**Τίτλος Εργασίας :** Machine Generated Text detection

& Attribution

Ον/μο Φοιτητή : Ζαχάρης Γεώργιος

Major：Artificial Intelligence

Name：Zacharis Georgios

Thesis Advisor:Dr.Gounaridis Ioannis

Due Date：[Type the due date]

UEL No :

Επιβλέπων Καθηγητής : Δρ./Κος. Γουναρίδης Ιωάννης

**Περίληψη**

H ταχεία ανάπτυξη του τομέα της τεχνητής νοημοσύνης τα τελευταία χρόνια έχει αποτελέσει σημαντικό παράγοντα στην διευκόλυνση της καθημερινής ζωής , μιας και πλέον η τεχνητή νοημοσύνη είναι θεμελιώδης βάση για πολλές εφαρμογές. Ένα νέο βήμα της τεχνητής νοημοσύνης που έχει αναπτυχθεί τον τελευταίο καιρό είναι η δημιουργία μοντέλων παραγωγής κειμένου .Τα μοντέλα αυτά έχουν δημιουργηθεί με την βοήθεια μοντέλων επεξεργασίας της φυσικής γλώσσας, κάτι το οποίο μπορεί να έχει αντίκτυπο στην καθημερινότητα της ανθρωπότητας . Κύριος στόχος της δημιουργίας των συγκεκριμένων μοντέλων ήταν η βοήθεια που μπορούν να προσφέρουν στην απάντηση ερωτημάτων ανθρώπινης προέλευσης .Οι απαντήσεις μπορούν να ποικίλουν σε διαφορετικούς τομείς από ερωτήσεις για την καθημερινότητα,τις επιχειρήσεις μέχρι και στην απάντηση πιο στοχευμένων ερωτημάτων περί του τεχνικού τομέα.Η εξέλιξη αυτή θέτει ερωτήματα όσον αφορά την αυθεντικότητα του κειμένου που αναφέρεται κάπου στο διαδίκτυο και την παραπληροφόρηση που είναι ικανά να δημιουργήσουν,κυρίως στα μέσα κοινωνικής δικτύωσης.Ο μεγαλύτερος αντίκτυπος των ενεργειών εκμετάλλευσης των αποτελεσμάτων των μοντέλων παραγωγής κειμένου εμφανίζεται κυρίως στον τομέα των μέσων κοινωνικής δικτύωσης.Σε αυτό το περιβάλλον υπάρχουν πάρα πολλοί τρόποι χρήσης των κειμένων αυτών όπως είναι η κερδοσκοπικές πράξεις , η χειραγώγηση του κοινού για προσωπικούς κερδοσκοπικούς λόγους και η δημιουργία καλής εντύπωσης βασισμένη σε ψεύτικα δημιουργημένα από μηχανή δεδομένα. Η παρούσα διατριβή ερευνά την ανάπτυξη και την εκτίμηση ενός μοντέλου το οποίο θα είναι ικανό να διακρίνει την διαφορά μεταξύ ενός tweet που έχει παραχθεί από άνθρωπο με ένα tweet που έχει παραχθεί από τεχνητή νοημοσύνη.Τα βασικά στοιχεία που περιλαμβάνει η έρευνα αφορούν ένα σύνολο δεδομένων από την εφαρμογή Χ , έναν συνδυασμό μοντέλων ταξινόμησης κειμένου και την αξιοποίηση ενός μοντέλου deep learning για να πρόβλεψη μεταξύ κειμένου παραγόμενο από άνθρωπο ή από μηχανή.Στην προσέγγιση αυτή το deep learning μοντέλο με την βοήθεια των αλγορίθμων ταξινόμησης είναι ικανό να εντοπίζει την διαφορά στο ύφος και την δομή μεταξύ των δύο τύπων κειμένων.Η παρούσα προσπάθεια λειτουργεί ως βοήθεια για την αντιμετώπιση της παραπληροφόρησης που μπορούν να προκαλέσουν τα κείμενα παραγόμενα από μηχανή στην πλατφόρμα X .Λόγω της συνεχούς εξέλιξης των συστημάτων των μοντέλων παραγωγής κειμένου και την ικανότητά τους να αποφεύγουν πολλά προγράμματα στο να εντοπίζουν το κείμενο το οποίο παράγουν,πλέον απαιτείται μεγάλη προσπάθεια για τον εντοπισμό τους.Έτσι τα εργαλεία εντοπισμού έχουν αρχίσει να μην είναι ικανά να συμβαδίσουν με την εξέλιξη τους.Η παρούσα διατριβή εστιάζει στην βελτίωση των μοντέλων αυτών με σκοπό την επιτυχή αναβάθμιση των δυνατοτήτων τους.

Abstract

The development of the field of artificial intelligence in recent years has been an important factor in facilitating everyday life, as artificial intelligence is now a fundamental basis for many applications. A new step in artificial intelligence that has been developed in recent times is the creation of text generation models .These models have been created with the help of natural language processing models , which may have an impact on the everyday life of mankind . The main aim of creating these models was to help them to provide help in answering questions of human origin .The answers can vary in different areas from questions about everyday life , business to answering more targeted questions about the technical field .This development raises questions about the authenticity of the text mentioned somewhere on the internet and the misinformation they are capable of creating, especially in social media.The greatest impact of the actions of exploiting the results of text production models occurs mainly in the social media sector.In this environment there are many ways of using these texts, such as speculative acts, manipulation of the public for personal profit and publicity, and the use of the text for the purpose of public relations. This thesis investigates the development and evaluation of a model that is able to distinguish the difference between a human-generated tweet and a tweet generated by artificial intelligence.The key elements of the research involve a dataset from application X , a combination of text classification models and the use of a deep learning model to predict between human or machine generated text.In this approach the deep learning model with the help of classification algorithms is able to identify the difference in style and structure between the two types of text.This effort serves to help address the misinformation that can be caused by machine generated text on the X platform.Due to the continuous evolution of text generation model systems and their ability to avoid many programs in detecting the text they produce, a great deal of effort is now required to detect them.Thus detection tools have begun to be unable to keep up with their evolution.This thesis focuses on improving these models.

**Ευχαριστίες**

Πρωτίστως, θα ήθελα να εκφράσω την βαθύτατη ευγνωμοσύνη μου στον επιβλέποντα καθηγητή μου,Δρ Γουναρίδη Ιωάννη για την καθοδήγηση,την ενθάρρυνση που πρόσφερε και το χρόνο που διέθεσε για την ολοκλήρωση της διπλωματικής αυτής.Θα ήθελα επίσης να εκφράσω θερμές ευχαριστίες προς το εκπαιδευτικό προσωπικό του Computer Science του Μητροπολιτικού Κολεγίου για την υποστήριξη και την βοήθεια που παρείχαν σε όλη την διάρκεια των ακαδημαϊκών μου σπουδών.Η προσφορά τους στην καλλιέργεια ενός φιλικού μαθησιακού περιβάλλοντος μέχρι και την τελευταία εβδομάδα φοίτησης εκτιμάται ιδιαίτερα.

Η παρούσα διπλωματική αφιερώνεται σε όλους όσους με στήριξαν και πίστεψαν σε εμένα .Οι συνεισφορές τους,είτε μεγάλες είτε μικρές βοήθησαν αρκετά για την επίτευξη του στόχου αυτού.

Σας ευχαριστώ όλους.

ΠΕΡΙΕΧΟΜΕΝΑ

**[1.Εισαγωγή 1](#_Toc25903)**

1.1 Yπόβαθρο ML..............................................................................................................................1

1.2 Aναφορά Προβλήματος...............................................................................................................3

1.3 Στόχος Διατριβής.........................................................................................................................3

1.4 Οργάνωση Διατριβής...................................................................................................................4

**2.Μachine Learning Background...........................................................................................................5**

2.1 Classification & Generalization...................................................................................................5

2.2 Data Preprocessing.......................................................................................................................7

2.3 Machine Learning Algorithms.....................................................................................................8

2.4 BERT Model..............................................................................................................................10

2.4.1 Fine-Tuning......................................................................................................................13

2.4.2 Feature-Based...................................................................................................................14

**[3. Μεθοδολογία 15](#_Toc25329)**

[3.1 Library Imports 15](#_Toc17244)

[3.2 Data Preparation & Preprocessing 16](#_Toc27830)

[3.3 Data Splitting & Model 18](#_Toc18891)

[3.4 Τraining & Validation 20](#_Toc15440)

[3.5 Testing 22](#_Toc18402)

**[4. Αποτελέσματα-Ευρήματα 24](#_Toc14972)**

4.1 Dataset & Preprocessing............................................................................................................24

4.2 Training & Results.....................................................................................................................24

4.3 Validation...................................................................................................................................25

4.4 Results........................................................................................................................................26

4.4.1 Accuracy.........................................................................................................................27

4.4.2 Precision..........................................................................................................................27

4.4.3 Recall..............................................................................................................................28

4.4.4 F1-Score..........................................................................................................................28

4.4.5 Compare Results.............................................................................................................29

4.4.6 Errors.............................................................................................................................. 30

**[5. Συμπεράσματα & Μελλοντικό Έργο 32](#_Toc9902)**

[5.1 Συμπεράσματα 32](#_Toc19016)

[5.2 Μελλοντικό Έργο 33](#_Toc19029)

**[Βιβλιογραφία 35](#_Toc24402)**

**1.Εισαγωγή**

1.1Υπόβαθρο ML

Έχει γίνει κοινωνικά αποδεκτό ότι τα τελευταία 20 χρόνια η τεχνητή νοημοσύνη(Artificial Intelligence) έχει κάνει ένα άλμα όσον αφορά την βοήθεια της προς την ανθρωπότητα.Πιο συγκεκριμένα,η μηχανική μάθηση είναι ένας κλάδος της τεχνητής νοημοσύνης ο οποίος αφορά την εξαγωγή γνώσης από τα δεδομένα. Είναι ένα ερευνητικό πεδίο το οποίο διασταυρώνει την στατιστική, την τεχνητή νοημοσύνη και την επιστήμη των υπολογιστών και είναι επίσης γνωστή ως προγνωστική ανάλυση ή στατιστική μάθηση(Andreas C.Muller p.2).Η εφαρμογή μεθόδων μηχανικής μάθησης έχει χρησιμοποιηθεί τα τελευταία χρόνια παρούσα στην καθημερινή ζωή. Από αυτόματες συστάσεις για το ποιες ταινίες να παρακολουθήσει κάποιος, μέχρι το τι φαγητό να παραγγείλει ή ποια προϊόντα να αγοράσει, μέχρι το εξατομικευμένο διαδικτυακό ραδιόφωνο, πολλοί σύγχρονοι ιστότοποι και συσκευές διαθέτουν αλγόριθμους μηχανικής μάθησης στον πυρήνα τους. Όταν εξετάζεται ένας πολύπλοκος ιστότοπος όπως το Facebook, το Amazon ή το Netflix, είναι πολύ πιθανό ότι σε κάθε τμήμα του ιστότοπου να περιέχονται πολλαπλά μοντέλα μηχανικής μάθησης για διάφορες λειτουργίες .Εκτός από τις εμπορικές εφαρμογές, η μηχανική μάθηση είχε τεράστια επιρροή στον τρόπο με τον οποίο γίνεται σήμερα η έρευνα με βάση τα δεδομένα.

Η μηχανική μάθηση εφαρμόζεται ευρέως σε πολλούς κλάδους. Τα recommendation engines,για παράδειγμα, τα οποία χρησιμοποιούνται από την ηλεκτρονική αγορά,τα μέσα κοινωνικής δικτύωσης και τους ειδησεογραφικούς οργανισμούς για να προτείνουν περιεχόμενο με βάση την συμπεριφορά και τις ενέργειες ενός πελάτη(Andreas C.Muller p.3). Στην υγειονομική περίθαλψη, η μηχανική μάθηση χρησιμοποιείται για τη διάγνωση και την δημιουργία θεραπευτικών σχεδίων. Άλλες συνήθεις περιπτώσεις χρήσης ML περιλαμβάνουν την ανίχνευση απάτης, το φιλτράρισμα ανεπιθύμητης αλληλογραφίας, την ανίχνευση απειλών από κακόβουλο λογισμικό, την προγνωστική συντήρηση και την αυτοματοποίηση επιχειρηματικών διαδικασιών.

