει άλλοι χρήστες. Το Hugging Face συχνά αποκαλείται το GitHub της τεχνητής νοημοσύνης, καθώς επιτρέπει στους ειδικούς να μοιράζονται ελεύθερα τη δουλειά τους, αλλά και να δοκιμάζουν ότι έχουν ανεβάσει άλλοι χρήστες. Πέρα από το χώρο αποθήκευσης των μοντέλων και των συνόλων δεδομένων, το Hugging Face παρέχει και τη βιβλιοθήκη “Transformers” της Python. Η συγκεκριμένη βιβλιοθήκη απλοποιεί τη διαδικασία λήψης και εκπαίδευσης μοντέλων τεχνητής νοημοσύνης. Η βιβλιοθήκη παρέχει στους ειδικούς τα εργαλεία που τους βοηθούν να συμπεριλάβουν προϋπάρχοντα μοντέλα από το Hugging Face Hub καθώς και να δημιουργήσουν ροή εργασίας από την προεπεξεργασία των δεδομένων, μέχρι την εκπαίδευση και τη χρήση τους σε τελικές εφαρμογές. Αυτή τη στιγμή το Hugging Face φιλοξενεί πάνω από 350.000 μοντέλα, 75.000 data sets και 150.00 demo εφαρμογές (spaces). Ο κύριος προσανατολισμός του Hugging Face είναι η επεξεργασία φυσικής γλώσσας (NLP).

3.3.3 Hugging Face Transformers

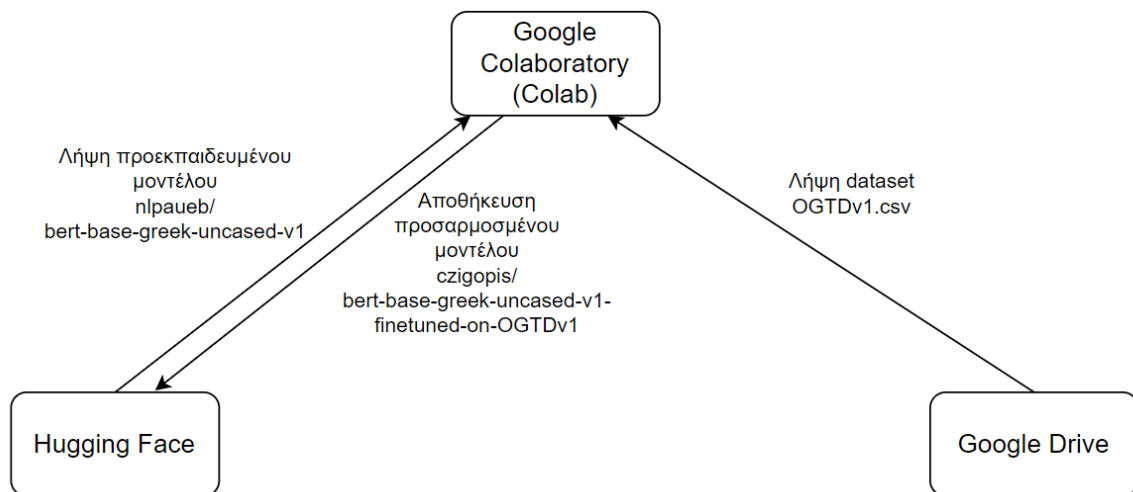
Η Hugging Face Transformers είναι μια βιβλιοθήκη που παρέχει εργαλεία και διεπαφές για το κατέβασμα προεκπαιδευμένων μοντέλων κορυφαίας απόδοσης καθώς και την προσαρμογή τους σε συγκεκριμένες εργασίες. Μπορούν να χρησιμοποιηθούν σε εφαρμογές όπως η επεξεργασία φυσικής γλώσσας, η μηχανική όραση, ομιλία κλπ. Η βιβλιοθήκη είναι φτιαγμένη με τέτοιο τρόπο ώστε κάθε μοντέλο μπορεί να χρησιμοποιηθεί είτε με τη βιβλιοθήκη Pytorch ή την Tensorflow, ανάλογα με της γνώσης του χρήστη.

3.3.4 Άλλα εργαλεία

Για την αποθήκευση του dataset χρησιμοποίησα το Google drive, το οποίο ανακτώ στο Colab με χρήση της βιβλιοθήκης google.colab. Το μοντέλο που χρησιμοποιώ ως βάση και στη συνέχεια προσαρμόζω και αποθηκεύω πίσω στο Hugging Face Hub, το κάνω με τις βιβλιοθήκες huggingface_hub και transformers.keras_callbacks.

3.3.5 Συνδυασμένη χρήση εργαλείων

Στο σχήμα φαίνεται ο συνδυασμός των εργαλείων που χρησιμοποίησα



Εικόνα 3.2 - Αρχιτεκτονική υλοποίησης

3.4 Αναλυτική περιγραφή αρχιτεκτονικής βαθιάς μάθησης

Όπως αναφέρθηκε και προηγουμένως το πρώτο βήμα της διαδικασίας είναι η φόρτωση και η προεπεξεργασία του σώματος κειμένων. Αφού συνδεθεί η μηχανή που τρέχει τον κώδικα με το Google drive, χρησιμοποιώντας τη βιβλιοθήκη `google.colab`, τα δεδομένα φορτώνονται με τη βιβλιοθήκη `Pandas` σε ένα `Pandas Dataframe`. Εδώ βλέπουμε ένα τυχαίο δείγμα των δεδομένων

idx		Tweet	Label
2251	2251	η κατερινα θα τα βαλει με την ελενα για τον μπιλ...επειδη ενοχληθηκε για τον χορο αλλα στην ουσια δεν ενοχληθηκε και η χριστινα θα ταβαλει με τη ελενα για τον παυλο επειδη ειπε οφ καμ οτι μπηκε για αυτον.. ποια εχει δικιο τωρα ρε φιλεε #poweroflovegr	Not Offensive
1564	1564	Από εδώ και στο εξής τα παιδιά που θα γεννιούνται στο σόι των Μητσοτακηδων θα παίρνουν ονόματα υπουργείων. #ΝΔ_ξεφτιλες	Not Offensive
2946	2946	Μπράβο στο υγιές κομμάτι της κοινωνίας που τους έκραξε! Αν δεν γινόταν αυτό θα λουζόμασταν νικήτρια την κάμπια! Εύγε! #masterchefgr_Xeftiles	Not Offensive
1237	1237	@USER Αμ δεν το ραβει κιολας..... κοντραριζεται στα ισα τη "φτου κακα"! ...αλλα οχι αγορι μου! Σαν "την οξω απο δω", ΚΑΝΕΙΣ! παλι δευτερος εισαι Τασο	Offensive
1099	1099	#ερχεται #Ευρωεκλογές2019 #TheNorthRemembers #syriza_xeftiles #ερχεται_ΠΟΥΤΣΑ και γι αυτούς που δεν είναι μαζί μας οταν οι άλλοι έπαιζαν θέατρο ξέροντας για νεκρούς!	Offensive

Εικόνα 3.3 - Απόσπασμα Ελληνικών tweets που φορτώθηκε στο Colab

Αμέσως μετά γίνεται η μετατροπή σε πεζά, η αφαίρεση των τόνων και των διακριτικών, καθώς και της φράσης `@User` με την οποία ο δημιουργός του dataset είχε αντικαταστήσει όλες τις αναφορές (mentions). Επίσης έχω αντικαταστήσει τις επαναλαμβανόμενες εμφανίσεις χαρακτήρων με ένα, π.χ σε πολλά σημεία υπήρχε “...” και το έκανα “.” Βλέπουμε πως γίνεται το dataset μετά την επεξεργασία

	idx	Tweet	Label	sentence	label
1956	1956	17 χρόνια φυγάς; 17χρονια μέσα έπρεπε να τον ξαναέβαζαν! Οχι μπήκε βγήκε! Να χαρώ εγώ δικαιοσύνη! #mesanyxta	Not Offensive	17 χρονια φυγας; 17χρονια μεσα επρεπε να τον ξαναεβαζαν! οχι μπηκε βγηκε! να χαρω εγω δικαιοσυνη! #mesanyxta	0
703	703	Ο Παυλος εχει φυγει απο το παιχνιδι με αρνητικες του κοσμου και τωρα μας δειχνουν τους φιλους τους Ετσι προωθηστε τον αγροφυλακα μαλακες #roweroflovegr	Offensive	ο παυλος εχει φυγει απο το παιχνιδι με αρνητικες του κοσμου και τωρα μας δειχνουν τους φιλους τους ετσι προωθηστε τον αγροφυλακα μαλακες #roweroflovegr	1
264	264	Αν είστε εσείς ηλίθιες δε φταίνει οι άντρες. Αν ασχολείστε με μαλάκες αντί να προσπερνάτε σίγουρα δε φταίνει οι μαλάκες.	Offensive	αν ειστε εσεις ηλιθιες δε φταινε οι αντρες. αν ασχολειστε με μαλακες αντι να προσπερνате σιγουρα δε φταινε οι μαλακες.	1
974	974	@USER Λαθος επικινδυνο ! Ο κυριος στοχος του Μητσοτακη ειναι ο κωλος σου! Βαζελινωσου συντροφε!	Offensive	λαθος επικινδυνο ! ο κυριος στοχος του μητσοτακη ειναι ο κωλος σου!βαζελινωσου συντροφε!	1
125	125	@USER Μα τη μπαναια δε ξερω αν ΕΣΥ εισαι ΤΟΣΟ μαλκισμενη η αν θεωρεις τους υπολοιπους μαλακες.ομολογω με ιντριγκάρει περισσότερο το πρώτο.	Offensive	μα τη μπαναια δε ξερω αν εσυ εισαι τοσο μαλκισμενη η αν θεωρεις τους υπολοιπους μαλακες.ομολογω με ιντριγκarei περισσοτερο το πρωτο.	1

Εικόνα 3.4 - Το dataset μετά την προεπεξεργασία

Το επεξεργασμένο κείμενο βρίσκεται στη στήλη “sentence” και έχω δημιουργήσει και τη στήλη “label” η οποία περιέχει την αριθμητική αντιστοίχιση του χαρακτηρισμού “Offensive” / “Not Offensive”, καθώς οι βιβλιοθήκες αναμένουν αριθμό ως label. Τέλος αφαιρώ τις περιττές στήλες και τα δεδομένα που απομένουν έχουν την παρακάτω μορφή

	idx	sentence	label
163	163	πολες γυναικες εχουν υποστει επιθεση απο αντιζηλος για τα ματια καποιου ανδρα! δεν τις καναμε ομως ευρωβουλευτες!	1
410	410	καλα ρε ξεφτιλες εκει στον #συριζα_ξεφτιλες το νεο ποινικο κωδικα τον σκεφτοσασταν πολυ καιρο; τι μαλακιες ειναι αυτες που κανετε ρε;	1
2635	2635	σου ευχομαι να εχεις μια γλυκια και ομορφη βραδια.! μου δινουν χαρα οι επιτυχιες σου και χθες με το σομερ.ειχες την απηχηση και την αγαπη του κοσμου και αυτο φαινεται.το ποσο καλη παραγωγος εισαι.	0
3175	3175	τωρα εισαι προκλητικος	0
2091	2091	κουκλα εισαι	0

Εικόνα 3.5 - Το dataset μετά την αφαίρεση των περιττών στηλών

Στη συνέχεια χωρίζω τα δεδομένα σε τρεις κατηγορίες:

Training - 70% των δεδομένων. Αυτά χρησιμοποιούνται κατά την προσαρμογή του μοντέλου ώστε να υπολογιστούν οι παράμετροι του νευρωνικού δικτύου.

Validation - 20% των δεδομένων. Αυτά παρέχονται στο μοντέλο κατά την προσαρμογή, ώστε να καθοδηγείται το μοντέλο προς την βέλτιστη λύση.

Test - 10%. Αυτό χρησιμοποιείται μετά την εκπαίδευση ώστε να ελεγχθεί η αποδοτικότητα του τελικού μοντέλου με δεδομένα που δεν έχει ξαναδεί.

Τέλος τα δεδομένα μεταφέρονται σε ένα HuggingFace αντικείμενο, τύπου DatasetDict. Ο λόγος είναι ότι στην επόμενη φάση, που θα γίνει το tokenization, η διαδικασία αυτή μπορεί να γίνει ταυτόχρονα σε ομάδες προτάσεων (batches) και να είναι πολύ πιο γρήγορη.

3.4.1 Ανάλυση σε σύμβολα

Στην επόμενη φάση πρέπει τα tweets να αναλυθούν σε σύμβολα (tokenization). Αφού χρησιμοποιώ μοντέλο που βασίζεται στο BERT, δε θα χρησιμοποιήσω κάποια από τα απλούστερα Bag of words, TfIdf, αλλά βασίζομαι στον tokenizer που παρέχεται από το βασικό μοντέλο. Στην ουσία είναι ένας WordPiece αναλυτής, όπως έχει περιγραφεί παραπάνω. Ας δούμε μερικά παραδείγματα ανάλυσης tweets. Αρχίζουμε με ένα υπόδειγμα που ταξινομήθηκε ως προσβλητικό

```
tokens = tokenizer.tokenize("δεν είναι καν ψηφος διαμαρτυριας. εβλεπαν  
καναλια, εφημεριδες, παπαδες, τεσσερα χρονια να λενε αλεξης ο ψευτης, ο  
μπηξε, ο δειξε, δεν εβλεπαν, δεν ακουγαν συριζα γυρω τους, σου λενε  
τοσο μαλακες που ειναι να μην απαντανε, κατι θα ισχυει απ' ολα αυτα. ")  
  
print(tokens)  
  
print(len(tokens))
```

Τα αποτελέσματα είναι τα ακόλουθα

```
['δεν', 'ειναι', 'καν', 'ψηφο', '##ς', 'διαμαρτυριας', '.', 'εβλεπαν',  
'καναλια', ',', 'εφημεριδες', ',', 'παπαδε', '##ς', ',', 'τεσσερα',  
'χρονια', 'να', 'λενε', 'αλεξης', 'ο', 'ψευτ', '##ης', ',', 'ο', 'μπ',  
'##ηξ', '##ε', ',', 'ο', 'δει', '##ξε', ',', 'δεν', 'εβλεπαν', ',',  
'δεν', 'ακουγα', '##ν', 'συριζα', 'γυρω', 'τους', ',', 'σου', 'λενε',  
'τοσο', 'μαλακες', 'που', 'ειναι', 'να', 'μην', 'απαντα', '##νε', ',',  
'κατι', 'θα', 'ισχυει', 'απ', '"', 'ολα', 'αυτα', '.']  
  
62
```

Εικόνα 3.6 - Παράδειγμα tokenization Ελληνικής πρότασης

Βλέπουμε πως αντιμετωπίζει ο tokenizer τις λέξεις που δεν αναγνωρίζει π.χ η “ψηφος” έγινε 'ψηφο', '##ς'. Στη συνέχεια ας δούμε πως γίνεται η ανάλυση σε σύμβολα ενός tweet που χαρακτηρίστηκε μη προσβλητικό

```
tokens = tokenizer.tokenize("ε αφου η λεγαμενη καταφερε να μειωσει τα
νουμερα του #masterchefgr, την εχω ικανη τελικα για ολα
.#masterchefgr_xeftiles")

print(tokens)

print(len(tokens))
```

που δίνει ως έξοδο

```
['ε', 'αφου', 'η', 'λεγαμε', '##ν', '##η', 'καταφερε', 'να', 'μειωσει',
'τα', 'νουμερα', 'του', '#', 'master', '##che', '##f', '##gr', ',',
'την', 'εχω', 'ικανη', 'τελικα', 'για', 'ολα', '.', '#', 'master',
'##che', '##f', '##gr', '_', 'x', '##ef', '##til', '##es']

35
```

Εικόνα 3.7 - Παράδειγμα tokenization Ελληνικής πρότασης που περιέχει hashtags

Εδώ μπορούμε να δούμε ότι ο tokenizer συνάντησε σημαντική δυσκολία στα hashtags όπως το #masterchefgr_xeftiles και το μετέτρεψε σε εννέα σύμβολα. Να υπενθυμίσω ότι έγιναν και δοκιμές όπου είχαν αφαιρεθεί τα hashtags

Στη συνέχεια ο tokenizer μετατρέπει τα σύμβολα σε αριθμούς, με βάση το λεξικό του. Για το προηγούμενο παράδειγμα, ζητώντας τους κωδικούς (ids) των συμβόλων

```
tokens = tokenizer.tokenize("ε αφου η λεγαμενη καταφερε να μειωσει τα
νουμερα του #masterchefgr, την εχω ικανη τελικα για ολα
.#masterchefgr_xeftiles ")

ids = tokenizer.convert_tokens_to_ids(tokens)

print(ids)

print(len(ids))
```

παίρνουμε την ακόλουθη λίστα

```
[237, 496, 239, 9300, 273, 267, 1726, 348, 7481, 355, 10831, 346, 110,
6744, 5514, 187, 4206, 119, 350, 576, 12084, 731, 353, 419, 121, 110,
6744, 5514, 187, 4206, 154, 179, 7068, 15555, 1065]

35
```

Εικόνα 3.8 - Παράδειγμα μετατροπής πρότασης σε λίστα από token ids

η οποία περιέχει και πάλι 35 τιμές που αντιστοιχούν στα σύμβολα του αρχικού tweet. Στην πραγματικότητα ο tokenizer προσθέτει επιπλέον σύμβολα που βοηθούν στον εντοπισμό της αρχής, του τέλους του κειμένου καθώς και του διαχωρισμού των προτάσεων. Συνεχίζοντας με το παραπάνω παράδειγμα, με την κατάλληλη χρήση του tokenizer

```
tokenized = tokenizer("ε αφου η λεγαμενη καταφερε να μειωσει τα νουμερα  
του #masterchefgr, την εχω ικανη τελικα για ολα  
.#masterchefgr_xeftiles")  
  
print (tokenized)  
  
print(len(tokenized['input_ids']))
```

βλέπουμε το ακόλουθο αποτέλεσμα

```
{'input_ids': [101, 237, 496, 239, 9300, 273, 267, 1726, 348, 7481,  
355, 10831, 346, 110, 6744, 5514, 187, 4206, 119, 350, 576, 12084, 731,  
353, 419, 121, 110, 6744, 5514, 187, 4206, 154, 179, 7068, 15555, 1065,  
102], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

37

Εικόνα 3.9 - Δημιουργία token ids, token type ids και attention mask, μιας πρότασης

Εδώ παρατηρούμε ότι οι κωδικοί των συμβόλων βρίσκονται στο πεδίο με κλειδί 'input_ids' αλλά το πλήθος τους είναι τώρα 37, δύο περισσότερα από πριν. Τα δύο νέα σύμβολα που έχουν προστεθεί έχουν κωδικούς 101, στην αρχή και 102, στο τέλος. Αυτοί είναι οι κωδικοί που αντιστοιχούν στα σύμβολα [CLS] και [SEP] και είναι οι ίδιοι σε όλα τα μοντέλα που βασίζονται στο BERT. Αξίζει να σημειωθεί ότι το μοντέλο που προσαρμόστηκε έχει μέγεθος λεξικού 35.000 σύμβολα

```
print(tokenizer.vocab_size)
```

35000

3.4.2 Εξαγωγή χαρακτηριστικών

Τα μοντέλα που βασίζονται στο BERT παράγουν ως έξοδο, διανύσματα πολλών διαστάσεων. με το βασικό μοντέλο να παράγει έξοδο διάστασης 768, για κάθε σύμβολο. Πιο

```
tweets = [
    "οχι λαθρομεταναστες, ειναι παρατυποι,οχι πουστηδες, ειναι gay, οχι  

    πασοκ, ειναι κιναλ,οχι εγκληματιες, ειναι ρουβικωνας,οχι γυφτοι, ειναι  

    ρομα, οχι απλυτοι, ειναι αναρχικοι,οχι αστυνομικοι, ειναι μπατσοι,οχι  

    εθνος ειναι κρατος.αν εισαι αριστερος τα ξερεις.#εκλογες_2019 #εκλογες",
    "τι ηλθια εισαι κοπελα μου; πραγματικα απορω."]

inputs = tokenizer(tweets, padding=True, truncation=True,
return_tensors="tf")

print(inputs)
```

[illegible]

Ζηγγόπης Χρήστος, Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής

Παρατηρούμε ότι πραγματοποιήθηκε γέμισμα (padding) στη δεύτερη πρόταση και το διάνυσμα (tensor) έχει διάσταση (2, 72), 2 γραμμές (μία για κάθε tweet) και 72 στήλες για τα σύμβολα της πρώτης πρότασης μαζί με τα CLS, SEP. Η συμπλήρωση στο δεύτερο tweet έγινε με τη χρήση του αναγνωριστικού (ID) 0 και ο διαχωρισμός των πραγματικών συμβόλων από τα γεμίσματα γίνεται στο πεδίο “attention_mask”, όπου μπορούμε να δούμε ότι στα πραγματικά σύμβολα αντιστοιχεί το 1 ενώ στα γεμίσματα, το 0. Αν τώρα ζητήσουμε από το μοντέλο να μας δώσει τα χαρακτηριστικά (hidden states) βλέπουμε τα εξής:

```
outputs = model(**inputs, output_hidden_states=True)
print(outputs.hidden_states)

(<tf.Tensor: shape=(2, 72, 768), dtype=float32, numpy=
array([[[[-0.3199019 ,  0.63365436,  0.3718311 , ...,  0.96305233,
          0.01841231,  0.6303837 ],
        [-0.5276943 , -1.4389311 ,  1.6404665 , ...,  1.2881508 ,
          -0.02972833,  0.57718843],
        [-0.42225495, -0.98332155,  0.39861748, ...,  1.5053941 ,
          -0.14159904, -0.5001013 ],
        ...,
        [-0.34375498, -0.4397482 , -0.41877627, ..., -0.19632004,
          -0.3885733 , -0.09417821],
        [ 1.1559227 , -0.12907135,  0.3640595 , ...,  0.05931137,
          -0.26151833,  1.1487795 ],
        [ 0.07606114,  0.5322272 , -0.26239884, ..., -0.02320243,
          -0.26022935, -0.68814975]]],
        [[[-0.3199019 ,  0.63365436,  0.3718311 , ...,  0.96305233,
          0.01841231,  0.6303837 ],
        [ 0.39607096,  0.31497663, -0.68153405, ...,  0.19850783,
          0.2360254 ,  0.34423536],
        [ 0.09274441, -0.827139 ,  0.46993172, ...,  0.6764323 ,
          -0.4472318 ,  0.04474012],
        ...,
        [ 0.37700066,  0.6330965 , -0.07519493, ..., -0.75705016,
          -0.48363525, -0.32543513],
        [ 0.24023575,  0.76669395,  0.37087804, ..., -0.86710393,
          -0.29588822, -0.1593717 ],
        [ 0.40694892,  0.89926696,  0.04090584, ..., -0.622632 ,
          -0.2871477 ,  0.14308244]]], dtype=float32)>, ...,
```

Εικόνα 3.11 - Embeddings που παράγει το BERT για τη λίστα προτάσεων

Τα χαρακτηριστικά που παράγει το μοντέλο έχουν διάσταση (2, 72, 768). Όπως είπαμε και παραπάνω η πρώτη διάσταση δείχνει το πλήθος των κειμένων, η δεύτερη το μέγιστο πλήθος συμβόλων μετά το γέμισμα (padding) και την προσθήκη των ειδικών συμβόλων (CLS, SEP) και η τρίτη είναι το πλήθος των χαρακτηριστικών (features) που είναι βασικό στοιχείο κάθε BERT μοντέλου. Επειδή το δικό μου είναι προσαρμογή του μοντέλου

“nlpaueb/bert-base-greek-uncased-v1” [29] το οποίο με τη σειρά του είναι προσαρμογή του “bert-base-uncased”, γι αυτό τα χαρακτηριστικά (hidden states) είναι 768.

3.4.3 Προσαρμογή (fine tuning) του μοντέλου

Έχοντας υλοποιήσει το μηχανισμό που εξάγει τα χαρακτηριστικά, προχώρησα στην προσαρμογή του μοντέλου με βάση το dataset των Ελληνικών tweet. Για το σκοπό αυτό χρησιμοποίησα την κλάση *TFAutoModelForSequenceClassification*, από τη βιβλιοθήκη Hugging Face. Η κλάση αυτή παρέχει ένα φιλικό τρόπο για τη δημιουργία νευρωνικού δικτύου που δέχεται ως είσοδο τα χαρακτηριστικά (hidden states) ενός μοντέλου BERT, το οποίο στη συνέχεια μπορεί να εκπαιδευτεί / προσαρμοστεί με τα δεδομένα που θα του δοθούν. Η αρχικοποίηση της κλάσης είναι πολύ απλή όπως φαίνεται και στον κώδικα