Ενώ η μηχανική μάθηση είναι ένα ισχυρό εργαλείο για την επίλυση προβλημάτων, τη βελτίωση των επιχειρηματικών λειτουργιών και την αυτοματοποίηση εργασιών, είναι επίσης μια πολύπλοκη και απαιτητική τεχνολογία, που απαιτεί βαθιά τεχνογνωσία και σημαντικούς πόρους. Η επιλογή του κατάλληλου αλγορίθμου για μια εργασία απαιτεί ισχυρή κατανόηση των μαθηματικών και της στατιστικής. Η εκπαίδευση αλγορίθμων μηχανικής μάθησης συχνά περιλαμβάνει μεγάλο όγκο δεδομένων καλής ποιότητας για την παραγωγή καλών αποτελεσμάτων.Τα ίδια τα αποτελέσματα μπορεί να είναι δύσκολο να γίνουν κατανοητά - ιδίως τα αποτελέσματα που παράγονται από πολύπλοκους αλγορίθμους, όπως τα νευρωνικά δίκτυα deep learning.Ο στόχος της τεχνητής νοημοσύνης είναι να δημιουργηθούν μοντέλα υπολογιστών που να επιδεικνύουν "ευφυείς συμπεριφορές" όπως οι άνθρωποι, σύμφωνα με τον Boris Katz, κύριο ερευνητή και επικεφαλής της ομάδας InfoLab στο CSAIL. Αυτό σημαίνει μηχανές που μπορούν να κατανοούν ένα κείμενο γραμμένο σε φυσική γλώσσα ή να εκτελούν μια ενέργεια στον φυσικό κόσμο.

Η μηχανική μάθηση είναι ένας τρόπος για να χρησιμοποιηθεί η Τεχνητή Νοημοσύνη. Ορίστηκε τη δεκαετία του 1950 από τον πρωτοπόρο της ΤΝ Άρθουρ Σάμιουελ ως "το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν χωρίς να προγραμματίζονται ρητά".

Ο ορισμός ισχύει, σύμφωνα με τον (Mikey Shulman), λέκτορα στο MIT Sloan και επικεφαλής της μηχανικής μάθησης στην Kensho, η οποία ειδικεύεται στην τεχνητή νοημοσύνη για τις κοινότητες των οικονομικών και των αμερικανικών μυστικών υπηρεσιών. Συγκρίνει τον παραδοσιακό τρόπο προγραμματισμού των υπολογιστών, ή "λογισμικό 1.0", με τη ζαχαροπλαστική, όπου η συνταγή απαιτεί ακριβείς ποσότητες υλικών και λέει στον αρτοποιό να ανακατέψει για ένα ακριβές χρονικό διάστημα. Ο παραδοσιακός προγραμματισμός απαιτεί ομοίως τη δημιουργία λεπτομερών οδηγιών που πρέπει να ακολουθήσει ο υπολογιστής.

Είναι γενικά αντιληπτό ότι τα LLM(Large Language Models) είναι συστήματα τα οποία εκπαιδεύονται για να προβλέπουν την πιθανότητα εμφάνισης ενός χαρακτήρα,μίας λέξης ή μιας συμβολοσειράς σε ένα συγκεκριμένο πλαίσιο (Bender et all.2021). Τα LLM μοντέλα είναι σε χρήση από την δεκαετία του 1980 ,τα οποία χρησίμευαν για μετάφραση και αυτόματη αναγνώριση ομιλίας (Debora Weber-Wulff et. All 2023).Για παράδειγμα το ChatGTP είναι ένα chat-bot τεχνητής νοημοσύνης το οποίο χρησιμοποιεί επεξεργασία φυσικής γλώσσας που ανήκει και αναπτύσσεται από την OpenAI,μια εταιρεία έρευνας και ανάπτυξης που ιδρύθηκε το 2015.To ChatGTP όπως και άλλα μοντέλα είναι ικανά να μιμηθούν την ανθρώπινη γλώσσα με τεράστια ακρίβεια κάνοντάς το δύσκολο στους χρήστες του διαδικτύου να καταλάβουν αν ένα κείμενο είναι γραμμένο από άνθρωπο ή από τεχνητή νοημοσύνη.Αν και το ChatGTP όπως και άλλα παρόμοια μοντέλα βοηθούν στην δημιουργία ιδεών και στην δημιουργία κώδικα για οποιαδήποτε χρήση,κρύβει κινδύνους όπως η παραπληροφόρηση και η δυσφήμηση.

1.2 Αναφορά Προβλήματος

Το πρόβλημα που καλείται να επιλύσει η διατριβή όσον αφορά τον εντοπισμό κειμένου που παράγεται από μηχανές είναι η εξάπλωση των κειμένων και γενικότερα του περιεχομένου που παράγεται από chat-bots στα κοινωνικά δίκτυα. Το κυρίαρχο πρόβλημα είναι ότι το περιεχόμενο που δημιουργείται από τα chat-bots μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς.Κάτι τέτοιο είναι η διάδοση της παραπληροφόρησης ,η παραπλάνηση της κοινής γνώμης για πολιτικούς σκοπούς και η εμπλοκή σε επικίνδυνες δραστηριότητες όπως η εξαπάτηση για κερδοφόρους σκοπούς.

Ο εντοπισμός κειμένου κατασκευασμένο από τεχνητή νοημοσύνη είναι εξαιρετικά σημαντικός για την διατήρηση της αξιοπιστίας του διαδικτυακού περιβάλλοντος καθώς έχει την δυνατότητα να μειώσει τις προσπάθειες παραπληροφόρησης και εξαπάτησης.

1.3 Στόχος της διατριβής

O στόχος της διατριβής είναι η ανάπτυξη και και η αξιολόγηση μεθόδων και μοντέλων για την αναγνώριση του περιεχομένου που παράγεται από chat-bots στο twitter.Ο στόχος συνεπάγεται στην μελέτη των χαρακτηριστικών του κειμένου που παράγεται από τεχνητή νοημοσύνη και στην ανάπτυξη μοντέλων τα οποία είναι ικανά να κάνουν διακρίσεις μεταξύ κείμενου παραγόμενο από άνθρωπο και κείμενου παραγόμενο από μηχανή.Κύριος στόχος της παρούσας διατριβής είναι η συνεισφορά στην ανάπτυξη εργαλείων για την καταπολέμηση της παραπληροφόρησης,της χειραγώγησης και των κακόβουλων δραστηριοτήτων στα μέσα κοινωνικής δικτύωσης.

1.4 Οργάνωση της διατριβής

Η συγκεκριμένη διατριβή υιοθετεί μια οργανωμένη προσέγγιση με στόχο την διερεύνηση και την ανάλυση της ανίχνευσης ενός κειμένου παραγόμενο από τεχνητή νοημοσύνη.Η οργάνωση της διατριβής περιγράφεται παρακάτω:

1. Εισαγωγή:Στο εισαγωγικό κεφάλαιο γίνεται μία γενικευμένη παρουσίαση του θέματος και των στόχων της έρευνας.Επιπλέον καθορίζει το ερευνητικό πρόβλημα και παρουσιάζει τα κεφάλαια που θα ακολουθήσουν.
2. Βιβλιογραφική Ανασκόπηση:Το κεφάλαιο αυτό επιχειρεί μια λεπτομερή εξέταση της βιβλιογραφίας που προορίζεται για την συγκεκριμένη έρευνα , η οποία αφορά τον εντοπισμό κειμένου που παράγεται από την τεχνητή νοημοσύνη.Κάνει μια σύνθεση ερευνών και θεωρητικών γνώσεων για την έρευνα.
3. Μεθοδολογία:Το κεφάλαιο αυτό περιγράφει λεπτομερώς την μεθοδολογία που χρησιμοποιήθηκε.Αναλυτικότερα, αναλύει τις διαδικασίες που ακολουθήθηκαν για την συλλογή δεδομένων, την προ επεξεργασία ,την διαμόρφωση των χαρακτηριστικών και την επιλογή του μοντέλου.
4. Αποτελέσματα:Το κεφάλαιο αυτό παρουσιάζει τα πορίσματα που προκύπτουν από την ανάλυση των δεδομένων του twitter. Περιγράφει το σύνολο δεδομένων που χρησιμοποιήθηκε,των σχολιασμό των αποτελεσμάτων , την τελειοποίηση των μοντέλων.
5. Συμπεράσματα:Το κεφάλαιο αυτό αποτελεί το τέλος της έρευνας. Περιέχει μια σύνοψη των πιο βασικών ευρημάτων,επαναλαμβάνει τις συνεισφορές της έρευνας και προβληματίζεται για την γενικότερη σημασία της μελέτης .Αναδεικνύει την αξία των αποτελεσμάτων και προτείνει τις κατευθύνσεις για περαιτέρω μελλοντική έρευνα.
6. Αναφορές:Το κεφάλαιο αυτό παρέχει μια πλήρη λίστα όλων των πηγών που αναφέρονται σε όλη την διατριβή

**2.Machine Learning**

2.1 Classification & Generalization

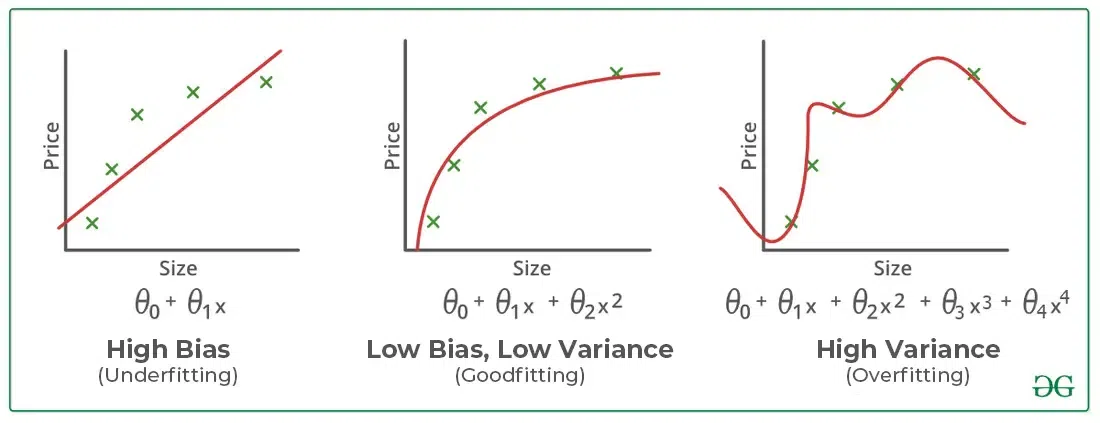
Στο Classification,ο κύριος στόχος της διεργασίας είναι η πρόβλεψη ενός class label το οποίο είναι ένα μέρος ενός προκαθορισμένου κατάλογου πιθανοτήτων[Andreas C.Muller p.25].Το Classification διαχωρίζεται σε διαφορετικά είδη ταξινομήσεων , όπως η δυαδική ταξινόμηση η οποία είναι μια περίπτωση διάκρισης δυο ακριβώς κλάσεων και σε πολυταξική ταξινόμηση η οποία είναι η ταξινόμηση μεταξύ περισσότερων κλάσεων.Γενικότερα η δυαδική ταξινόμηση αναφέρεται σε μια θετική και μια αρνητική κλάση.

Στην επιβλεπόμενη μάθηση(Supervised Learning)[Andreas C.Muller p.26] πρέπει να δημιουργηθεί ένα μοντέλο με δεδομένα εκπαίδευσης και στην συνέχεια να είναι σε θέση να κάνει ορθές προβλέψεις σε απροσδιόριστα δεδομένα τα οποία έχουν παρόμοια ή και ίδια χαρακτηριστικά με τα δεδομένα που χρησιμοποιήθηκαν στο κομμάτι εκπαίδευσης.Για να γίνει αυτό πρέπει το αρχείο των δεδομένων να μήν έχει λάθη κατά την προσθήκη του στο μοντέλο. Στην περίπτωση που το μοντέλο μπορεί να κάνει ακριβείς προβλέψεις σε δεδομένα που δεν γνωρίζει τότε το μοντέλο είναι σε θέση να κάνει γενίκευση.Η γενίκευση με την μεγαλύτερη δυνατή ακρίβεια είναι η πιο αποτελεσματική ,καθώς ότι νέα δεδομένα εκπαίδευσης πρόκειται να πάρει θα τα διαχειριστεί πολύ αποτελεσματικά.Στην περίπτωση που ένα μοντέλο ανταποκρίνεται σωστά σε ένα training set ,αλλά δεν αποδίδει καλά σε νέα δεδομένα που μπορεί να του ανατεθούν τότε κάνει overffiting[].Υπάρχει μοντέλο το οποίο έχει την ικανότητα να αποστηθίσει ένα training set ,δηλαδή να μάθει την σωστή απάντηση για κάθε νέο set χωρίς να μπορεί να κάνει την ταξινόμησή του. Το μόνο πρόβλημα του συγκεκριμένου μοντέλου είναι ότι παρουσιάζει πολύ μεγάλη χωρητικότητα λόγω της ικανότητας της αποστήθισης του training set.Επιπλέον τα δεδομένα αποθηκεύονται σε κάθε παράμετρο του μοντέλου,αυτό σημαίνει ότι ένα μοντέλο το οποίο περιέχει περισσότερες παραμέτρους στον πυρήνα του μπορεί να αποστηθίσει περισσότερα δεδομένα[8].

Για την καλύτερη απόδοση ενός μοντέλου πρέπει να υπάρχει καλύτερη γενίκευση(Generalization).Για να γίνει αυτό διαχωρίζονται τα δεδομένα σε τρεις κατηγορίες όπου η μια είναι το Training data set ,η άλλη το Validation data set και η τελευταία το Test data set[9].Το Training data set είναι ένα αρχείο δεδομένων με το οποίο θα γίνει η εκπαίδευση του μοντέλου,το Validation set χρησιμοποιείται για την επεξεργασία των υπερπαραμέτρων (hyperparameters) και το test set χρησιμοποιείται για να υπολογιστεί η απόδοση της γενίκευσης ,δηλαδή η απόδοση της ικανότητας της λειτουργίας του σε νέα δεδομένα.Αναλυτικότερα είναι αναγκαίο να υπάρχει ένα training data set και ένα test data set ,καθώς εάν εκπαιδευτεί το μοντέλο στα δεδομένα δοκιμής (test data) δεν είναι γνωστό αν το μοντέλο θα κάνει σωστή γενίκευση ή απλά θα αποστηθίζει τα δεδομένα εκπαίδευσης.Όσον αφορά τα hyperparameters δεν μπορεί να γίνει η ρύθμισή τους στα δεδομένα εκπαίδευσης ή στα δεδομένα δοκιμής διότι χρειάζονται values τα οποία θα κάνουν γενίκευση ,στα δεδομένα εκπαίδευσης και στα δεδομένα δοκιμής δεν μπορεί να γίνει γενίκευση καθώς έχουν άλλες λειτουργίες.Ο πιο συνηθισμένος τρόπος για την ρύθμιση των υπερπαραμέτρων είναι το grid search [10]. Στο grid search , για κάθε παράμετρο επιλέγεται ένα σύνολο τιμών το οποίο μπορεί να αλλάζει μέχρι να βρεθεί το καλύτερο δυνατό αποτέλεσμα.Μια εναλλακτική στο grid search είναι το random search το οποίο προκαλεί τυχαίες διαμορφώσεις των υπερπαραμέτρων μέχρι να βρεθεί το καλύτερο δυνατό αποτέλεσμα.Το πλεονέκτημα του random search έναντι του grid search είναι ότι το random search υπάρχει περίπτωση να βρει τον ακριβή διάταξη των παραμέτρων για την καλύτερη γενίκευση του αλγορίθμου.Έτσι με οποιονδήποτε από τους δύο τρόπους μπορεί να γίνει το validation set το οποίο είναι σημαντικό για μια καλύτερη γενίκευση.

Το underffiting είναι το αντίθετο του overffiting κάτι που σημαίνει ότι το μοντέλο ή ο αλγόριθμος μηχανικής μάθησης δεν είναι αρκετά σύνθετος για να αναλύσει με ακρίβεια τα δεδομένα τα οποία θα λάβει από το αρχείο. Ένα σημείο που προδίδει ότι έχει γίνει underffiting είναι ότι στο μοντέλο που χρησιμοποιείται εντοπίζεται υψηλό bias και χαμηλό variance(το ακριβώς αντίθετο με την υπερπροσαρμογή η οποία έχει χαμηλό bias και υψηλό variance)[11].Γενικότερα το underffiting μπορεί να διαπιστωθεί από το bias-variance tradeoff, το οποίο ,το οποίο είναι η μέθοδος που αναλύεται ένα μοντέλο ή αλγόριθμος για bias σφάλμα,variance σφάλμα και irreducible σφάλμα. Στην περίπτωση υψηλού bias και χαμηλού variance το αποτέλεσμα που προκύπτει είναι ότι θα αναπαριστά εσφαλμένα τα σημεία δεδομένων και επομένως θα είναι θα είναι ανεπαρκώς ικανό να προβλέψει μελλοντικά αποτελέσματα δεδομένων.Οι λόγοι που συμβαίνει underffiting είναι οι εξής :

* Το μοντέλο είναι υπερβολικά απλό.Έτσι μπορεί να μην είναι ικανό να παρουσιάσει πολύπλοκα μοτίβα στα δεδομένα.
* Τα χαρακτηριστικά εισόδου που χρησιμοποιούνται για την εκπαίδευση του μοντέλου δεν είναι επαρκή .
* Η υπερβολική τακτοποίηση χρησιμοποιείται για να αποτραπεί το overffiting ,κάτι που περιορίζει το μοντέλο να συλλέξει καλά τα δεδομένα.



Εικόνα 1

2.2 Data Preprocessing

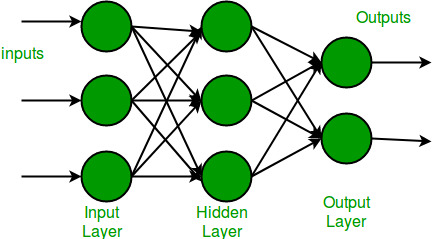
Η προεπεξεργασία των δεδομένων πριν την εισαγωγή τους στον πυρήνα του νευρωνικού δικτύου έχει καθοριστικό ρόλο στην ομαλή λειτουργία του. Γενικά έχει παρατηρηθεί ότι σε ένα μοντέλο αρκετές από τις αποφάσεις που παίρνει καθώς λειτουργεί ,οι οποίες επηρεάζουν την ακρίβεια της πρόγνωσης του λαμβάνονται κατά την προεπεξεργασία των δεδομένων[12]. Αναλυτικότερα η προεπεξεργασία των δεδομένων εστιάζει σε τεχνικές οι οποίες εμπλουτίζουν την λεπτομέρεια του μοντέλου και αυξάνουν απόδοσή του. Η έννοια προεπεξεργασίας των δεδομένων που επηρεάζει το αποτέλεσμα ενός μοντέλου Machine learning είναι κοινά αποδεκτή. Παρόλα αυτά μια από τις ελάχιστες έρευνες που έχουν γίνει πάνω στο συγκεκριμένο ζήτημα είναι αυτή των(L.Blum et al.,1997). Η προεπεξεργασία δεδομένων έχει ένα μεγάλο εύρος τεχνικών που μπορούν να χρησιμοποιηθούν για διαφορετικές λειτουργίες όπως προεπεξεργασία κειμένου ,κειμένου από εικόνα,ψηφίων κ.α . Στην συγκεκριμένη διατριβή η τεχνική που θα χρησιμοποιηθεί είναι η προεπεξεργασία κειμένου διότι τα δεδομένα είναι σχόλια του twitter.

Στόχος της προεπεξεργασίας κειμένου είναι να παρουσιαστεί κάθε κομμάτι κειμένου ως διάνυσμα χαρακτηριστικών το οποίο σημαίνει να χωριστεί το κείμενο σε μεμονωμένες λέξεις. Μετά την ανάγνωση του κειμένου το επόμενο στάδιο προεπεξεργασίας διαιρεί το κείμενο σε στοιχεία τα οποία ονομάζονται (tokens,λέξεις και χαρακτηριστικά).Επίσης αναπαριστά το αρχικό κείμενο σε δεδομένα τα οποία είναι τα συγκεκριμένα στοιχεία καθώς και βάρη τους. Μια από τις αρκετές λειτουργίες του είναι τα stop-words, τα οποία είναι συνηθισμένες.

2.3 Machine Learning Algorithms

Η επιβλεπόμενη μάθηση είναι μια μέθοδος μηχανικής μάθησης όπου οι αλγόριθμοι μαθαίνουν από τα δεδομένα που δέχονται. Με αυτόν τον τρόπο σε περίπτωση που τους ανατεθούν παρόμοια δεδομένα στο μέλλον θα έχουν την ικανότητα να τα αξιολογήσουν και να τα προβλέψουν. Στόχος του supervised machine learning είναι ο αλγόριθμος να εκπαιδευτεί σωστά έτσι ώστε να μπορεί να προβλέπει ακριβείς στοιχεία για δεδομένα τα οποία δεν γνωρίζει.Οι πιο γνωστοί και συχνά χρησιμοποιημένοι αλγόριθμοι φαίνονται παρακάτω:

* *k-Nearest Neighbors:* Ο αλγόριθμος k-NN είναι σίγουρα ο ποιο απλός αλγόριθμος μηχανικής μάθησης.Το χτίσιμο του μοντέλου προορίζεται μόνο για την αποθήκευση των δεδομένων.Για να καταλήξει σε μια πρόβλεψη για ένα νέο σημείο δεδομένων , ο αλγόριθμος k-NN εντοπίζει τα πιο κοντινά σημεία δεδομένων στο σύνολο δεδομένων δηλαδή στους πλησιέστερους γείτονες του σημείου αυτού[13].Πιο συγκεκριμένα, στην πιο απλή εκδοχή του ο αλγόριθμος k-NN μελετάει μόνο έναν πλησιέστερο γείτονα ο οποίος είναι το πιο κοντινό σημείο δεδομένων στο σημείο που πρέπει να γίνει η συγκεκριμένη πρόβλεψη.Με λίγα λόγια δηλαδή αυτή η πρόβλεψη είναι η έξοδος για το σημείο αυτό.Όταν τα γειτονικά σημεία είναι περισσότερα από ένα τότε το μοντέλο γίνεται πιο απλό και αρχίζει να χάνει την ακρίβειά του.Παρόλα αυτά,ο αλγόριθμος μπορεί να διαχειριστεί διάφορα είδη δεδομένων,από αριθμητικά μέχρι και κατηγορικά κάτι που τον καθιστά κατάλληλη επιλογή για αρκετά είδη δεδομένων.
* *Linear Models:* Τα γραμμικά μοντέλα είναι μια κατηγορία μοντέλων η οποία πηγάζει από τον προηγούμενο αιώνα σαν ιδέα και έχει διατηρηθεί μέχρι και σήμερα.Τα μοντέλα αυτά κάνουν προβλέψεις χρησιμοποιώντας μια γραμμική συνάρτηση των χαρακτηριστικών εισόδου[14]. Αναλυτικότερα ,μπορούν να χαρακτηριστούν ως μοντέλα παλινδρόμησης των οποίων η πρόβλεψη είναι μια συγκεκριμένη γραμμή για ένα μόνο χαρακτηριστικό ,ένα επίπεδο όταν χρησιμοποιούνται δύο χαρακτηριστικά ,ή και ένα μεγαλύτερο επίπεδο όταν χρησιμοποιούνται πάνω από δύο χαρακτηριστικά.Ωστόσο, στην σημερινή εποχή η χρήση των γραμμικών μοντέλων δεν είναι πολύ χρήσιμη.Αυτό γίνεται διότι οι προβλέψεις των μοντέλων γίνονται σε ευθεία γραμμή κάτι το οποίο αποστασιοποιεί τις λεπτομέρειες των δεδομένων.
* Neural Networks: Τα νευρωνικά δίκτυα είναι μια ομάδα αλγορίθμων γνωστά και ως “deep learning” τα οποία χρησιμοποιούνται κυρίως για την δημιουργία machine learning εφαρμογών ή εργαλείων. Ειδικεύονται σε συγκεκριμένα προβλήματα και έχουν πολύ συγκεκριμένο σκοπό.Μια από τις κύριες μεθόδους είναι τα multilayer perceptrons (MLP’s) για ταξινόμηση και παλινδρόμηση τα οποία είναι η αρχή για πιο λεπτομερείς μεθόδους[15]. Αναλυτικότερα τα ΜLP’s μπορούνε να θεωρηθούν η βάση των νευρωνικών δικτύων καθώς βασίζονται σε μια αλληλουχία επιπέδων.Τα επίπεδα αυτά διαχωρίζονται σε φανερά και κρυφά όπου τα κρυφά και τα φανερά αποτελούνται από αρκετά επίπεδα. Η λειτουργία των multilayer perceptrons ξεκινά από το στρώμα εισόδου(Input layer) όπου γίνεται η εισαγωγή των δεδομένων είτε ως κείμενο είτε ως αρχείο το οποίο θα μετατραπεί σε πίνακα ,στην συνέχεια υπάρχει το κρυφό στρώμα(Hidden layer) το οποίο θεωρείται ο πυρήνας των MLP’s. Στο στρώμα αυτό γίνονται όλες οι κρίσιμες διαδικασίες του μοντέλου με τον ορισμό κάποιων παραμέτρων[16].Tο στρώμα εξόδου(Output layer) είναι υπεύθυνο για την πρόβλεψη που απαιτείται να γίνει για την επίλυση του προβλήματος για το οποίο χρησιμοποιήθηκε ο αλγόριθμος.Τέλος να επισημανθεί ότι υπάρχει περίπτωση το κρυφό στρώμα να είναι παραπάνω από ένα αντιθέτως με την παρακάτω εικόνα.