```
from transformers import TFAutoModelForSequenceClassification
import tensorflow as tf
model = TFAutoModelForSequenceClassification.from_pretrained(
    'nlpaueb/bert-base-greek-uncased-v1',
    num_labels = 2,
    id2label = {0: "Not Offensive", 1: "Offensive"},
    label2id = {"Not Offensive": 0, "Offensive": 1}
)
```

Εικόνα 3.12 - Αρχικοποίηση του μοντέλου με βάση το GreekBERT

Εδώ η μέθοδος “from_pretrained” θα κατεβάσει (download) το μοντέλο που παρέχεται ως παράμετρος, από το Hugging Face Hub. Μπορούμε τώρα να δούμε τη μορφή του μοντέλου:

```
model.summary()

Model: "tf_bert_for_sequence_classification"

```

Layer (type)	Output Shape	Param #
[0]bert (TFBertMainLayer)	multiple	112.921.344
[1]dropout_37 (Dropout)	multiple	0
[2]classifier (Dense)	multiple	1.538

```

Total params: 112.922.882 (430.77 MB)
Trainable params: 112.922.882 (430.77 MB)
Non-trainable params: 0 (0.00 Byte)

```

Εικόνα 3.13 - Αρχιτεκτονική του μοντέλου για προσαρμογή (fine tuning)

Το τελικό μοντέλο όπως φαίνεται παραπάνω χρησιμοποιεί ένα προεκπαιδευμένο μοντέλο BERT για την εξαγωγή χαρακτηριστικών και έχει προστεθεί ένα απλό επίπεδο ταξινόμησης για την εργασία που θέλουμε να πραγματοποιήσουμε. Έτσι χρησιμοποιούμε την υπάρχουσα γνώση (transfer learning) και την προσαρμόζουμε για ταξινόμηση κειμένου. Όπως μπορούμε να διαπιστώσουμε, το αρχικό μοντέλο BERT περιέχει 112M (εκατομμύρια παραμέτρους), ένα ενδιάμεσο Dropout, για αποφυγή υπερπροσαρμογής, και το τελικό, που θα πρέπει να εκπαιδευτεί, είναι ένα Dense Layer (πλήρως συνεκτικό επίπεδο/στρώμα) με 1.538 παραμέτρους. Εκτελώντας τον παραπάνω κώδικα βλέπουμε ως αποτέλεσμα το μήνυμα

```
tf_model.h5: 100% 564M/564M [00:42<00:00, 9.92MB/s]
All model checkpoint layers were used when initializing
TFBertForSequenceClassification.

Some layers of TFBertForSequenceClassification were not initialized
from the model checkpoint at nlpaueb/bert-base-greek-uncased-v1
and are newly initialized: ['classifier']

You should probably TRAIN this model on a down-stream task to be
able to use it for predictions and inference.
```

Εικόνα 3.14 - Προτροπή του HuggingFace για προσαρμογή του μοντέλου

το οποία μας ενημερώνει ότι το **συνολικό** μοντέλο πρέπει να εκπαιδευτεί/προσαρμοστεί ώστε να μπορέσει το νευρωνικό δίκτυο να χρησιμοποιηθεί για την ταξινόμηση κειμένου που θέλουμε να κάνουμε.

Αν τώρα εκτελέσουμε τον ακόλουθο κώδικα

```
classifier_layer = model.layers[2]
print(classifier_layer.get_config())
{'name': 'classifier', 'trainable': True, 'dtype': 'float32',
 'units': 2, 'activation': 'linear', 'use_bias': True,
 'kernel_initializer': {'module': 'keras.initializers',
 'class_name': 'TruncatedNormal', 'config': {'mean': 0.0, 'stddev':
 0.02, 'seed': None}, 'registered_name': None}, 'bias_initializer':
 {'module': 'keras.initializers', 'class_name': 'Zeros', 'config':
 {}, 'registered_name': None}, 'kernel_regularizer': None,
 'bias_regularizer': None, 'activity_regularizer': None,
 'kernel_constraint': None, 'bias_constraint': None}
```

Εικόνα 3.15 - Χαρακτηριστικά του classifier επιπέδου

καταλαβαίνουμε ότι το επίπεδο είναι εκπαιδεύσιμο ('trainable': True), δηλαδή τα βάρη και τα biases μπορούν να αλλάξουν, τα βάρη είναι πραγματικοί (float32), χρησιμοποιείται για δυαδική ταξινόμηση ('units': 2), ενώ η παράμετρος activation: 'linear' υποδεικνύει ότι δεν εφαρμόζεται κάποια συνάρτηση ενεργοποίησης στην έξοδο, παράγοντας μη επεξεργασμένα βάρη για κάθε κλάση. Τέλος τα πεδία με κατάληξη _regularizer: 'None' δείχνουν ότι δεν γίνεται καμία κανονικοποίηση στα βάρη ή τα biases.

Για το Dropout Layer έχουμε

```
dropout_layer = model.layers[1]
print(dropout_layer.get_config())

{'name': 'dropout_37', 'trainable': True, 'dtype': 'float32',
'rate': 0.1, 'noise_shape': None, 'seed': None}
```

Εικόνα 3.16 - Χαρακτηριστικά του dropout επιπέδου

όπου φαίνεται ότι από τα στοιχεία εισόδου απορρίπτεται τυχαία το 10%, για αποφυγή υπερπροσαρμογής.

Τέλος για να δούμε τη μορφή του προεκπαιδευμένου μοντέλου (GreekBERT [29]) εκτελούμε

```
bert_layer = model.layers[0]
print(bert_layer.get_config())
```

με έξοδο (αρκετές ιδιότητες αφαιρέθηκαν λόγω μεγέθους)

```
{'name': 'bert', 'trainable': True, 'dtype': 'float32', 'config':
{'return_dict': True, 'output_hidden_states': False,
'output_attentions': False, ..., 'is_encoder_decoder': False,
'is_decoder': False, 'cross_attention_hidden_size': None,
'add_cross_attention': False, 'tie_encoder_decoder': False,
'max_length': 20, 'min_length': 0, 'do_sample': False,
'diversity_penalty': 0.0, 'temperature': 1.0, 'top_k': 50, 'top_p':
1.0, 'typical_p': 1.0, 'repetition_penalty': 1.0, 'length_penalty':
1.0, 'no_repeat_ngram_size': 0, 'encoder_no_repeat_ngram_size': 0,
'bad_words_ids': None, 'num_return_sequences': 1,
'chunk_size_feed_forward': 0, 'output_scores': False,
'return_dict_in_generate': False, 'forced_bos_token_id': None,
'forced_eos_token_id': None, 'remove_invalid_values': False,
'exponential_decay_length_penalty': None, 'suppress_tokens': None,
'begin_suppress_tokens': None, 'architectures': ['BertForPreTraining'],
'finetuning_task': None, 'bos_token_id': None, 'pad_token_id': 0,
```

```
'eos_token_id': None, 'sep_token_id': None, 'decoder_start_token_id':
None, 'task_specific_params': None, 'problem_type': None,
'_name_or_path': 'nlpauueb/bert-base-greek-uncased-v1',
'transformers_version': '4.35.2', 'model_type': 'bert', 'output_past':
True, 'vocab_size': 35000, 'hidden_size': 768, 'num_hidden_layers': 12,
'num_attention_heads': 12, 'hidden_act': 'gelu', 'intermediate_size':
3072, 'hidden_dropout_prob': 0.1, 'attention_probs_dropout_prob': 0.1,
'max_position_embeddings': 512, 'type_vocab_size': 2,
'initializer_range': 0.02, 'layer_norm_eps': 1e-12,
'position_embedding_type': 'absolute', 'use_cache': True,
'classifier_dropout': None}}
```

Εικόνα 3.17 - Χαρακτηριστικά του BERT επιπέδου

Από το συνδυασμό των παραμέτρων *'architectures': ['BertForPreTraining']* και *'finetuning_task': None*, συμπεραίνουμε ότι το μοντέλο έχει μια γενική αντίληψη της γλώσσας (και μάλιστα της Ελληνικής) που απέκτησε κατά την προεκπαίδευση, αλλά δεν έχει προσαρμοστεί σε πιο εξειδικευμένη εργασία όπως η αναγνώριση εχθρικού λόγου, στην οποία θα εκπαιδευτεί στη συνέχεια.

```
matthews_correlation_metric = load("matthews_correlation")
f1_metric = load("f1")
accuracy_metric = load("accuracy")

def compute_metrics(eval_predictions):
    predictions, expected_labels = eval_predictions
    binary_predictions = np.argmax(predictions, axis=1)
    ...
    return mcm | f1 | accuracy
metric_callback = KerasMetricCallback(
    metric_fn=compute_metrics, eval_dataset=tf_validation_dataset
)
callbacks = [metric_callback]
initial_learning_rate = 5e-5
lr_scheduler = PolynomialDecay(
    initial_learning_rate = initial_learning_rate, end_learning_rate =
0.0, decay_steps = total_train_steps
)
optimizer_with_decay = Adam(learning_rate=lr_scheduler)
loss_function = SparseCategoricalCrossentropy(from_logits=True)

model.compile(optimizer=optimizer_with_decay, loss=loss_function)