Εικόνα 2

Ένα άλλο είδος νευρωνικών δικτύων είναι τα Convolutional Neural Networks(CNN). Τα νευρωνικά αυτά δίκτυα είναι παρόμοια με τα τεχνητά νευρωνικά δίκτυα, το κοινό τους σημείο είναι ότι αποτελούνται από νευρώνες οι οποίοι βελτιώνονται από μόνοι τους μέσω της διαδικασίας της μάθησης.Ο κάθε νευρώνας λαμβάνει μια είσοδο και εκτελεί μια λειτουργία η οποία είναι χρήσιμη στην λειτουργία του μοντέλου. Η διαφορά των CNN σε σχέση με τα υπόλοιπα νευρωνικά δίκτυα είναι ότι χρησιμοποιούνται κυρίως για την αναγνώριση προτύπων σε εικόνες. Σε αυτή την περίπτωση δεν θα φανεί χρήσιμο στην παρούσα διατριβή.

2.4 Bert Model

To μοντέλο BERT και πιο αναλυτικά “Bidirectional Encoder Representations from Transformers” είναι ένα μοντέλο ταξινόμησης κειμένου το οποίο έχει σχεδιαστεί για να εκπαιδεύει αμφίδρομες αναπαραστάσεις μη επισημασμένου κειμένου καθώς εξετάζει και προς τις δύο κατευθύνσεις το κείμενο σε κάθε επίπεδο.Συνεπώς το προ-εκπαιδευμένο μοντέλο BERT μπορεί να βελτιστοποιηθεί με ένα εξίσου στρώμα εξόδου για την δημιουργία σύνθετων μοντέλων τα οποία καλύπτουν ένα εύρος εργασιών όπως απάντηση ερωτήσεων, δημιουργία κειμένου και προβλέψεις όσον αφορά το κείμενο. Σαν μοντέλο ταξινόμησης είναι σχετικά απλό και ισχυρό μοντέλο για ταξινόμηση κειμένου και επιτυγχάνει κορυφαία αποτελέσματα σε εφαρμογές επεξεργασίας φυσικής γλώσσας[16]. Το μοντέλο BERT χρησιμοποιεί προ-εκπαίδευση πριν την χρήση του για άλλα ζητήματα, δηλαδή από την στιγμή μου εισάγεται το BERT στα δεδομένα στα οποία θα χρησιμοποιηθεί θεωρείται προ εκπαιδευμένο και στην συνέχεια γίνεται περαιτέρω εκπαίδευση πάνω στα δεδομένα που απαιτούνται για επίλυση του προβλήματος. Η συγκεκριμένη προσέγγιση της προ-εκπαίδευσης έχει αποδειχθεί πολύ αποτελεσματική για την βελτίωση αρκετών γλωσσικών μοντέλων[17].Σε αυτές συμπεριλαμβάνονται εργασίες , όπως η διατύπωση συμπερασμάτων φυσικής γλώσσας [18] και η παράφραση [19], οι οποίες έχουν ως στόχο την πρόβλεψη των σχέσεων μεταξύ των προτάσεων αναλύοντάς τες σε μεγάλο βάθος.

Για την υλοποίηση και την εφαρμογή ενός μοντέλου BERT υπάρχουν δύο υπάρχουσες προσεγγίσεις: η βασισμένη σε χαρακτηριστικά και η τελειοποιημένη ρύθμιση[20].Η προσέγγιση βασισμένη σε χαρακτηριστικά χρησιμοποιεί συγκεκριμένες αρχιτεκτονικές οι οποίες περιλαμβάνουν προ-εκπαιδευμένες αναπαραστάσεις. Από την άλλη, η προσέγγιση της τελειοποιημένης ρύθμισης ,όπως ο γεννητικός προ-εκπαιδευμένος Transformer (OpenAI GPT)[21], εισάγει λίγες παραμέτρους αλλά εκπαιδεύεται σε downstream διεργασίες με τον απλό τρόπο της σωστής ρύθμισης όλων των προϋπαρχόντων προ-εκπαιδευμένων παραμέτρων. Αν και οι προσεγγίσεις είναι αρκετά διαφορετικές στον τρόπο που συντάσσονται , διαθέτουν την ίδια συνάρτηση κατά την προ-εκπαίδευση στην οποία χρησιμοποιούν γλωσσικά μοντέλα. Γενικότερα το μοντέλο BERT με την προ-εκπαίδευση που έχει στην ταξινόμηση κειμένου καταφέρνει να είναι αποδοτικό σε πολλούς τομείς της. Χρησιμοποιώντας μοντέλα τα οποία έχουν εκπαιδευτεί πριν την περαιτέρω χρήση τους σε νέα δεδομένα , περιορίζονται η ισχύς και οι δυνατότητες των μοντέλων. Παρόλα αυτά τα διαφορετικά μοντέλα και οι προσεγγίσεις που χρησιμοποιεί το BERT το κάνουν μια πολύ καλή επιλογή για ταξινόμηση κειμένου και πιο συγκεκριμένα πρόβλεψη πάνω στα δεδομένα που έχουν δοθεί.

Έκτος από μοντέλα το BERT προσφέρει και κωδικοποίηση(Tokenization) του κειμένου.Πιο αναλυτικά, για να έχει την ικανότητα το μοντέλο να εκπαιδευτεί σωστά με τα δεδομένα θα πρέπει τα δεδομένα να έχουν υποστεί την κατάλληλη προ επεξεργασία.Σε αυτή την περίπτωση χρησιμοποιούνται οι κωδικοποιητές(Τokenizers),οι οποίοι χωρίζουν το κείμενο σε μικρότερα μέρη.Αναλυτικότερα ,οι κωδικοποιητές έχουν δύο βασικές λειτουργίες,αρχικά διαχωρίζουν τα δεδομένα σε μικρότερα μέρη τα οποία ονομάζονται tokens και στην συνέχεια μετατρέπουν τα tokens σε αριθμούς 0 και 1 για να μπορεί στην συνέχεια το μοντέλο μηχανικής μάθησης να κατανοήσει.Πιο συγκεκριμένα o κωδικοποιητής του BERT χρησιμοποιεί αλγόριθμο κωδικοποίησης υπολέξεων (Subword Tokenization Algorithm).Ο αλγόριθμος αυτός διαχωρίζει το κείμενο το οποίο ήταν σε κανονική μορφή σε μικρότερες υπο-λέξεις και διαχειρίζεται τις σπάνιες και πιο ιδιαίτερες λέξεις χωρίς να αναγκάζεται να τις αγνοήσει.Το κείμενο το οποίο παίρνει ο κωδικοποιητής διαχωρίζεται σε μια ακολουθία από tokens όπου το κάθε ένα από αυτά παίρνει ένα ID.Τα tokens στην συνέχεια χρησιμοποιούνται για την περαιτέρω εκπαίδευση του μοντέλου στα νέα για αυτό δεδομένα.

Τα tokens διαχωρίζονται σε τρεις διαφορετικές κατηγορίες: το token ταξινόμησης ,το token διαχωρισμού και το padding token.Το token ταξινόμησης χρησιμοποιείται στην αρχή μιας ακολουθίας και αποτελεί το κύριο token για την υπόλοιπη ακολουθία. Στην συνέχεια το token διαχωρισμού (separation token) βρίσκεται στο τέλος κάθε πρότασης μιας ακολουθίας και χρησιμοποιείται για να διασφαλίσει ότι όλες οι ακολουθίες έχουν το ίδιο μέγεθος.Η συμβολή του είναι απαραίτητη διότι το BERT επεξεργάζεται την διαφορά μεταξύ των προτάσεων και καταλαβαίνει τα όρια της κάθε πρότασης.Τέλος ,το padding token βρίσκεται στο τέλος μιας ακολουθίας με σκοπό να αυξήσει το μέγεθος της ακολουθίας έτσι ώστε όλες οι ακολουθίες να είναι ίδιου μεγέθους.Αναλυτικότερα, τα padding tokens τοποθετούνται σε μικρότερες ακολουθίες για να αυξήσουν το μέγεθός τους έτσι ώστε να ταιριάζει το μέγεθός τους με το μέγεθος μεγαλύτερων ακολουθιών[22].Το BERT χρησιμοποιεί attention μηχανισμούς οι οποίοι μπορούν και ελέγχουν όλα τα tokens σε μια ακολουθία.Για να διασφαλιστεί ότι τα padding tokens δεν επηρεάζουν την απόδοση και το αποτέλεσμα του μοντέλου χρησιμοποιείται μια φόρμα (Mask).H συγκεκριμένη φόρμα ενημερώνει το μοντέλο για το ποια tokens είναι αληθινά δεδομένα και ποια tokens είναι padding.Μετά την διευκρίνηση αυτή το μοντέλο εστιάζει στα αληθινά δεδομένα και δεν περιλαμβάνει τα padding tokens στον υπολογισμό και στην εκπαίδευσή του.

Για την ολοκλήρωση του tokenization είναι αναγκαίο να χρησιμοποιηθούν οι εξής κωδικοποιήσεις.Το μοντέλο BERT είναι ικανό να λαμβάνει ids εισόδου για να είναι ικανό τα προχωρήσει στην εκπαίδευσή του στα νέα δεδομένα.Τα Ids αυτά αντιστοιχούν σε κάθε token μιας ακολουθίας τα οποία στην συνέχεια θα εισαχθούν στον βρόγχο εκπαίδευσης.Επίσης ο τομέας της κωδικοποίησης πρέπει να περιέχει attention mask για εξασφαλιστεί η παράλληλη και ταυτόχρονη επεξεργασία των διαφορετικών tokens διαφορετικών ακολουθιών οι οποίες διαφέρουν σε μέγεθος.Σημαντικό ρόλο στην κωδικοποίηση πριν την εκπαίδευση παίζουν και τα Token Type Ids.

Αρκετές εργασίες όπως η απάντηση ερωτήσεων και η εξαγωγή συμπερασμάτων φυσικής γλώσσας (Natural Language Inference) ,απαιτούν κατανόηση της σχέσης μεταξύ δύο προτάσεων[Jacob Devlin et al,2018]. Για να εκπαιδευτεί ένα μοντέλο το οποίο μπορεί να κατανοήσει τι σχέση έχουν δύο προτάσεις μεταξύ τους πρέπει να γίνει προ-εκπαίδευση για εργασία πρόβλεψης επόμενης πρότασης ( next-sentence prediction). Στην πρόβλεψη της επόμενης πρότασης βοηθούν τα Τoken type Ids τα οποία καθορίζουν ποια tokens της ακολουθίας αποτελούν την πρώτη πρόταση και ποία την δεύτερη.Αυτό γίνεται με τον ορισμό 0 στα ids της πρώτης πρότασης και 1 στα ids της δεύτερης πρότασης.Αντίθετα σε περίπτωση που η εργασία δεν είναι πρόβλεψη της επόμενης πρότασης όλα τα tokens μιας ακολουθίας κωδικοποιούνται με το 0,στην περίπτωση δηλαδή sequence classification τα Token type ids δεν επηρεάζουν το αποτέλεσμα.

2.4.1 Fine-Tuning Bert

Η προσέγγιση Fine-Tuning είναι απλοϊκή δεδομένου ότι ο μηχανισμός του Transformer επιτρέπει στο μοντέλο να πραγματοποιήσει αρκετές εργασίες ανεξαρτήτως το είδος και την διαμόρφωση του κειμένου.Το Bert χρησιμοποιεί μηχανισμό αυτόματης προσοχής για να ανακαλύψει τι είναι το κάθε κείμενο το οποίο εισάγεται για την εκπαίδευση του μοντέλου.Για την κάθε διαφορετική διεργασία συνδέονται οι αρμόδιες μέθοδοι εισόδου και εξόδου στο BERT και τελειοποιούνται όλες οι παράμετροι.Με λίγα λόγια δηλαδή η προσέγγιση Fine-Tuning είναι η τελειοποιημένη ρύθμιση των παραμέτρων που θα χρησιμοποιηθούν από την διαδικασία της εκπαίδευσης του μοντέλου.Οι παράμετροι που μπορούν να ρυθμιστούν είναι οι παράμετροι εισαγωγής των tokens στην διαδικασία της εκπαίδευσης καθώς και τα επίπεδα του μοντέλου Bert. Σε περίπτωση ανάλυσης κειμένου και πρόβλεψης της αρχικής πηγής του κειμένου απαιτούνται τα token ids καθώς και τα labeled data.

2.4.2 Feature-based

Σε αντίθεση με την προσέγγιση λεπτομερούς ρύθμισης η προσέγγιση με βάση τα χαρακτηριστικά έχει ορισμένα πλεονεκτήματα.Αρχικά, με την πεποίθηση ότι όλες οι εργασίες δεν μπορούν να παρουσιαστούν από την Transformer αρχιτεκτονική , έτσι είναι αναγκαίο να χρησιμοποιηθεί ένα μοντέλο βασισμένο στα χαρακτηριστικά και τις απαιτήσεις της διεργασίας.Επίσης , είναι ωφέλιμο να γίνει προ επεξεργασία των δεδομένων που χρησιμοποιηθούν έτσι ώστε να γίνει πιο σαφής και κατανοητή από το μοντέλο το οποίο θα λάβει τα δεδομένα αυτά για εκπαίδευση.Αν και οι προσεγγίσεις fine-tuning μέχρι σήμερα είναι πιο αποτελεσματικές ,οι προσεγγίσεις που δίνουν βάση στα χαρακτηριστικά μπορούν να εξελιχθούν επίσης σε μεγάλο βαθμό.

1. **Μεθοδολογία** 
   1. Library Imports

Στην συγκεκριμένη ενότητα θα παρουσιαστεί και θα περιγραφεί η μεθοδολογία που ακολουθήθηκε για την δημιουργία του project εντοπισμού μηχανικά παραγόμενου κειμένου.Πιο αναλυτικά η μεθοδολογία ακολουθεί μια προσέγγιση για την δημιουργία ενός μοντέλου BERT το οποίο θα εκτελεί ταξινόμηση κειμένου από tweets τα οποία έχουν εισαχθεί μέσω ενός database.

Για να γίνει εξήγηση του κώδικα που χρησιμοποιήθηκε πρέπει αρχικά να αναφερθούν οι εισαγωγές των βιβλιοθηκών που θα χρησιμοποιηθούν.Στο παρόν έχει εισαχθεί η βιβλιοθήκη “nltk”, μια βιβλιοθήκη η οποία χρησιμοποιείται κυρίως για επεξεργασία φυσικού κειμένου στην αγγλική γλώσσα και εισάγεται μόνο στην γλώσσα python. Στην συνέχεια εισάχθηκαν οι βιβλιοθήκες “pandas” η οποία χαρακτηρίζεται ως βιβλιοθήκη για διαχείριση δεδομένων καθώς και για εισαγωγή του αρχείου δεδομένων στον κώδικα, και η “numpy” η οποία λειτουργεί επίσης για διαχείριση δεδομένων και υπολογισμών πάνω σε έναν πίνακα δεδομένων.Επίσης με την χρήση μαθηματικών συναρτήσεων λειτουργεί πάνω σε εργασίες που έχουν δοθεί για την κανονικοποίηση των δεδομένων.Μια άλλη σημαντική βιβλιοθήκη για το παρόν project είναι η “re” η οποία παρέχει regular expressions για να βρίσκει τα διαφορετικά μοτίβα κειμένου και τα μοτίβα κειμένου που ταιριάζουν.Στην συνέχεια, εισάγεται η “sklearn” η οποία είναι υπεύθυνη για τον διαχωρισμό των δεδομένων σε κατανοητά μέρη έτσι ώστε να μπορεί να γίνει σωστή εκπαίδευση και σωστή δοκιμή του μοντέλου.Μια άλλη βιβλιοθήκη η οποία χρησιμοποιείται είναι η “transformers”,η οποία είναι μια βιβλιοθήκη ανοιχτού κώδικα που παρέχει μια ποικιλία από προ-εκπαιδευμένα μοντέλα τα οποία βασίζονται κυρίως στην επεξεργασία φυσικής γλώσσας (Natural Language Processing).Η τελευταία βιβλιοθήκη που χρησιμοποιείται στο εξής project είναι η “torch” ή πιο συγκεκριμένα “Pytorch”.Η βιβλιοθήκη αυτή χρησιμοποιείται για να προσφέρει επαρκείς πηγές για την δημιουργία deep neural networks ή και πιο συγκεκριμένα deep learning μοντέλα όπως είναι το ChatGPT της OpenAI.Επιπλέον για την χρήση πόρων από την κάρτα γραφικών ή τον επεξεργαστή απαιτείται η περαιτέρω εισαγωγή του “cuda” από την βιβλιοθήκη “torch”.O κώδικας που χρησιμοποιήθηκε φαίνεται παρακάτω :

import nltk

import pandas as pd

import numpy as np

import re

import seaborn as sns

import tensorflow as tf

import matplotlib.pyplot as plt

from nltk.corpus import stopwords

from sklearn.model\_selection import train\_test\_split

from torch.optim.lr\_scheduler import ReduceLROnPlateau

import torch

import torch.cuda

from torch import int64

import torch.nn as nn

from torch.utils.data import DataLoader,TensorDataset,random\_split

from transformers import BertTokenizer, BertForSequenceClassification, AdamW

from sklearn.metrics import accuracy\_score,classification\_report,precision\_score,recall\_score,f1\_score

* 1. Data Preparation & Preprocessing

Με την προϋπόθεση λειτουργίας του μοντέλου πρόβλεψης , είναι αναγκαίο να εισαχθούν και να επεξεργαστούν τα δεδομένα με σωστό τρόπο.Ο τρόπος με τον οποίο μπορεί να πραγματοποιηθεί αυτό αρχικά είναι η εισαγωγή του αρχείου “.csv” η οποία γίνεται με την μέθοδο read\_csv της βιβλιοθήκης pandas,παράλληλα ορίζεται η κωδικοποίηση και ο διαχωρισμός των στηλών του αρχείου csv με κόμματα.Στην συνέχεια δημιουργείται μια νέα στήλη με τιμή 0 σε κάθε γραμμή του αρχείου για να υπάρχει η δυνατότητα να ληφθούν σαν ετικέτες(labels) στον βρόγχο της επικύρωσης(Validation Loop).Έπειτα επιλέγονται μόνο οι δύο συγκεκριμένες στήλες οι οποίες χρειάζονται για να γίνει η ανάλυση και οι γραμμές οι οποίες περιείχαν τιμές που έλειπαν απορρίφθηκαν για να διατηρηθεί η συνοχή των δεδομένων.Μέσω της nltk βιβλιοθήκης κατέβηκαν κάποια “stopwords” τα οποία είναι λέξεις που δεν παίζουν σημαντικό ρόλο στην ανάλυση του κειμένου.Οι συγκεκριμένες λέξεις αποθηκεύτηκαν σε ένα σύνολο αντικειμένων το οποίο θα χρησιμοποιηθεί για την απομάκρυσή τους κατά την διάρκεια της προ-επεξεργασίας.

Στην συνέχεια ορίστηκε μια συνάρτηση προ-επεξεργασίας με όνομα “preprocess” η οποία έχει σκοπό να καθαρίσει και να ομαλοποιήσει το κείμενο .Η συγκεκριμένη συνάρτηση εκτελεί αρκετές διεργασίες οι οποίες είναι η μετατροπή όλων των λέξεων από κεφάλαια σε μικρά,η αφαίρεση μη αλφαβητικών γραμμάτων και των stopwords. Επίσης αφαιρεί ειδικούς χαρακτήρες όπως το δολάριο ($).Μετά την προ-επεξεργασία το καθαρό πλέον κείμενο τοποθετείται στον πίνακα στο ίδιο σημείο που ήταν και πριν, δηλαδή δεν τοποθετείται το κείμενο σε διαφορετικές σειρές.Ο παρακάτω κώδικας αντιστοιχεί στα στοιχεία που αναλύθηκαν παραπάνω:

tweets = pd.read\_csv("train.csv", encoding='latin', header=None, delimiter=",", quotechar='"')

print(tweets.head())

tweets['tweets']= tweets[1]

tweets['label']=tweets[2]

label\_mapping = {'bot': 1, 'human': 0}

tweets['generated'] = tweets['label'].map(label\_mapping)

print(tweets[['label','generated']].head())

tweets = tweets.replace([np.inf, -np.inf], np.nan).dropna(subset=['generated'])

tweets['generated'] = tweets['generated'].astype(int)

columns\_to\_select =['tweets','generated']

tweets\_new = tweets.loc[:,columns\_to\_select].copy()

tweets\_new.dropna(inplace=True)

nltk.download("stopwords")

stop\_words = set(stopwords.words("english"))

def preprocess(tweets\_new):

tweets\_new = re.sub(r'([a-z])([A-Z])',r'\1 \2',tweets\_new)

words = tweets\_new.split()

words = [word.lower() for word in words if word.isalpha()]

words = [word for word in words if word not in stop\_words]

tweets\_new=' '.join(words)

tweets\_new = re.sub(r'[^\w\s]','',tweets\_new)

tweets\_new = tweets\_new.replace('$','')

return tweets\_new

tweets\_new['preprocess'] = tweets\_new['tweets'].apply(preprocess)

sns.countplot(data= tweets\_new, x='generated')

plt.show()

print(tweets\_new[['preprocess','generated']])

* 1. Data Splitting & Model

Mε την προϋπόθεση ότι το preprocessing έχει πραγματοποιηθεί πάρθηκαν συγκεκριμένα δεδομένα του αρχείου,διότι το αρχείο ήταν αρκετά μεγάλο και διαχωρίστηκαν σε δεδομένα εκπαίδευσης (training),επικύρωσης(validation) και δοκιμής (testing).Ο διαχωρισμός ωφελεί την απόδοση του μοντέλου σε όχι γνωστά για αυτό δεδομένα και αποτρέπει το overfitting. O διαχωρισμός έγινε σε ποσοστό 80-20 για την εκπαίδευση-επικύρωση και για την εκπαιδευση-δοκιμή αντίστοιχα.Στην συνέχεια πραγματοποιήθηκε tokenizing στα δεδομένα κειμένου χρησιμοποιώντας τον tokenizer του BERT. Η διαδικασία αυτή είναι ένα σημαντικό σημείο στην προετοιμασία των δεδομένων για εισαγωγή στο μοντέλο καθώς μετατρέπει το κείμενο σε token ids τα οποία είναι αριθμητικές αναπαραστάσεις του κειμένου στο οποίο αντιστοιχούν.Ο tokenizer έτσι διασφαλίζει ότι όλες οι ακολουθίες που υπάρχουν στα δεδομένα είναι ίδιου μεγέθους και κατάλληλες για επεξεργασία από το deep learning μοντέλο. Μετά το tokenization τα δεδομένα και οι ετικέτες των δεδομένων μετατρέπονται σε tensors τα οποία φορτώνονται σε objets με ονομασία “DataLoader”.Τα object αυτά χειρίζονται με τέτοιο τρόπο τα δεδομένα έτσι ώστε κατά την διαδικασία της εκπαίδευσης το μοντέλο να καταλαβαίνει μοτίβα τα οποία θα το βοηθήσουν στην πρόβλεψη και όχι απλά να αποστηθίζει τα δεδομένα.