history = model.fit(
    tf_train_dataset,
    validation_data=tf_validation_dataset,
    epochs=num_epochs,
    callbacks=callbacks)
```

Εικόνα 3.18 - Προσαρμογή - *fit* (fine tuning) του μοντέλου με τα Ελληνικά tweets

3.5 Παρουσίαση αποτελεσμάτων πειραμάτων για διάφορους συνδυασμούς υπερ παραμέτρων

Σε όλα τα ακόλουθα πειράματα χρησιμοποιήθηκαν οι εξής κοινές υπερ-παραμέτροι:

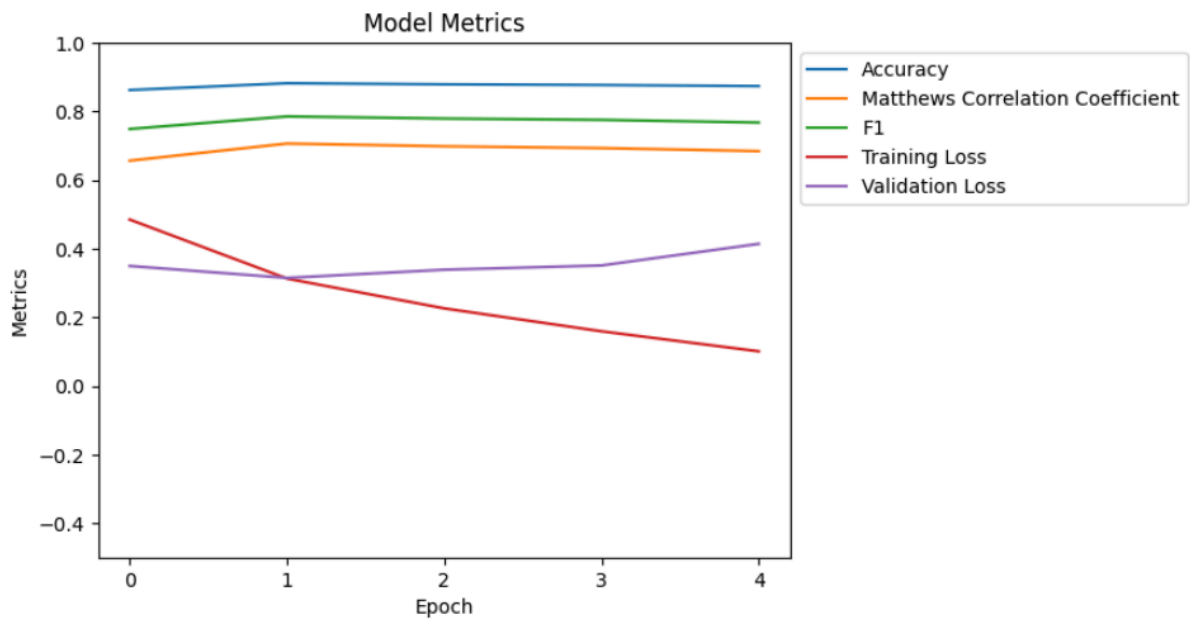
Loss function	SparseCategoricalCrossentropy(from_logits=True)
Epochs	5
Optimizer	Adam

3.5.1 Μη ισορροπημένο dataset, με σταθερό learning rate

Dataset	Imbalanced (Offensive: 29,3%, Not Offensive: 70,7%)
Hashtags	Included
Initial Learning Rate	1e-5
Learning Rate Decay	None

Το αποτέλεσμα ήταν το εξής

```
output
Epoch 1/5
209/209 [=====] - 123s 393ms/step - loss: 0.4849 -
val_loss: 0.3497 - matthews_correlation: 0.6566 - f1: 0.7490 - accuracy:
0.8626
Epoch 2/5
209/209 [=====] - 62s 294ms/step - loss: 0.3128 -
val_loss: 0.3148 - matthews_correlation: 0.7066 - f1: 0.7856 - accuracy:
0.8824
Epoch 3/5
209/209 [=====] - 62s 296ms/step - loss: 0.2260 -
val_loss: 0.3387 - matthews_correlation: 0.6986 - f1: 0.7795 - accuracy:
0.8793
Epoch 4/5
209/209 [=====] - 62s 298ms/step - loss: 0.1591 -
val_loss: 0.3511 - matthews_correlation: 0.6933 - f1: 0.7757 - accuracy:
0.8772
Epoch 5/5
209/209 [=====] - 67s 322ms/step - loss: 0.1008 -
val_loss: 0.4141 - matthews_correlation: 0.6844 - f1: 0.7678 - accuracy:
0.8741
```



Εικόνα 3.19 - Γράφημα των metrics για μη ισορροπημένο dataset, με σταθερό learning rate

Παρατηρούμε από τα αποτελέσματα ότι εμφανίζεται υπερπροσαρμογή (overfitting) μετά τη δεύτερη εποχή. Τα metrics στο test dataset ήταν τα ακόλουθα

```
epochs: 3
Accuracy: 0.8837209302325582,
F1: 0.7908745247148289,
Matthews_correlation: 0.7103589017145717

epochs: 5
Accuracy: 0.8604651162790697,
F1: 0.7272727272727272,
Matthews_correlation: 0.639321895567359
```

3.5.2 Ισορροπημένο dataset, με σταθερό learning rate

Dataset	Balanced
Hashtags	Included
Initial Learning Rate	1e-5
Learning Rate Decay	None

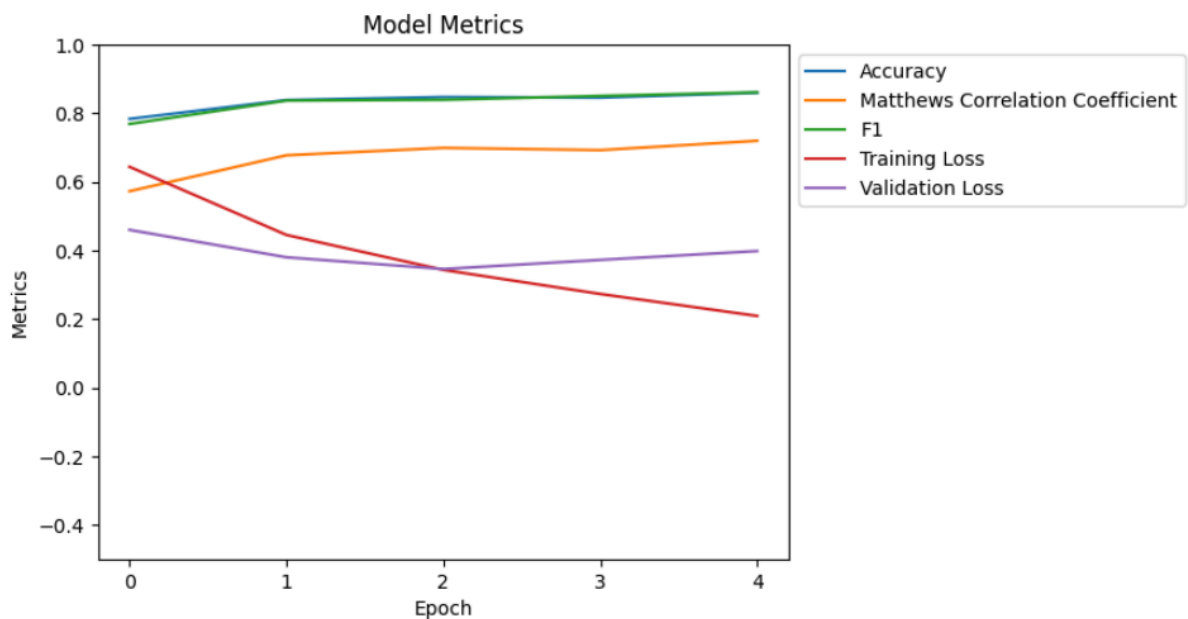
Το αποτέλεσμα ήταν το εξής

```
output
Epoch 1/5
```

```

122/122 [=====] - 106s 453ms/step - loss: 0.6439 -
val_loss: 0.4600 - matthews_correlation: 0.5728 - f1: 0.7689 - accuracy:
0.7837
Epoch 2/5
122/122 [=====] - 39s 316ms/step - loss: 0.4452 -
val_loss: 0.3801 - matthews_correlation: 0.6775 - f1: 0.8372 - accuracy:
0.8387
Epoch 3/5
122/122 [=====] - 39s 322ms/step - loss: 0.3433 -
val_loss: 0.3459 - matthews_correlation: 0.6991 - f1: 0.8396 - accuracy:
0.8475
Epoch 4/5
122/122 [=====] - 37s 305ms/step - loss: 0.2731 -
val_loss: 0.3724 - matthews_correlation: 0.6927 - f1: 0.8508 - accuracy:
0.8457
Epoch 5/5
122/122 [=====] - 37s 304ms/step - loss: 0.2090 -
val_loss: 0.3982 - matthews_correlation: 0.7199 - f1: 0.8612 - accuracy:
0.8599

```



Εικόνα 3.20 - Γράφημα των metrics για ισορροπημένο dataset, με σταθερό learning rate

Εδώ η υπερπροσαρμογή (overfitting) εμφανίζεται μετά την τρίτη εποχή. Τα metrics στο test dataset ήταν τα ακόλουθα

```

epochs: 3
Accuracy: 0.8710359408033826,
F1: 0.7312775330396475,
Matthews_correlation: 0.6645816448768699

```

epochs: 5

Accuracy: 0.8158844765342961,

F1: 0.8271186440677967,

Matthews Correlation: 0.6328017726582021

3.5.3 Μη ισορροπημένο dataset, με σταθερό **μεγάλο** learning rate

Dataset	Imbalanced (Offensive: 29,3%, Not Offensive: 70,7%)
Hashtags	Included
Initial Learning Rate	5e-5
Learning Rate Decay	None

Epoch 1/5

209/209 [=====] - 120s 378ms/step - loss: 0.3562 - val_loss: 0.3170 - matthews_correlation: 0.7264 - f1: 0.8083 - accuracy: 0.8845

Epoch 2/5

209/209 [=====] - 62s 297ms/step - loss: 0.2318 - val_loss: 0.4287 - matthews_correlation: 0.6604 - f1: 0.7648 - accuracy: 0.8554

Epoch 3/5

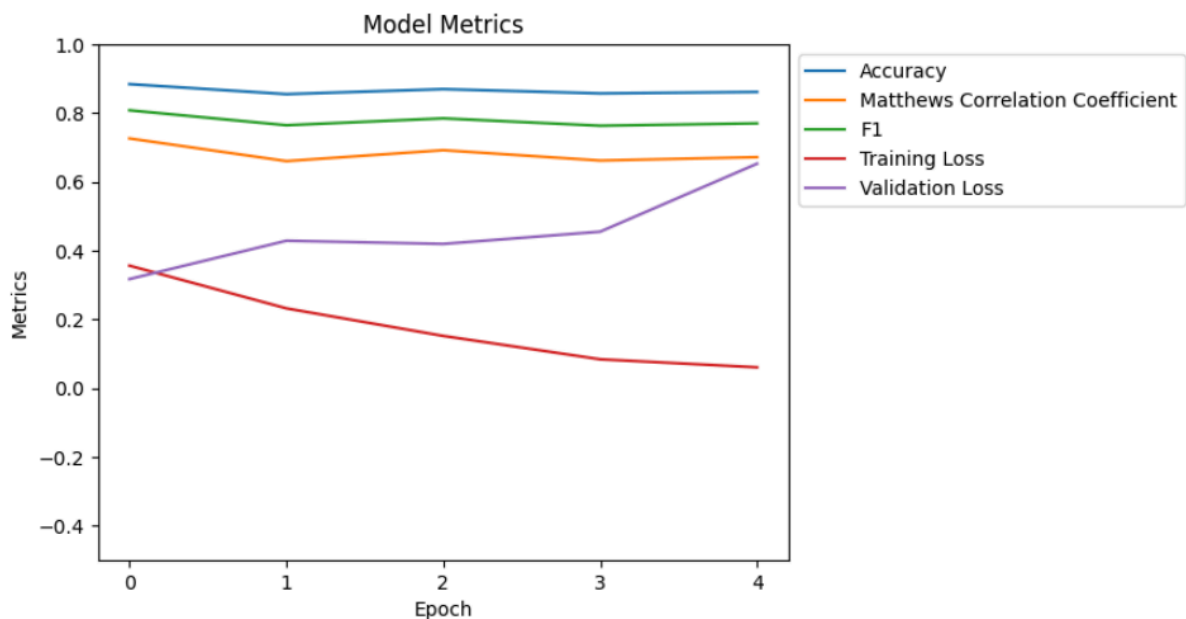
209/209 [=====] - 63s 301ms/step - loss: 0.1516 - val_loss: 0.4194 - matthews_correlation: 0.6922 - f1: 0.7849 - accuracy: 0.8699

Epoch 4/5

209/209 [=====] - 62s 298ms/step - loss: 0.0835 - val_loss: 0.4550 - matthews_correlation: 0.6621 - f1: 0.7634 - accuracy: 0.8574

Epoch 5/5

209/209 [=====] - 63s 300ms/step - loss: 0.0601 - val_loss: 0.6529 - matthews_correlation: 0.6720 - f1: 0.7703 - accuracy: 0.8616



Εικόνα 3.21 - Γράφημα των metrics για μη ισορροπημένο dataset, με σταθερό (μεγάλο) learning rate

epochs: 3

Accuracy: 0.8668076109936576,
F1: 0.7974276527331191,
Matthews_correlation: 0.7003646793252822

epochs: 5

Accuracy: 0.8202959830866807,
F1: 0.7536231884057971,
Matthews_correlation: 0.6322651459319248

3.5.4 Ισορροπημένο dataset, με σταθερό **μεγάλο** learning rate

Dataset	Balanced
Hashtags	Included
Initial Learning Rate	5e-5
Learning Rate Decay	None

Epoch 1/5

122/122 [=====] - 99s 469ms/step - loss: 0.5300 - val_loss: 0.3282 - matthews_correlation: 0.7249 - f1: 0.8577 - accuracy: 0.8617

Epoch 2/5

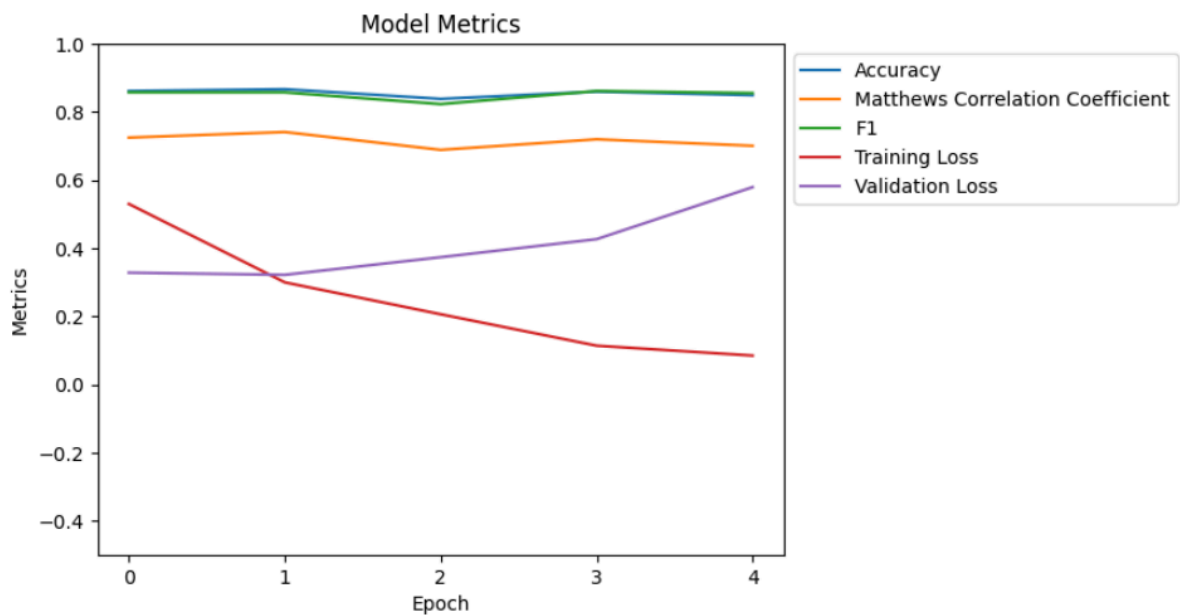
122/122 [=====] - 38s 314ms/step - loss: 0.2997 - val_loss: 0.3218 - matthews_correlation: 0.7413 - f1: 0.8577 - accuracy: 0.8670

Epoch 3/5

```

122/122 [=====] - 37s 304ms/step - loss: 0.2060 -
val_loss: 0.3736 - matthews_correlation: 0.6889 - f1: 0.8233 - accuracy:
0.8387
Epoch 4/5
122/122 [=====] - 37s 304ms/step - loss: 0.1137 -
val_loss: 0.4266 - matthews_correlation: 0.7200 - f1: 0.8616 - accuracy:
0.8599
Epoch 5/5
122/122 [=====] - 37s 302ms/step - loss: 0.0846 -
val_loss: 0.5793 - matthews_correlation: 0.7008 - f1: 0.8557 - accuracy:
0.8493

```



Εικόνα 3.22 - Γράφημα των metrics για ισορροπημένο dataset, με σταθερό (μεγάλο) learning rate

Παρατηρούμε ότι σε σχέση με το προηγούμενο ισορροπημένο dataset, η υπερπροσαρμογή (overfitting) εμφανίζεται γρηγορότερα - μετά τη δεύτερη εποχή. Τα metrics στο test dataset είναι τα ακόλουθα:

```

epochs: 3
Accuracy: 0.8773784355179705,
F1: 0.7819548872180451,
Matthews_correlation: 0.7068300073211548

epochs: 5
Accuracy: 0.8158844765342961,
F1: 0.8247422680412371,
Matthews_correlation: 0.6318727928352269

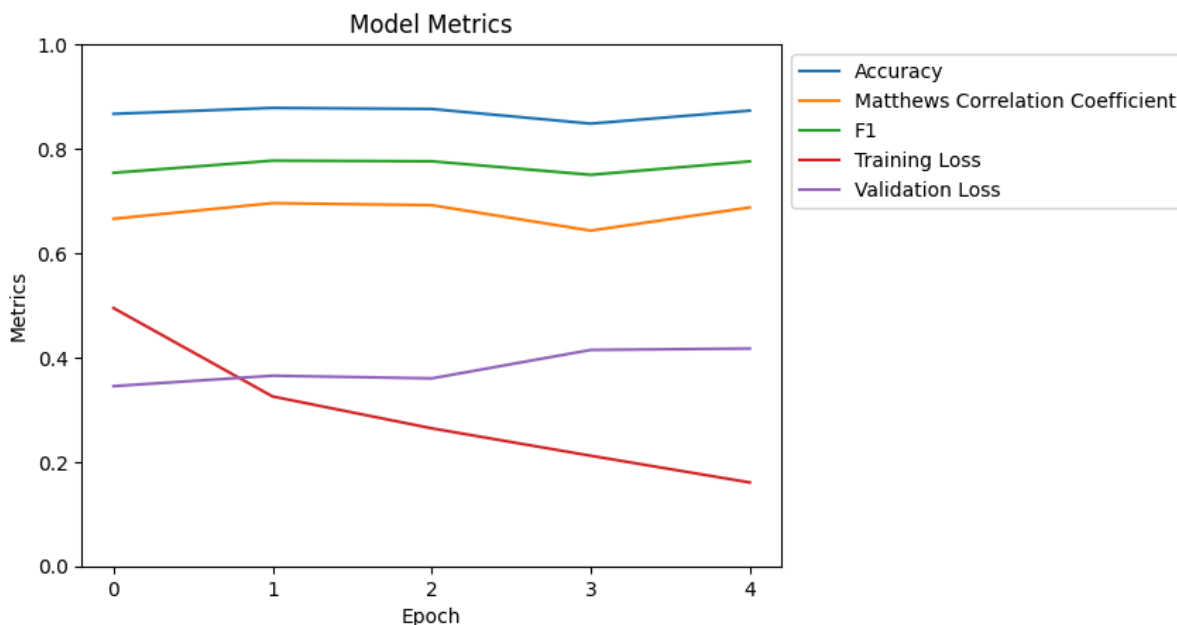
```

3.5.5 Μη ισορροπημένο dataset, με learning rate που μειώνεται γραμμικά

Dataset	Imbalanced (Offensive: 29,3%, Not Offensive: 70,7%)
Hashtags	Included
Initial Learning Rate	1e-5
Learning Rate Decay	PolynomialDecay (1e-5 -> 0.0)

Εδώ δοκιμάζω να μην έχω σταθερό learning rate αλλά να μειώνεται γραμμικά μέχρι το 0. Το αποτέλεσμα ήταν το εξής

```
output
Epoch 1/5
209/209 [=====] - 122s 379ms/step - loss: 0.4946 - val_loss: 0.3450 - matthews_correlation: 0.6658 - f1: 0.7538 - accuracy: 0.8668
Epoch 2/5
209/209 [=====] - 62s 296ms/step - loss: 0.3252 - val_loss: 0.3650 - matthews_correlation: 0.6958 - f1: 0.7771 - accuracy: 0.8783
Epoch 3/5
209/209 [=====] - 62s 296ms/step - loss: 0.2642 - val_loss: 0.3599 - matthews_correlation: 0.6918 - f1: 0.7759 - accuracy: 0.8762
Epoch 4/5
209/209 [=====] - 62s 296ms/step - loss: 0.2116 - val_loss: 0.4144 - matthews_correlation: 0.6430 - f1: 0.7500 - accuracy: 0.8481
Epoch 5/5
209/209 [=====] - 62s 295ms/step - loss: 0.1604 - val_loss: 0.4171 - matthews_correlation: 0.6875 - f1: 0.7757 - accuracy: 0.8730
```



Εικόνα 3.23 - Γράφημα των metrics για μη ισορροπημένο dataset, με γραμμικά μειούμενο learning rate

Τα αποτελέσματα παραμένουν συγκρίσιμα με το προηγούμενο, όπου είχα σταθερό learning rate

3.5.6 Ισορροπημένο dataset, με learning rate που μειώνεται γραμμικά

Dataset	Balanced
Hashtags	Included
Initial Learning Rate	1e-5
Learning Rate Decay	PolynomialDecay (1e-5 -> 0.0)

Με learning rate που μειώνεται γραμμικά μέχρι το 0. Το αποτέλεσμα ήταν το εξής

```

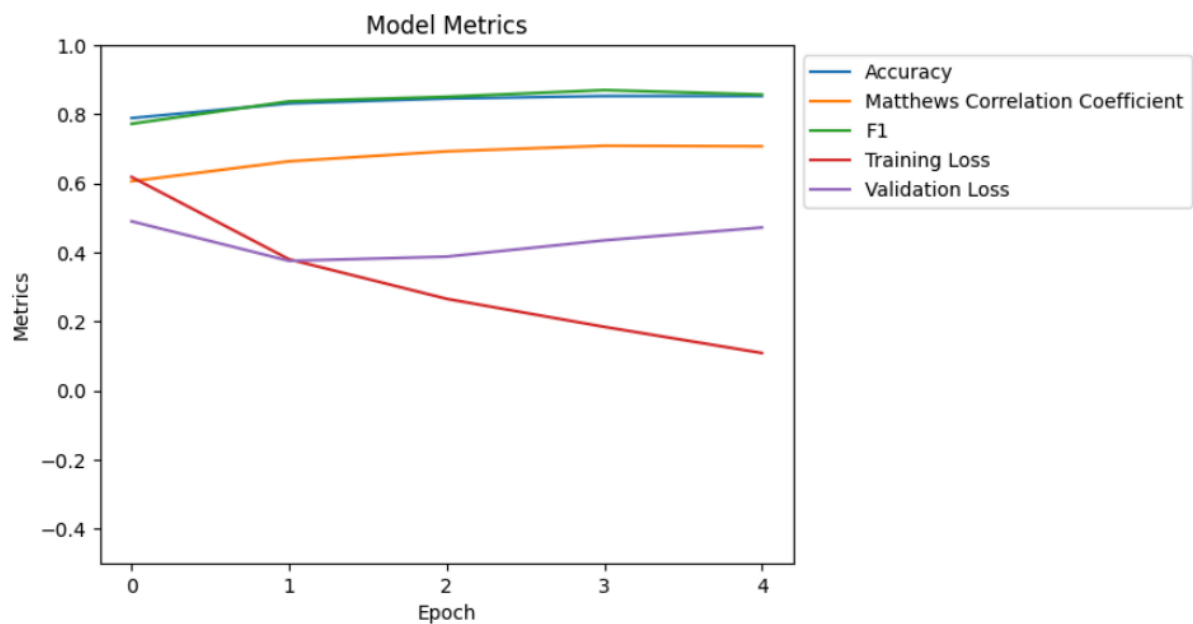
output
Epoch 1/5
122/122 [=====] - 106s 498ms/step - loss: 0.6183 -
val_loss: 0.4900 - matthews_correlation: 0.6060 - f1: 0.7725 - accuracy:
0.7890
Epoch 2/5
122/122 [=====] - 37s 299ms/step - loss: 0.3799 -
val_loss: 0.3754 - matthews_correlation: 0.6638 - f1: 0.8376 - accuracy:
0.8316
Epoch 3/5

```