Αφού έγινε η παραπάνω διαδικασία τα δεδομένα είναι έτοιμα να εισαχθούν στο μοντέλο.Το μοντέλο που χρησιμοποιήθηκε είναι το BERT μοντέλο για ταξινόμηση ακολουθιών (Sequence Classification) ορίζοντας κάποια προ-εκπαιδευμένα βάρη από το “bert-base-uncased” μοντέλο.Το μοντέλο έπειτα θα ταξινομήσει το κείμενο σε δύο κατηγορίες δηλαδή θα πραγματοποιήσει δυαδική ταξινόμηση.Επειδή ο χρόνος παίζει σημαντικό ρόλο σε τέτοιας κατηγορίας εκπαίδευση ορίστηκε να χρησιμοποιηθούν οι πόροι της κάρτας γραφικών για να επιταχυνθεί η διαδικασία και οι υπολογισμοί.Ο βελτιστοποιητής (optimizer) που χρησιμοποιήθηκε είναι ο Adam ο οποίος είναι γνωστός για την αποδοτικότητα που έχει σε μοντέλα τύπου transformer. Τέλος τα epochs ορίστηκαν σε 10.

Η παραπάνω ανάλυση αντιστοιχεί στον εξής κώδικα:

sample\_size = 10000

sampled\_tweets = tweets\_new.sample(n=sample\_size, random\_state=42)

print(len(tweets\_new['preprocess']), len(tweets\_new['generated']))

X\_train\_val, X\_test, y\_train\_val, y\_test = train\_test\_split(sampled\_tweets['preprocess'], sampled\_tweets['generated'],test\_size=0.2, random\_state=42)

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_train\_val,y\_train\_val, train\_size=0.8,random\_state=42)

print(len(X\_train), len(y\_train))

print(len(X\_val), len(y\_val))

tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased', do\_lower\_case =True, padding = True,truncation = True, max\_length = 128)

encoded\_train = tokenizer(X\_train.tolist(),padding=True, truncation=True,return\_tensors='pt')

encoded\_val = tokenizer(X\_val.tolist(),padding=True,truncation=True,return\_tensors='pt' )

train\_label = torch.tensor(y\_train.values)

val\_label = torch.tensor(y\_val.values)

train\_datast = TensorDataset(encoded\_train['input\_ids'],encoded\_train['attention\_mask'],train\_label)

val\_datast = TensorDataset(encoded\_val['input\_ids'],encoded\_val['attention\_mask'],val\_label)

train\_loader = DataLoader(train\_datast,batch\_size=8,shuffle=True)

val\_loader = DataLoader(val\_datast,batch\_size=8,shuffle=False)

model = BertForSequenceClassification.from\_pretrained('bert-base-uncased', num\_labels=2)

device =torch.device("cuda"if torch.cuda.is\_available() else "cpu")

model.to(device)

optimizer = AdamW(model.parameters(), lr=2e-5,correct\_bias=False,weight\_decay=0.01)

scheduler = ReduceLROnPlateau(optimizer ,"min",patience=2)

epochs = 10

* 1. Τraining & Validation

O τομέας της εκπαίδευσης του μοντέλου είναι ο πιο βασικός για την δημιουργία ενός καλά δομημένου μοντέλου,για αυτό τον λόγο δεν πρέπει να έχει οποιαδήποτε λάθη στη δομή του.Εδώ ξεκινάει ο βρόγχος εκπαίδευσης ο οποίος επαναλαμβάνεται ανάλογα με τις εποχές που ορίστηκαν στο μοντέλο.Στην εκπαίδευση το μοντέλο μαθαίνει τις συσχετίσεις των προτάσεων του κειμένου και υπολογίζει την απώλεια του μοντέλου στα δεδομένα. Χρησιμοποιήθηκε gradient clipping για την αποτροπή exploding gradients τα οποία υπήρχε περίπτωση να καθυστερήσουν την εκπαίδευση του μοντέλου.Στο τέλος κάθε εποχής ,υπολογίστηκε ο μέσος όρος απώλειας της εκπαίδευσης και εκτυπώθηκε για την επίβλεψη της διαδικασίας της εκπαίδευσης.

Μετά την διαδικασία της εκπαίδευσης ,έγινε η αξιολόγηση του μοντέλου στο σύνολο επικύρωσης.Ο βρόγχος επικύρωσης εκτελεί το μοντέλο σε λειτουργία αξιολόγησης όπου απλά γίνεται επικύρωση και σύγκριση των αποτελεσμάτων χωρίς περαιτέρω ενέργειες όπως στον βρόγχο εκπαίδευσης.Οι προβλέψεις που έγιναν συγκρίθηκαν με τις πραγματικές ετικέτες για να υπολογιστεί η ακρίβεια του μοντέλου.Οι βρόγχος εκπαίδευσης είναι :

#Training loop Starting

from torch import int64

import torch.nn as nn

criterion = nn.CrossEntropyLoss()

train\_loss\_history = []

train\_accuracy\_history = []

val\_loss\_history = []

val\_accuracy\_history = []

best\_val\_accuracy = 0.0

best\_model\_state\_dict = None

for epoch in range(epochs):

model.train()

total\_loss = 0

total\_train\_loss = 0

correct\_train\_preds = 0

total\_train\_samples = 0

for batch in train\_loader:

input\_ids, attention\_mask,labels = batch

input\_ids, attention\_mask,labels =input\_ids.to(device), attention\_mask.to(device),labels.to(device,dtype=torch.int64)

optimizer.zero\_grad()

outputs = model(input\_ids,attention\_mask=attention\_mask)

logits = outputs.logits

loss = criterion(logits, labels)

total\_loss += loss.item()

loss.backward()

torch.nn.utils.clip\_grad\_norm\_(model.parameters(),1.0) #Gradient clipping to avoid exploding gradients

optimizer.step()

predictions = logits.argmax(dim=-1)

correct\_train\_preds += (predictions == labels).sum().item()

total\_train\_samples += labels.size(0)

avg\_train\_loss= total\_loss / len(train\_loader)

train\_accuracy = correct\_train\_preds / total\_train\_samples

train\_loss\_history.append(avg\_train\_loss)

train\_accuracy\_history.append(train\_accuracy)

print(f"Epoch {epoch + 1}/{epochs},Average Training Loss: {avg\_train\_loss:.2f},Training Accuracy:{train\_accuracy:.2f}")

Βρόγχος Επικύρωσης :

#Validation Loop

model.eval()

val\_preds = []

val\_labels = []

with torch.no\_grad():

for batch in val\_loader:

input\_ids, attention\_mask, labels = batch

input\_ids, attention\_mask, labels = input\_ids.to(device), attention\_mask.to(device), labels.to(device,dtype=torch.int64)

outputs = model(input\_ids, attention\_mask=attention\_mask)

logits = outputs.logits

val\_preds.extend(torch.argmax(logits,dim=1).cpu().numpy())

val\_labels.extend(labels.cpu().numpy())

val\_accuracy = accuracy\_score(val\_labels, val\_preds)

val\_precision = precision\_score(val\_labels, val\_preds)

val\_recall = recall\_score(val\_labels,val\_preds )

val\_f1 = f1\_score(val\_labels,val\_preds)

print(f"Validation Accuracy:{val\_accuracy:.2f}")

print(f"Validation Precision:{val\_precision:.2f}")

print(f"Validation Recall:{val\_recall:.2f}")

print(f"Validation F1-score:{val\_f1:.2f}")

* 1. Testing

To τελικό στάδιο του project είναι η δοκιμή του μοντέλου πάνω σε δεδομένα και η απαίτηση να εμφανίσει προβλέψεις σχετικά με το ποια tweets έχουν παραχθεί από άνθρωπο και ποια από τεχνητή νοημοσύνη.Οι προβλέψεις αυτές αφού δημιουργηθούν αποθηκεύονται σε ένα αρχείο csv με ονομασία “submission.csv”.To αρχείο αυτό έπειτα περιέχει τα αποτελέσματα του μοντέλου και είναι έτοιμο για περαιτέρω ανάλυση.

Συμπερασματικά, η μεθοδολογία αυτή προβάλει μια ολοκληρωμένη προσέγγιση στην αξιοποίηση του μοντέλου BERT για εργασίες ταξινόμησης κειμένου, εκπαίδευσης,επικύρωσης και πρόβλεψης. Κάθε τομέας του project δημιουργήθηκε έτσι ώστε να επιτευχθεί ο στόχος δημιουργίας ενός μοντέλου το οποίο με την συνεισφορά της προ επεξεργασίας να είναι ικανό να κάνει προβλέψεις πάνω στην ιδιοκτησία διάφορων tweets.O βρόγχος δοκιμής προβάλλεται παρακάτω :

import torch

import pandas as pd

from transformers import BertTokenizer, BertForSequenceClassification

device = torch.device("cuda" if torch.cuda.is\_available() else "cpu")

X\_test\_preprocessed = X\_test.astype(str)

batch\_size = 4

tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased', do\_lower\_case=True, padding=True, truncation=True, max\_length=128)

model = BertForSequenceClassification.from\_pretrained('bert-base-uncased').to(device)

model.eval()

predictions = []

with torch.no\_grad():

for i in range(0, len(X\_test\_preprocessed), batch\_size):

batch\_data = X\_test\_preprocessed.iloc[i:i+batch\_size].tolist()

tokenized\_batch = tokenizer(batch\_data, padding=True, truncation=True, return\_tensors='pt')

tokenized\_batch = {key: value.to(device) for key, value in tokenized\_batch.items()}

try:

outputs = model(\*\*tokenized\_batch)

logits = outputs.logits

batch\_predictions = torch.softmax(logits, dim=1)[:, 1].cpu().numpy()

predictions.extend(batch\_predictions)

except RuntimeError as e:

print(f"Error at batch {i}: {e}")

torch.cuda.empty\_cache()

submission = pd.DataFrame({'generated': predictions})

submission.to\_csv('submission.csv', index=False)

1. **Αποτελέσματα-Ευρήματα**

Στο παρόν κεφάλαιο θα παρουσιαστούν τα αποτελέσματα τα οποία το μοντέλο μηχανικής μάθησης έβγαλε σχετικά με το training loss,την απόδοση επικύρωσης ,την ακρίβεια επικύρωσης, την ανάκληση επικύρωσης και το f1-score επικύρωσης.Επίσης θα παρουσιαστούν τα errors τα οποία προέκυψαν κατά την διάρκεια της δημιουργίας του συγκεκριμένου project όπως και μια γενικότερη συζήτηση πάνω στο θέμα.

* 1. Dataset &Preprocessing

Πριν την ανάλυση των αποτελεσμάτων του μοντέλου είναι σημαντικό να αναλυθούν τα δεδομένα τα οποία χρησιμοποιήθηκαν και τα στάδια προ-επεξεργασίας τα οποία πραγματοποιήθηκαν.Το σύνολο των δεδομένων αποτελείται από tweets στα οποία πραγματοποιήθηκε προ-επεξεργασία για την καλύτερη συνοχή του συνόλου και την αφαίρεση του θορύβου των tweets.Η προ-επεξεργασία περιέχει βήματα όπως η απομάκρυνση των stopwords , ο διαχωρισμός των λέξεων και το lemmatization.Η διαδικασία lemmatization επαναφέρει μια λέξη η οποία είναι σε διαφορετικό χρόνο στην αρχική της ρίζα.Αυτό χρησιμοποιείται για καλύτερη κατανόηση των σχέσεων μεταξύ λέξεων και στην συνέχεια προτάσεων από το μοντέλο τεχνητής νοημοσύνης. Στην συνέχεια τα preprocessed πλέον δεδομένα διαχωρίζονται σε 80% για την εκπαίδευση και την επικύρωση και 20% για την δοκιμή πάνω σε δεδομένα στα οποία δεν έχει εκπαιδευτεί το μοντέλο.