```

122/122 [=====] - 36s 294ms/step - loss: 0.2654 -
val_loss: 0.3877 - matthews_correlation: 0.6927 - f1: 0.8508 - accuracy:
0.8457
Epoch 4/5
122/122 [=====] - 35s 289ms/step - loss: 0.1842 -
val_loss: 0.4348 - matthews_correlation: 0.7088 - f1: 0.8701 - accuracy:
0.8528
Epoch 5/5
122/122 [=====] - 35s 290ms/step - loss: 0.1084 -
val_loss: 0.4720 - matthews_correlation: 0.7073 - f1: 0.8571 - accuracy:
0.8528

```



Εικόνα 3.24 - Γράφημα των metrics για ισορροπημένο dataset, με γραμμικά μειούμενο learning rate

Τα αποτελέσματα για το test dataset είναι

```

epochs: 3
Accuracy: 0.8772563176895307,
F1: 0.881118881118881,
Matthews_correlation: 0.7565470046208181

epochs: 5
Accuracy: 0.855595667870036,
F1: 0.8591549295774648,
Matthews_correlation: 0.7142277058141389

```

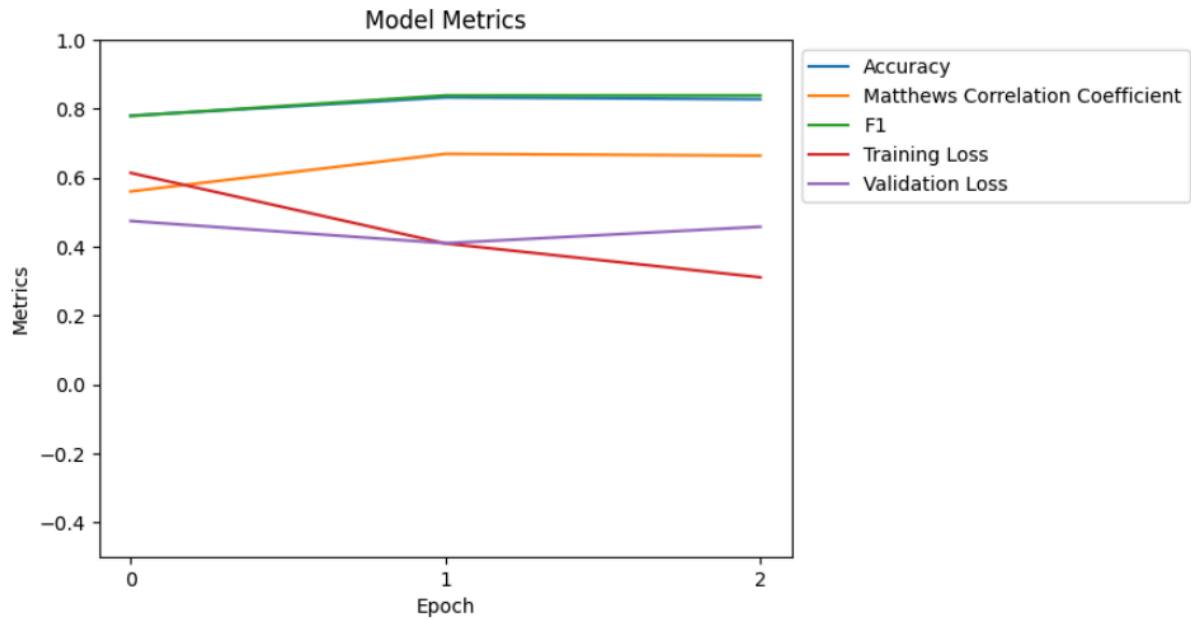
3.5.7 Ισορροπημένο dataset, **χωρίς** τα hashtags, με learning rate που μειώνεται γραμμικά

Με βάση τα προηγούμενα παρατηρούμε ότι υπάρχει υπερπροσαρμογή μετά την τρίτη εποχή, οπότε εδώ εφαρμόζω 3 εποχές

Epochs	3
Dataset	Balanced
Hashtags	NOT Included
Initial Learning Rate	1e-5
Learning Rate Decay	PolynomialDecay (1e-5 -> 0.0)

Το αποτέλεσμα ήταν το εξής

```
output
Epoch 1/3
122/122 [=====] - 101s 454ms/step - loss: 0.6139 -
val_loss: 0.4744 - matthews_correlation: 0.5603 - f1: 0.7794 - accuracy:
0.7801
Epoch 2/3
122/122 [=====] - 35s 288ms/step - loss: 0.4090 -
val_loss: 0.4098 - matthews_correlation: 0.6693 - f1: 0.8390 - accuracy:
0.8333
Epoch 3/3
122/122 [=====] - 35s 284ms/step - loss: 0.3108 -
val_loss: 0.4577 - matthews_correlation: 0.6641 - f1: 0.8391 - accuracy:
0.8280
```



Εικόνα 3.25 - Γράφημα των metrics για ισορροπημένο dataset, χωρίς hashtags, με γραμμικά μειούμενο learning rate

Τα αποτελέσματα για το test dataset είναι

```
epochs: 3
Accuracy: 0.7978339350180506,
F1: 0.8181818181818182,
Matthews_correlation: 0.5926614343051332