4.2 Training & Results

To μοντέλο το οποίο εκπαιδεύτηκε ήταν το “bert-base-uncased” που υπάρχει στην βιβλιοθήκη Transformers και πιο συγκεκριμένα στο BERT (Bidirectional Encoder Representations from Transformers).H διαδικασία αυτή έγινε με την μέθοδο της ταξινόμησης ακολουθιών (Bert For Sequence Classification).Στην διαδικασία της εκπαίδευσης ορίστηκαν τα epochs σε 10 με batch size 8 για την αποφυγή σφαλμάτων μνήμης της κάρτας γραφικών και ρυθμό εκμάθησης (Learning rate) 2e-5.Ο AdamW optimizer χρησιμοποιήθηκε για να αλλάξει τιμές στα βάρη του μοντέλου και ήταν η παράμετρος που όρισε τον ρυθμό εκμάθησης που θα έχει το μοντέλο.

Ο βρόγχος εκπαίδευσης (Τraining loop) συμπεριέλαβε αρχιτεκτονική για τον ορισμό των logits τα οποία είναι raw scores που παράγει το μοντέλο για την περαιτέρω δημιουργία πιθανοτήτων.Επιπλέον χρησιμοποιήθηκε μια συνάρτηση gradient clipping για την αποφυγή exploding gradients. Οι παράμετροι της εκπαίδευσης του μοντέλου είναι οι εξής:

* Epochs : 10
* Batch Size : 8
* Learning Rate : 2e-5
* Optimizer :AdamW
* Loss Function : Cross-Entropy Loss

4.3Validation

Η διαδικασία της επικύρωσης είναι η διαδικασία υπολογισμού και καταγραφής των αποτελεσμάτων του μοντέλου τα οποία εμφανίζει στο τέλος σαν accuracy, precision, recall και f1-score,αναλόγως τι χρειάζεται το κάθε project.Eισήχθηκε μετά από κάθε epoch για να καταγράφει την απόδοση του μοντέλου στα δεδομένα.Στην συγκεκριμένη περίπτωση επειδή τα στοιχεία που απαιτούνται σχετικά με την απόδοση του μοντέλου είναι εξαιρετικά βασικός παράγοντας στην αξιολόγηση του, χρησιμοποιούνται τα metrics accuracy , precision, recall και F1-score.

4.4Αποτελέσματα

Παρακάτω περιγράφονται τα αποτελέσματα τα οποία δημιουργήθηκαν κατά την εκπαίδευση και την επικύρωση του μοντέλου.

Μέτα την εκπαίδευση του μοντέλου στα δεδομένα και την διαδικασία επικύρωσης τα εξής αποτελέσματα δημιουργήθηκαν:

|  |  |
| --- | --- |
| Training Loss | 0.02 |
| Validation Accuracy | 0.70 |
| Validation Precision | 0.72 |
| Validation Recall | 0.60 |
| Validation F1-Score | 0.66 |
| Training Accuracy | 0.99 |

Στο παραπάνω διάγραμμα φαίνεται ότι η απώλεια εκπαίδευσης είναι 0.02 δηλαδή το μοντέλο δεν έχει σοβαρή απώλεια κατά την εκπαίδευση.Επίσης το Accuracy 0.70 αναδεικνύει ότι η ικανότητα του μοντέλου να κάνει σωστές προβλέψεις είναι αρκετά καλή.Η τιμή του Precision η οποία είναι 0.72 είναι μια περαιτέρω έκδοση του Accuracy η οποία δηλώνει ότι το 72% των σωστών προβλέψεων που έκανε το μοντέλο ήταν όντως σωστές.Η τιμή του Recall η οποία ήταν 0.60 είναι μία καλή τιμή σχετικά με τις σωστές εκτιμήσεις που έκανε το μοντέλο.Το F1-Score με τιμή 0.66 είναι μια πολύ καλή τιμή για μια μέτρηση ενδιάμεσα στο Precision και το recall.To Τraining Accuracy είναι η ακρίβεια εκπαίδευσης των δεδομένων από το μοντέλο, η τιμή 0.99 σε αυτή την μέτρηση δηλώνει ότι το μοντέλο κατάφερε να μάθει πολύ καλά τα δεδομένα στα οποία εκπαιδεύτηκε.

4.4.1 Accuracy

Η ακρίβεια είναι σχετικά μια απλή μέτρηση που χρησιμοποιείται για την διάγνωση της ικανότητας του μοντέλου στο να πραγματοποιεί σωστές προβλέψεις.Η ακρίβεια της επικύρωσης με τιμή 0.70 δηλώνει ότι το 70% των τιμών της επικύρωσης έκανε σωστές προβλέψεις,δηλαδή το 70% των προβλέψεων ήταν σωστές.Σύμφωνα με την τιμή της ακρίβειας η συγκεκριμένη απόδοση θεωρείται αρκετά καλή,αυτό διασφαλίζει ότι το μοντέλο μπορεί να διακρίνει σε μεγάλο βαθμό μεταξύ κειμένου παραγόμενο από άνθρωπο και κειμένου παραγόμενο από μηχανή.Γενικότερα η τιμή της ακρίβειας θα πρέπει να είναι από 70% και πάνω για να θεωρείται ένα μοντέλο αποτελεσματικό.Στην περίπτωση που η ακρίβεια είναι κάτω από 70% είναι σίγουρο ότι απαιτείται επίλυση , η οποία μπορεί να πραγματοποιηθεί με πειραματικές δοκιμές και αλλαγές της προ επεξεργασίας , των hyper parameters και της δομής του μοντέλου.

Η ακρίβεια σε αρχικό στάδιο πριν την πιο βελτιωμένη ρύθμιση ήταν 0.49, κάτι τέτοιο αποδεικνύει ότι το μοντέλο δεν ήταν ικανό να προβλέψει σωστά ένα σημαντικό ποσοστό των δεδομένων.Αυτό θα είχε ως αποτέλεσμα την αποτυχία του μοντέλου στην δυαδική ταξινόμηση και θα χρειαζόταν να παρθούν σημαντικά μέτρα για την αύξηση της μέτρησης της ακρίβειας.Γνωρίζοντας αυτό διορθώθηκαν κάποια λάθη που υπήρχαν στα variables και ορίστηκε μια ακόμη παράμετρος για την κανονικοποίηση και την αύξηση του ποσοστού της ακρίβειας.Από την στιγμή που με τις αλλαγές που πραγματοποιήθηκαν στον κώδικα η ακρίβεια επικύρωσης αυξήθηκε στο 0.70 σημαίνει ότι έγινε σημαντικές αλλαγές στις παραμέτρους και στην δομή του μοντέλου.

4.4.2 Precision

To precision είναι μια περαιτέρω έκδοση της ακρίβειας η οποία αποδεικνύει αν οι απαντήσεις ενός μοντέλου είναι σωστές.Το precision με τιμή 0.72 σημαίνει ότι το 72% των προβλέψεων για τα κείμενα παραγόμενα από μηχανή είναι όντως παραγόμενα από μηχανή.Γενικότερα , το precision είναι μια πολύ καλή μέτρηση για διεργασίες δυαδικής ταξινόμησης όπως συμβαίνει και στην παρούσα έρευνα, κάτι που το κάνει απαραίτητο για την επίβλεψη της αποδοτικότητας ενός μοντέλου deep learning για προβλέψεις σε δεδομένα.

Ένα υψηλό σχετικά precision μπορεί να αποτελέσει σημαντικό παράγοντα στην ικανότητα του μοντέλου να κάνει σωστές προβλέψεις σε άγνωστα για αυτό δεδομένα.Για αυτό τον λόγο είναι μια καλή στρατηγική να αξιοποιηθούν τα αποτελέσματα του μοντέλου έτσι ώστε να επιτευχθεί αύξηση της μέτρησης precision.

Το precision πριν τις αλλαγές στον κώδικα ήταν 0.55 κάτι που σημαίνει ότι το 55% των προβλέψεων για την προέλευση των tweets (machine generated) ήταν σωστές.Με τις μετρήσεις αυτές αποδεικνύεται ότι οι προβλέψεις που πραγματοποιούνται είναι τυχαίες ,δηλαδή δεν υπάρχει κάποια βαρύτητα στις προβλέψεις του μοντέλου.Για τον λόγο αυτό πραγματοποιήθηκαν οι κατάλληλες προσαρμογές στον κώδικα για την αύξηση του precision σε 0.72 .

4.4.3 Recall

Η ανάκληση (Recall) είναι μια πιο συγκεκριμένη μέτρηση για την σωστή εκτίμηση της αποδοτικότητας του μοντέλου που χρησιμοποιήθηκε. Η τιμή 0.60 της ανάκλησης δηλώνει ότι το 60% των tweet τα οποία παράχθηκαν από τεχνητή νοημοσύνη εκτιμήθηκαν σωστά από το μοντέλο.Η τιμή πριν την προσαρμογή της βελτίωσης του κώδικα ήταν 0.50 το οποίο δείχνει ότι μόνο το 50% των προβλέψεων για tweet παραγόμενα από μηχανή ήταν σωστά.Κάτι τέτοιο θα αποτελούσε ανησυχίες για την λειτουργία και την απόδοση του μοντέλου σε νέα αθέατα δεδομένα.Η συγκεκριμένη μέτρηση είναι προφανώς χρήσιμη σε θέματα δυαδικής ταξινόμησης όπως στην συγκεκριμένη έρευνα καθώς δίνει μία “εικόνα” για την απόδοση του μοντέλου πάνω στα δεδομένα που δόθηκαν.Στην περίπτωση που το recall είναι από 50 και κάτω όπως ήταν στην αρχική εκπαίδευση και επικύρωση του μοντέλου ,υιοθετήθηκε το εξής πλάνο για την αύξησή του σε ένα επιθυμητό αποτέλεσμα το οποίο θα είναι σχετικά συμβατό με τις απαιτήσεις του τομέα που ανήκει η έρευνα:

* Έγινε έλεγχος για πιθανή ανισορροπία μεταξύ των label των κειμένων τα οποία έχουν δημιουργηθεί από ανθρωπινό παράγοντα και των κειμένων τα οποία έχουν δημιουργηθεί από τεχνητή νοημοσύνη.Σε περίπτωση που υπάρχει μικρή ή και μεγάλη ανισορροπία δεδομένων μεταξύ human και machine generated κειμένων τότε το recall όπως και τα υπόλοιπα είδη μέτρησης θα έχουν χαμηλό score.
* Έγινε έλεγχος για περαιτέρω χρήση διαφορετικών παραμέτρων οι οποίες ρυθμίζουν το μοντέλο και γενικότερα την απόδοσή του.Η προσθήκη νέων παραμέτρων και η ρύθμιση των υπαρχόντων αποδείχθηκε σωστή καθώς με τις ρυθμίσεις που έγιναν το accuracy και οι υπόλοιπες μετρήσεις παρουσίασαν σχετική αύξηση σε σχέση με τα αρχικά αποτελέσματα.

4.4.4 F-1 Score

Η μέτρηση F1-Score είναι εξίσου σημαντική για την αξιολόγηση της απόδοσης του μοντέλου ταξινόμησης.Θεωρείται ένας μέσος όρος του precision και του recall καθώς προβάλει μια αποτελεσματική τιμή η οποία αναδεικνύει την ικανότητα του μοντέλου στον εντοπισμό θετικών αποτελεσμάτων και στην ελάττωση αρνητικών αποτελεσμάτων.Η συγκεκριμένη υπο-ενότητα δίνει μια ολοκληρωμένη ανάλυση στην μέτρηση F1-Score για το μοντέλο ταξινόμησης και τις διαφορές που σημειώθηκαν με την ρύθμιση και την αλλαγή του κώδικα.