epochs: 5
Accuracy: 0.8014440433212996,
F1: 0.808362369337979,
Matthews_correlation: 0.6049065631888123
```

3.6 Σύγκριση αποτελεσμάτων

Σε όλα τα πειράματα χρησιμοποίησα ως loss function την SparseCategoricalCrossentropy της βιβλιοθήκης HuggingFace και optimizer τον Adam.

Στον ακόλουθο πίνακα συνοψίζω τα συγκριτικά αποτελέσματα

AA	Ισορροπημένο Dataset	Περιλαμβάνει Hashtags	Learning Rate Decay	Learning Rate	Εποχή βέλτιστων τιμών	Accuracy	F1	MCC*	Training Loss	Validation Loss
1	Όχι	Ναι	Όχι	1e-5	2	0.8824	0.7856	0.7066	0.3128	0.3148
2	Ναι	Ναι	Όχι	1e-5	5	0.8599	0.8612	0.7199	0.2090	0.3982

3	Όχι	Ναι	Όχι	5e-5	1	0.8845	0.8083	0.7264	0.3562	0.3170
4	Ναι	Ναι	Όχι	5e-5	2	0.8670	0.8577	0.7413	0.2997	0.3218
5	Όχι	Ναι	Polynomial Decay	1e-5 -> 0.0	2	0.8783	0.7771	0.6958	0.3252	0.3650
6	Ναι	Ναι	Polynomial Decay	1e-5 -> 0.0	4	0.8528	0.8701	0.7088	0.1842	0.4348
7	Ναι	Όχι	Polynomial Decay	1e-5 -> 0.0	2	0.8333	0.8390	0.6693	0.4090	0.4098

(*) MCC: Matthews Correlation Coefficient

Εικόνα 3.26 - Σύγκριση των metrics για τις διάφορες κατηγορίες πειραμάτων

Παρατηρώντας τις τιμές των metrics Accuracy, F1 και Matthews Correlation Coefficient, συμπεραίνουμε ότι σε γενικές γραμμές τις καλύτερες επιδόσεις έχουμε στις περιπτώσεις 2, 3, 4 και 6. Στις περιπτώσεις 3 και 4 όπου ξεκινάμε με σχετικά μεγάλο learning rate, παρατηρούμε ότι η υπερπροσαρμογή συμβαίνει πολύ γρήγορα (εποχή 1 ή 2). Αντίθετα ξεκινώντας με μικρό αρχικό learning rate (1e-5) η υπερπροσαρμογή εμφανίζεται λίγο αργότερα (περιπτώσεις 2 και 6). Επίσης η εισαγωγή learning rate decay δε φαίνεται να επηρεάζει τα αποτελέσματα της εκπαίδευσης. Ακόμη αν λάβουμε υπόψη τις καλύτερες περιπτώσεις (2, 3, 4, 6), η χρήση ενός ισορροπημένου dataset φαίνεται να δίνει ελαφρώς καλύτερα αποτελέσματα, καθώς τρεις από τις τέσσερις αυτές βασίζονται σε ισορροπημένο dataset. Τέλος η επιλογή που φαίνεται ξεκάθαρα να επηρεάζει τα αποτελέσματα της εκπαίδευσης είναι χρήση ή μη των hashtags. Όπως μπορούμε να δούμε στην περίπτωση 7, αφαιρώντας τα hashtags, έχουμε ξεκάθαρα χαμηλότερες τιμές Accuracy και Matthews Correlation Coefficient καθώς και πολύ υψηλές τιμές test και validation loss. Αυτό δεν μας προκαλεί έκπληξη καθώς το dataset είναι tweets, τα οποία έχουν συχνά hashtags που μεταφέρουν το πνεύμα των αντίστοιχων συζητήσεων.

3.6.1 Διερεύνηση διαφορών ταξινόμησης, στο test dataset

Στη συνέχεια προχώρησα σε διερεύνηση των περιπτώσεων όπου το μοντέλο προέβλεψε διαφορετική ταξινόμηση από την επισήμανση που υπήρχε στο dataset. Για το λόγο αυτό χρησιμοποίησα ισορροπημένο dataset από το οποίο αφαιρέθηκαν τυχαία 1.977 tweets, που είχαν επισημανθεί ως μη προσβλητικά. Η προσαρμογή έγινε με βάση τα υπόλοιπα 2.802, από τα οποία δεν αφαίρεσα τα hashtags. Ο λόγος που επέλεξα αυτό το dataset είναι ότι στα

περισσότερα πειράματα, το μοντέλο είχε καλύτερα αποτελέσματα όταν προσαρμοζόταν με ισορροπημένο dataset. Επίσης χρησιμοποίησα τρεις εποχές και σταθερό learning rate = $2e-5$. Στο τέλος της τρίτης εποχής τα metrics ήταν τα εξής: Matthews Correlation Coefficient = 0.6380, F1 = 0.8269 και Accuracy = 0.8174. Παρόλα αυτά όταν δοκίμασα την αποτελεσματικότητα στο test dataset τα metrics ήταν πολύ καλύτερα: Accuracy = 0.89, F1 = 0.90, Matthews Correlation Coefficient = 0.7898. Το test dataset ήταν το 10% του ισορροπημένου, είχε 277 tweets, από τα οποία μισά ήταν προσβλητικά και τα άλλα όχι. Συνολικά 30 από τα 277 ταξινομήθηκαν διαφορετικά σε σχέση με την επισήμανση. Τα tweets αυτά φαίνονται στον παρακάτω πίνακα. Με κόκκινο τονίζω τις λέξεις/φράσεις που θεωρώ ότι ώθησαν το μοντέλο να ταξινομήσει **σωστά** τα tweet ως προσβλητικά, ενώ η επισήμανση των ανθρώπων ήταν (λανθασμένα), μη προσβλητικά. Με μπλε τονίζω τις λέξεις/φράσεις που θεωρώ ότι δικαιώνουν τους ανθρώπους και διέφυγαν από το μοντέλο. Επίσης στις στήλες “Επισήμανση” και “Εκτίμηση Μοντέλου” μαρκάρω έντονο το κείμενο, στη στήλη που θεωρώ ότι κάνει τη σωστή ταξινόμηση. Τέλος με κίτρινο μαρκάρω λέξεις που μπορούν να θεωρηθούν ως προσβλητικές, χωρίς όμως να είναι ξεκάθαρο.

ID	Πρόταση	Επισήμανση	Εκτίμηση Μοντέλου
855	@USER Τονίζουμε τίς λέξεις,... δουλα ;,μς; Βρέ μπάς και είσαι δωδεκάχρονο;	Not Offensive	Offensive
627	Ρε μαλακες.. Τι εκνευριστική φωνή έχει αυτή; #poweroflovegr	Not Offensive	Offensive
2280	Ο Κοντιζάς που το έπαιζε παστρικοθοδώρα στον Παντελή και του έκανε τον βίο αβίωτο, έχει φάει φαΐ σερβιρισμένο από τα βρωμόνυχα της Σπυριδούλας και μύδι με εσάνς πατώματος από τον Μανώλη. Θα δω μέχρι να φάω το βραδινό μου. Μετά θα το κλείσω. #MasterChefGR_Xeftiles #MasterChefGR	Offensive	Not Offensive
2486	ποσοι τραγελαφοι να χωρεσουν σ'ενα τσιρκουλο ο ; ακομα και η γελοιωτητα εχει ενα οριο , ακομα και για τους #νδ_ξεφτιλες	Not Offensive	Offensive
3154	Φιρι-φιρι το παει να χασει. @USER εισαι πολυ κωλοφαρδος λεμε.	Not Offensive	Offensive
4152	@USER Κοτούλα που κρύβεις το πρόσωπο είσαι εσύ. Εμείς στον Σύριζα έχουμε άλλη αντίληψη. Όσο για τις εκλογές κέρδισε ανέλπιστα ηλίθιος κατά γενική ομολογία τον σοβαρό. Πως να αντιμετωπίσεις επίπεδο Κουλη;	Not Offensive	Offensive

ID	Πρόταση	Επισήμανση	Εκτίμηση Μοντέλου
30	@USER Ποσό πρόβατο πρέπει να είσαι για να πιστεύεις ότι ένας εφοπλιστής θα έφερνε ηρωίνη για να βγάλει όσα βγάζει νόμιμα σε 6 μήνες..... ολόκληρη DEA κ δεν έχει ασχοληθεί αλλά.....ξέρετε εσείς τα συριζοτρολ!	Not Offensive	Offensive
3451	#gynaikaxwrisonoma πρόσεχε μαρή Έλλη μην δαγκώσεις την γλώσσα σου Και πάθεις καμία δηλητηρίαση	Offensive	Not Offensive
3474	Όβερ ρε μαλάκες! #turgre	Not Offensive	Offensive
1514	@USER Τα εξωφρενικά είναι βέβαια και άλλα. Όπως η άνεση με την οποία λαιδορρήθηκαν ευσυνειδητοί πολίτες ώστε να γίνουν όλα "ίσα και όμοια" ώστε η Δικαιοσύνη να πάει περίπατο ή η άνεση με την οποία οι λαοπλάνοι πάτησαν και κεφαλαιοποίησαν την αποστροφή του κόσμου.	Not Offensive	Offensive
4582	Τα γαμονυχία τώρα τα είδα #shoppingstar	Offensive	Not Offensive
2111	@USER Ρε Georgy έξυπνος άνθρωπος είσαι σύγκρινε που μίλησε στην Λάρισα ο Κούλης που ο Τσιπρας κανε σύγκριση και μαζέψου, την Δευτέρα έχει φούμο και ψυχρολουσία, μη ανατρέψιμα πράγματα. Μην ποντάρεις σε κουτσό άλογο πόσο μάλλον αν είναι και ζαβό.	Not Offensive	Offensive
1780	Και κάπως έτσι αγαπητοί μου θα δικαιολογήσουν το πάρτυ στο μεγάλο φαγοπότι που ετοιμάζουν... με χρυσά κουτάλια κ μασέλες βαλμένες στην πρίζα! #Εκλογες_2019 #ΝΔ_ξεφτιλες	Not Offensive	Offensive
4224	Ρε παιδιά δεν είναι λίγο βλακεία να είσαι μετανάστης και πατριώτης σε ξένο χώρα και να φωνάζεις έξω οι ξένοι από την χώρα σου;	Not Offensive	Offensive
3417	Γωγώ ας όψονται οι βαθμολογίες που έβαλες όλη την εβδομάδα και η γκαμήλα η κερά Βίκυ. #shoppingstar	Not Offensive	Offensive
1615	Μια ζωή ντουγάνι είσαι ρε Μαρίνα. Έρμαιο της κάθε μηχανορραφίας. #GynaikaXwrisOnoma	Not Offensive	Offensive
4695	@USER Σωστά και ως καλό τετράποδο που είσαι , ενδιαφέρεσαι να μας το υπενθυμίζεις. Δεν παρεξηγούμε ρε βαζέλα. Δεν είμαστε ρατσιστές, και τα ζώα αξίζουν να έχετε ζωή	Offensive	Not Offensive
35	@USER Αξιος και λίγα λες, σπάνιο να βρεις τέτοιο κωλοτουμπα. #ΣΥΡΙΖΑ_ξεφτιλες	Not Offensive	Offensive
2971	@USER Ότι Εσύ είσαι μπεμπεκα δλδ	Not Offensive	Offensive

ID	Πρόταση	Επισήμανση	Εκτίμηση Μοντέλου
4460	Η νέα γενιά, απ' ό,τι αποδεικνύεται, είναι αμόρφωτη. #giatinparea	Offensive	Not Offensive
1676	@USER Επίσης,καλο ΣΚ κ καλο μην...απορω τι ψυχολογικα εχει καποιος να βριζει ετσι τις γυναίκες και κατα ποσο ειναι επικινδυνος γενικα...	Not Offensive	Offensive
756	Μαλάκες δεν είναι αστείο άλλο, ας ενδιαφερθεί κάποιος.	Not Offensive	Offensive
625	γιατί ανοίγω κουβέντα για φεμινισμό σε γυναίκες που βλέπουν 16χρονης με σορτσάκια κ τις κρίνουν; #ταβορ	Not Offensive	Offensive
2497	Οι πιο άθλιες και πιο αντικοινωνικές ερωτήσεις που μπορείς να κανείς:Ποσο χρονων είσαι;Ποσα κιλά είσαι;γιατί δεν έχεις παντρευτεί;γιατί δεν έχεις παιδιά;δεν τα ρωτάμε ποτε αυτά.	Not Offensive	Offensive
4162	@USER Όχι δεν είσαι, αλλά όκ πλήρωσες 14 ευρώ για κάτι που δεν τρώς. Αν το πήρε για σοκολάτα τότε 2 τινά συμβαίνουν. Είναι τρελά ευκολόπιστος (είμαι ευγενική) 1 και 2 το μάρκετινγκ τους γαμάει . Αλλά και η Νουτέλα σοκολατένια πραλίνα φουντουκιού είναι ντάξ Καλοφάγωτη.	Not Offensive	Offensive
2815	'Αλβ ριχτο" !; ..ποιο ακριβως να..ριζει ρε Λαρτζ Πολακη;..το σπιτι σου; Το 1.5 διαφορα που εβγαζε ο Μπαγκς Μπανι ο νεογκεμπελισκος περιμενουμε να μας πειτε που το βρηκατε. #Τσιπρα παιξε μας μια εξτρα μπονους φορολογια να τα παρεις να τα μοιρασεις στους ρομα #ΣΥΡΙΖΑ_ξεφτιλες	Not Offensive	Offensive
4932	#GynaikaXwrisOnoma πόσο βλακας ο Μιχάλης.. Αντί να το πει στη Μαρίνα το είπε στη #τρεΛιλίαν πόσο στόκος πια	Not Offensive	Offensive
2979	Ρε μαλακες	Not Offensive	Offensive
2589	@USER βρε δικηγορος της εισαι; για να ξερω. και δεν σας ειπε κανεις μαλακες . μην βάζεις λεξεις στο στομα μου που δεν εχω πει. σου προτεινω. μια βοτκα με λεμονι ολα τα επουλωνει	Not Offensive	Offensive
1779	@USER Είσαι ωραίος. Σε ξέχασε ο Καμμένος στον ΣΥΡΙΖΑ και από ΧΩΝΙ των ANEL έγινες ΧΩΝΙ των ΣΥΡΙΖΑΙΩΝ ;	Not Offensive	Offensive

Εικόνα 3.27 - Tweets που ταξινομήθηκαν διαφορετικά από το μοντέλο σε σχέση με την επισήμανση

Βλέποντας τα παραπάνω καταλαβαίνουμε πόσο σημαντική είναι η σωστή επισήμανση του κειμένου, η οποία είναι πολλές φορές υποκειμενική και δημιουργεί ψευδώς αρνητικά αποτελέσματα. Κατά τη γνώμη μου, στις 19 από τις 30 ταξινομήσεις όπου υπήρξαν διαφορές,

το μοντέλο έκανε πιο σωστή πρόβλεψη από τους εθελοντές που έκαναν την επισήμανση, στις 10 έκανε λάθος το μοντελο και μία είναι διφορούμενη.

3.6.2 Σύγκριση με αντίστοιχες εργασίες

Θα συγκρίνω τώρα τα αποτελέσματα του μοντέλου μου με άλλες παρόμοιες εργασίες. Συγκεκριμένα θα παρουσιάσω τα αποτελέσματα από αντίστοιχη εργασία [30] όπου συμμετείχαν 528 ομάδες, από τις οποίες υπέβαλαν αποτελέσματα οι 145 και τα σώματα κειμένου ήταν στα Ελληνικά, Δανικά, Αραβικά, Αγγλικά και Τουρκικά και αποτελούνταν από Tweets της πλατφόρμας X, εκτός των Δανικών που προήλθαν από το Facebook και το Reddit. Το ζητούμενο ήταν τρεις υπο-εργασίες, όπου η πρώτη ήταν η αναγνώριση προσβλητικού λόγου. Οι περισσότερες ομάδες χρησιμοποίησαν embeddings για την αναπαράσταση των χαρακτηριστικών και παραλλαγές των μοντέλων BERT, RoBERTa, ALBERT, αλλά επίσης RNNs, CNNs, GRUs και SVMs. Η κατανομή των tweets καθώς και τα αποτελέσματα για το Macro-averaged F1 ανά γλώσσα φαίνονται παρακάτω:

Γλώσσα	Προσβλητικά Tweets	MH Προσβλητικά Tweets	Πλήθος Υπερβολών	Μέγιστο σκορ	Ελάχιστο σκορ	Διάμεσο σκορ
Αγγλικά	1.448.861	7.640.279	81	0.9204	0.0728	0.9088
Αραβικά	1.589	6.411	53	0.9017	0.4451	0.8221
Δανικά	384	2.577	38	0.8119	0.4913	0.7267
Ελληνικά	2.486	6.257	37	0.8522	0.2688	0.8115
Τουρκικά	6.131	25.625	46	0.8258	0.3109	0.7496

Εικόνα 3.28 - Αποτελέσματα ταξινόμησης Tweets, για πέντε γλώσσες

Τα παραπάνω σκορ είναι για Macro-averaged F1 το οποίο όμως είναι ίσο με το F1 όταν έχουμε ισορροπημένο dataset και δυαδική ταξινόμηση. Στα δικά μου αποτελέσματα (εικόνα 3.26) από τους τέσσερις συνδυασμούς που το dataset ήταν ισορροπημένο, οι τιμές του F1 ήταν 0,8701 , 0,8612 , 0,8577 και 0,8390. Επίσης το F1 στο test dataset ήταν 0,90. Συμπεραίνουμε λοιπόν ότι το μοντέλο που έχω εκπαιδεύσει τα πηγαίνει καλύτερα από τα περισσότερα που εκπαιδεύτηκαν για την Ελληνική γλώσσα. Η μόνη γλώσσα για την οποία τα αντίστοιχα μοντέλα δίνουν ξεκάθαρα καλύτερα αποτελέσματα από το δικό μου είναι η

Αγγλική. Επίσης, αν και το καλύτερο μοντέλο για την Αραβική γλώσσα έχει καλύτερο σκορ από το δικό μου, παρόλα αυτά ο διάμεσος (0,8221) είναι χαμηλότερος από όλα τα μοντέλα που έχω δοκιμάσει. Αν παρατηρήσουμε προσεκτικά τον πίνακα στην εικόνα 3.28, βλέπουμε ότι για την Αγγλική γλώσσα, που έχουμε τα καλύτερα αποτελέσματα, τα tweets είναι συνολικά 9.089.140, τα οποία είναι 286 φορές περισσότερα από τη δεύτερη γλώσσα, τα Τουρκικά. Επίσης το χειρότερο διάμεσο σκορ έχουν τα Δανικά, για τα οποία έχουμε και το μικρότερο σώμα κειμένου. Από αυτά μπορούμε να καταλάβουμε πόσο σημαντικό είναι το μέγεθος του σώματος κειμένου. Όταν αυτό είναι πολύ μικρό, όπως στα Ελληνικά έχουμε τεράστιες διαφορές (ελάχιστο 0.2688, μέγιστο 0.8522)

4. Σύνοψη

4.1 Συμπεράσματα

Με βάση την παραπάνω σύγκριση θεωρώ ότι το μοντέλο που προσαρμόστηκε παρουσιάζει αξιόλογα αποτελέσματα, καθώς στις περισσότερες παραλλαγές επιτυγχάνει Accuracy και F1 περισσότερο από 85% καθώς και Matthews Correlation Coefficient από 70% έως 75%, το οποίο αντικατροπτρίζει την ακρίβεια του μοντέλου σε μη ισορροπημένα datasets όπως αυτό που χρησιμοποίησα σε κάποια πειράματα. Το μεγαλύτερο πρόβλημα που αντιμετώπισα ήταν η δυσκολία εύρεσης έτοιμων επισημασμένων dataset στην Ελληνική γλώσσα. Αυτό που χρησιμοποίησα τελικά είχε σχετικά μικρό σύνολο δεδομένων, περίπου 5.000 tweets στα Ελληνικά, και γι αυτό το λόγο δε μπόρεσα να βελτιώσω ακόμη περισσότερο το τελικό μοντέλο.

Επίσης, η χρήση προεκπαιδευμένου μοντέλου (GreekBERT [29]) καθώς και οι βιβλιοθήκες της Hugging Face βοήθησαν ώστε η προσαρμογή (fine tuning) να πραγματοποιηθεί σε σχετικά μικρό χρόνο και με πολύ καλά metrics καθώς η μεταφορά γνώσης από το GreekBERT έδωσε σημαντική ώθηση στη διαδικασία. Η υπόδειξη της Hugging Face ότι η προσαρμογή συνήθως επιτυγχάνεται σε περίπου τρεις εποχές φαίνεται να επιβεβαιώνεται και από τα αποτελέσματα του πίνακα στην εικόνα 3.26. Παράλληλα, το γεγονός ότι το dataset είναι μη ισορροπημένο δε φαίνεται να επηρέασε ιδιαίτερα την προσαρμογή. Διότι ναι μεν η προσαρμογή με ισορροπημένο dataset είχε ελαφρώς καλύτερα αποτελέσματα, από την άλλη επειδή το dataset μίκρυνε και άλλο (60% του αρχικού) δεν έδινε μεγάλες διαφορές στα metrics. Εδώ θα βοηθούσε ιδιαίτερα ένα πολύ μεγαλύτερο dataset.

4.2 Βελτιώσεις - μελλοντικές επεκτάσεις

Έχοντας δοκιμάσει διάφορους συνδυασμούς υπερ παραμέτρων, επιλογή ισορροπημένων και μη ισορροπημένων dataset, καθώς και διατήρηση ή αφαίρεση των hashtags, θεωρώ ότι ο βασικότερος παράγοντας που περιορίζει την αποτελεσματικότητα του μοντέλου είναι το μικρό μέγεθος του dataset. Η άποψή μου αυτή ενισχύεται και από τη σύγκριση με τα αποτελέσματα άλλων ομάδων (Εικόνα 3.28), με τα καλύτερα αποτελέσματα να επιτυγχάνονται στην Αγγλική γλώσσα, όπου το dataset είναι πολύ μεγάλο. Έτσι ο αρχικός στόχος οποιασδήποτε εργασίας με αντικείμενο την ταξινόμηση Ελληνικών κειμένων σε κοινωνικά δίκτυα θα πρέπει να είναι η συλλογή μεγάλου όγκου αναρτήσεων. Επίσης αυτές θα πρέπει να έχουν ποικιλία στο περιεχόμενο, στο χρονικό διάστημα που έχουν δημοσιευτεί, καθώς και να προέρχονται από διαφορετικές πλατφόρμες, αν αυτό είναι εφικτό. Πολύ βασικός παράγοντας είναι και η σωστή επισήμανση του dataset, καθώς όπως φαίνεται στην διερεύνηση των διαφορών (3.6.1) αρκετά tweets έχουν, κατά τη γνώμη μου, λάθος επισήμανση. Για να γίνει βέβαια η επισήμανση σε μεγάλο όγκο δεδομένων θα απαιτηθεί πολύς χρόνος και αρκετοί εθελοντές, πράγμα που το καθιστά δύσκολο στα πλαίσια μιας τυχαίας εργασίας.

Αντί για ανθρώπους που θα κάνουν την επισήμανση ίσως να μπορεί να γίνει εκτίμηση από τις online πλατφόρμες που βασίζονται στα LLMs, όπως το ChatGPT, το Gemini κ.α. Αυτές παρέχουν APIs στα οποία θα μπορούν να αποστέλλονται τα κείμενα προς επισήμανση με ερώτηση αν το περιεχόμενο είναι προσβλητικό. Στη συνέχεια μπορούμε να θεωρούμε το κείμενο ως προσβλητικό αν η πλειοψηφία των πλατφορμών επιστρέφει “NAI” αλλιώς θα το θεωρούμε μη προσβλητικό. Ένα πρόβλημα που υπάρχει για αυτή την υλοποίηση, είναι ότι ακόμη οι περισσότερες πλατφόρμες δεν υποστηρίζουν Ελληνικά, αλλά αυτό θα αλλάξει σύντομα καθώς οι εταιρείες προσθέτουν συνέχεια νέα χαρακτηριστικά και γλώσσες. Τέλος υπάρχει και το θέμα του κόστους, καθώς η ελεύθερη χρήση αυτών των API έχει περιορισμούς στο πλήθος των ερωτήσεων ανά χρονική περίοδο, άρα θα απαιτηθεί αρκετός χρόνος αν το σώμα κειμένου είναι μεγάλο.

Επίσης άλλη μια βελτίωση, θα ήταν να επεκταθεί το μοντέλο και αντί να κάνει δυαδική ταξινόμηση (Προσβλητικό περιεχόμενο ή Μη Προσβλητικό) να ταξινομεί σε περισσότερες κατηγορίες, όπως προσβλητικό με βάση τη φυλή, το θρήσκευμα, τις σεξουαλικές προτιμήσεις κ.α. Αυτό βέβαια, κάνει το έργο της σωστής επισήμανσης του σώματος κειμένου ακόμη πιο δύσκολη.

5. Κώδικας υλοποίησης και προσαρμοσμένο μοντέλο

Ο κώδικας που χρησιμοποίησα για την υλοποίηση της διπλωματικής εργασίας βρίσκεται στο [Google Colab](#) και το προσαρμοσμένο μοντέλο βρίσκεται σε [αποθετήριο της Hugging Face](#)

6. Βιβλιογραφία

- [1] P. Tan, M. Steinbach, A. Karpatne, V. Kumar. Εισαγωγή στην Εξόρυξη Δεδομένων, 2η Έκδοση, 2021
- [2] S. Russell, P. Norvig. Τεχνητή Νοημοσύνη, Μια σύγχρονη προσέγγιση, 4η αμερικανική έκδοση, 2021
- [3] Activation function, https://en.wikipedia.org/wiki/Activation_function, 24/02/2024
- [4] Perceptron: Concept, function, and applications, <https://datascientest.com/en/perceptron-definition-and-use-cases>, 25/02/2024
- [5] Multi-Layer Neural Network, <http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks>, 25/02/2024
- [6] What is deep learning?, <https://www.ibm.com/topics/deep-learning>, 25/02/2024
- [7] Introduction to Convolutional Neural Networks (CNN), <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn>, 25/02/2024
- [8] Convolutional Neural Networks, Explained, <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>, 25/02/2024
- [9] An Introduction to Convolutional Neural Networks (CNNs), <https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns>, 25/02/2024
- [10] Deep Neural Network: The 3 Popular Types (MLP, CNN and RNN)
Read more at: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types>, 25/02/2024
- [11] Long short-term memory, https://en.wikipedia.org/wiki/Long_short-term_memory, 25/02/2024
- [12] S. Hochreiter, The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 6(2):107-116, April 1998
- [13] Understanding LSTM Networks, <https://colah.github.io/posts/2015-08-Understanding-LSTMs>, 25/02/2024
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, arXiv:1706.03762

- [15] The Transformer Model, <https://machinelearningmastery.com/the-transformer-model>, 25/02/2024
- [16] Text Encoding: A Review, <https://www.kdnuggets.com/2019/11/text-encoding-review.html>, 06/04/2024
- [17] N-gram Language Modeling in Natural Language Processing, <https://www.kdnuggets.com/2022/06/ngram-language-modeling-natural-language-processing.html>, 06/04/2024
- [18] tf-idf, <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>, 06/04/2024
- [19] Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin, <https://web.stanford.edu/~jurafsky/slp3/14.pdf>
- [20] Word2vec, <https://en.wikipedia.org/wiki/Word2vec>, 06/04/2024
- [21] Vector Space Models, <https://towardsdatascience.com/vector-space-models-48b42a15d86d>, 06/04/2024
- [22] Text Preprocessing Methods for Deep Learning, <https://www.exxactcorp.com/blog/Deep-Learning/text-preprocessing-methods-for-deep-learning>, 06/04/2024
- [23] J. Landthaler, B. Walzl, D. Huth, Extending Thesauri Using Word Embeddings and the Intersection Method, Second Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2017), London, UK
- [24] BERT language model, <https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model>, 07/04/2024
- [25] Word2Vec vs BERT, <https://www.saltdatalabs.com/blog/word2vec-vs-bert>, 07/04/2024
- [26] Encoder models, <https://huggingface.co/learn/nlp-course/chapter1/5?fw=tf>, 07/04/2024
- [27] 10 Leading Language Models For NLP In 2022, <https://www.topbots.com/leading-nlp-language-models-2020/>, 07/04/2024
- [28] Transformer Architectures for Dummies - Part 2 (Decoder Only Architectures), <https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-qi6vc/>, 07/04/2024
- [29] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, I. Androutsopoulos, GreekBERT: The Greeks Visiting Sesame Street, 11th Hellenic Conference on Artificial Intelligence 2020, Athens, Greece, ACM ISBN 9781450388788
- [30] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, C. Coltekin, SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media, OffenseEval 2020