Σύμφωνα με την πρώτη προσπάθεια εκπαίδευσης του μοντέλου τα αποτελέσματα δεν ήταν σχετικά όπως ακριβώς θα έπρεπε να είναι ,καθώς υπήρχε overffiting στο μοντέλο και το αποτέλεσμα του F1-Score ήταν 0.52.Μετά τις απαραίτητες ρυθμίσεις και αλλαγές στις παραμέτρους η τιμή του F1-Score καθώς και των άλλων μετρήσεων αυξήθηκε.Πιο συγκεκριμένα η τιμή του F1-Score αυξήθηκε από 0.52 που ήταν στην αρχή σε 0.66,κάτι που σημαίνει ότι οι αλλαγές που πραγματοποιήθηκαν στον κώδικα είχαν θετική επίδραση στην αποδοτικότητα του μοντέλου.Το F1-Score με τιμή 0.66 δηλώνει ότι το μοντέλο έχει μια καλή ισορροπία ανάμεσα στις μετρήσεις Precision η οποία ήταν 0.72 και Recall η οποία ήταν 0.60 .

4.4.5 Compare Results

Η αξιολόγηση των αποτελεσμάτων ενός μοντέλου είναι σημαντικές σε μεγάλο βαθμό για την αποδοτικότητά του και τα πιθανά σημεία βελτίωσης.Στην συγκεκριμένη υπο-ενότητα θα παρουσιαστεί μια συγκριτική ανάλυση μεταξύ των δύο διαδικασιών εκπαίδευσης και επικύρωσης.Η πρώτη διαδικασία εκπαίδευσης και επικύρωσης έγινε σε ένα αρχικό σύνολο παραμέτρων παρέχοντας σχετικά μέτρια αποτελέσματα.Η δεύτερη διαδικασία εκπαίδευσης και επικύρωσης έγινε σε ένα ρυθμισμένο σύνολο αναβαθμισμένων παραμέτρων η οποία απέδωσε σε σημαντικές βελτιώσεις στην απόδοση του μοντέλου.

|  |  |
| --- | --- |
| Training Loss | 0.06 |
| Validation Accuracy | 0.50 |
| Validation Precision | 0.55 |
| Validation Recall | 0.50 |
| Validation F1-Score | 0.52 |
| Training Accuracy | 0.99 |

|  |  |
| --- | --- |
| Training Loss | 0.02 |
| Validation Accuracy | 0.70 |
| Validation Precision | 0.72 |
| Validation Recall | 0.60 |
| Validation F1-Score | 0.66 |
| Training Accuracy | 0.98 |

Στους παραπάνω πίνακες απεικονίζονται τα αποτελέσματα των μετρήσεων του μοντέλου στις δύο διαφορετικές διαδικασίες :

1. Στην πρώτη διαδικασία το training loss ήταν σχετικά μικρό δηλαδή η απώλεια που είχε το μοντέλο ήταν αρκετά μικρή (0.06).Επίσης φαίνεται ότι το Training Accuracy δηλαδή η ακρίβεια της εκπαίδευσης ήταν πολύ καλή (0.99).Το Validation Accuracy ήταν σχετικά χαμηλό σε σχέση με το Accuracy της εκπαίδευσης (0.50).Κάτι τέτοιο είχε ως αποτέλεσμα την αναγκαστική βελτίωση της απόδοσης του μοντέλου.Το precision και το recall ήταν (0.55) και (0.50) αντίστοιχα, κάτι που δείχνει ότι το μοντέλο ήταν μέτριο ως προς τις σωστές προβλέψεις που έκανε και έκανε λάθος ένα μεγάλο ποσοστό προβλέψεων.Τέλος η μέτρηση F1-Score ήταν αρκετά χαμηλή (0.52) λόγω βέβαια των χαμηλών Precision και Recall,κάτι το οποίο χρειάζεται βελτίωση για να είναι σχετικά λειτουργικό.
2. Στην δεύτερη διαδικασία το training loss ήταν πιο μικρό από την πρώτη δείχνοντας ότι η απώλεια του μοντέλου βελτιώθηκε (0.02).Το training accuracy δεν παρουσίασε μεγάλη αλλαγή καθώς και στην πρώτη διαδικασία ήταν πολύ καλό.Τo validation accuracy φαίνεται ότι βελτιώθηκε αρκετά (0.70) σε σχέση με την πρώτη διαδικασία,δηλαδή οι αλλαγές των παραμέτρων και η προσθήκη νέων βελτίωσαν την ακρίβεια του μοντέλου στις προβλέψεις.Επίσης με τις αλλαγές των παραμέτρων βελτιώθηκαν και το precision (0.72) με το recall (0.60) δείχνοντας έτσι μια καλύτερη ισορροπία μεταξύ παραγωγής σωστών προβλέψεων και ελάττωσης των λάθος προβλέψεων του μοντέλου.Η μέτρηση F1-Score έλαβε επίσης σημαντική αύξηση σε σχέση με την προηγούμενη διαδικασία από (0.52) σε (0.60), σίγουρο φαινόμενο καθώς προέκυψε και βελτίωση του precision και recall.

Η ρύθμιση των υπερπαραμέτρων και οι τεχνικές κανονικοποίησης έπαιξαν σημαντικό ρόλο στην βελτίωση των τιμών και της απόδοσης του μοντέλου.Η προσαρμογή των υπερπαραμέτρων και η προσθήκη νέων , όπως το batch size,το weight\_decay και το scheduler.Ο συντονισμός και ο προσεκτικός συνδυασμός των τριών αυτών παραμέτρων επέφερε σε καλύτερα αποτελέσματα κατά την δεύτερη διαδικασία.

H συγκριτική ανάλυση των επιδόσεων του μοντέλου σε δύο διαδικασίες εκπαίδευσης και επικύρωσης τονίζει τις σημαντικές βελτιώσεις που επιτεύχθηκαν μέσω αλλαγών στην δομή του μοντέλου ταξινόμησης.Η εξέλιξη από ένα αρχικό Accuracy (0.50) ,Precision (0.55),Recall (0.50) και F1-Score (0.52) σε Accuracy (0.70) ,Precision(0.72) ,Recall(0.60) και F1-Score (0.66) αποδεικνύουν την αποτελεσματικότητα των αλλαγών που πραγματοποιήθηκαν.

4.4.6 Errors

Τα errors τα οποία προέκυψαν κατά την δημιουργία του παρόν project ήταν:

* Label Values : Τα label values τα οποία χρειαζόταν για την διαδικασία της επικύρωσης δεν είχαν ίδιο αριθμό με τα κείμενα στα οποία θα γινόταν επικύρωση.Αυτό το πρόβλημα λύθηκε καθώς έγινε σωστός διαχωρισμός των στοιχείων του συνόλου δεδομένων για την διαδικασία της εκπαίδευσης , της επικύρωσης και της δοκιμής αντίστοιχα.
* Value Integer Not Accepted : Το συγκεκριμένο error συνέβη στην διαδικασία της εκπαίδευσης και της επικύρωσης καθώς το column των labels περιείχε έναν ακέραιο 0 ή 1 σε κάθε σειρά για όλο το σύνολο δεδομένων.Για να επιλυθεί το error αυτό προστέθηκε η συνάρτηση “dtype=torch.int64” μέσα στον βρόγχο της εκπαίδευσης και της επικύρωσης.
* Memory error :Το συγκεκριμένο error συνέβη στην διαδικασία της δοκιμής του μοντέλου σε αθέατα δεδομένα.Για να επιλυθεί το συγκεκριμένο ζήτημα προστέθηκε μια λειτουργία “empty\_cache()” η οποία αδειάζει χώρο για να υπάρχει μετά η δυνατότητα αποθήκευσης των προβλέψεων και να γίνει αντιγραφή αυτών στο αρχείο csv.

1. **Συμπεράσματα & Μελλοντικό Έργο**

5.1 Συμπεράσματα

Ο κύριος στόχος της διατριβής αυτής ήταν να αναπτυχθεί και να αξιολογηθεί ένα ισχυρό μοντέλο μηχανικής μάθησης, το οποίο θα ήταν ικανό να ξεχωρίζει μεταξύ κειμένου παραγόμενου από άνθρωπο και κειμένου παραγόμενο από τεχνητή νοημοσύνη.Το συγκεκριμένο πρόβλημα είναι πλέον πολύ έντονο και η συνεχής εξέλιξη των μοντέλων τεχνητής νοημοσύνη έχει προκαλέσει αρκετά προβλήματα με προϋπάρχοντα μοντέλα εντοπισμού.Η διαφορά του συγκεκριμένου μοντέλου είναι η συμβολή που θα έχει στον εντοπισμό tweets τα οποία έχουν παραχθεί από τεχνητή νοημοσύνη.

Αρχικά η απόδοση του μοντέλου ήταν μέτρια καθώς είχε Αccuracy (0.50), Precision (0.55), Recall (0.50) και F1-Score (0.52).Oι μετρήσεις αυτές έδειξαν ότι το μοντέλο έχει την ικανότητα να ξεχωρίσει αν ένα κείμενο είναι παραγόμενο από ανθρωπο ή από τεχνητή νοημοσύνη.Από την άλλη αντιμετώπιζε προβλήματα,καθώς οι σωστές προβλέψεις δεν είχαν μεγάλο ποσοστό επιτυχίας.

Μετά από μια σειρά αλλεπάλληλων βελτιώσεων οι οποίες συμπεριελάμβαναν τις ρυθμίσεις των παραμέτρων και της εφαρμογής τεχνικών κανονικοποίησης, σημειώθηκαν σημαντικές βελτιώσεις στην απόδοση του μοντέλου.Το τελικό αποτέλεσμα παρουσίασε εξαιρετικά υψηλότερο Accuracy το οποίο ήταν (0.70),Precision το οποίο ήταν(0.72),Recall το οποίο ήταν (0.60) και F1-Score το οποίο ήταν (0.66).Οι βελτιώσεις των μετρήσεων τονίζουν την εξαιρετική σημασία των επαναληπτικών διαδικασιών που πρέπει να ακολουθηθούν για την βελτίωση της αποδοτικότητας ενός μοντέλου .

5.2 Μελλοντικό Έργο

Αν και τα αποτελέσματα τα οποία προέκυψαν από την συγκεκριμένη έρευνα ήταν αρκετά καλά,υπάρχουν επιπλέον μέθοδοι για μελλοντικές έρευνες οι οποίες μπορούν να εξελίξουν το μοντέλο περαιτέρω πάνω στο συγκεκριμένο ζήτημα.

Προηγμένα Μοντέλα : Η αναζήτηση πιο προηγμένων μοντέλων ταξινόμησης κειμένου παίζει σημαντικό ρόλο στην βελτίωση των αποδόσεων του μοντέλου.Πιο συγκεκριμένα , μοντέλα όπως το GPT και διαφορετικές εκδοχές του μοντέλου BERT αναλόγως το επιθυμητό αποτέλεσμα μπορούν να χαρακτηριστούν πιο λειτουργικά σε συγκεκριμένες διεργασίες και να αποφέρουν ακόμη πιο βελτιωμένα αποτελέσματα.Η επιλογή και η δοκιμή διάφορων μοντέλων ταξινόμησης μπορεί να αποφέρει μεγαλύτερες και πιο αποτελεσματικές μετρήσεις.

Διαφορετικά Σύνολα δεδομένων : Η παρούσα έρευνα χρησιμοποιεί ένα σχετικά μέτριο σε μέγεθος σύνολο δεδομένων καθώς εξυπηρετούσε της ανάγκες της.Σαν μελλοντική έρευνα πάνω στο συγκεκριμένο θέμα θα πρέπει να εξεταστεί η εκδοχή χρήσης διαφορετικών συνόλων δεδομένων μεγαλύτερων σε μέγεθος , καθώς υπάρχει η περίπτωση αύξησης της απόδοσης του μοντέλου.Πιο συγκεκριμένα με την χρήση μεγαλύτερων συνόλων δεδομένων αυξάνεται η πιθανότητα βελτίωσης της απόδοσης του μοντέλου καθώς εκπαιδεύεται σε περισσότερα και πιο περίπλοκα δεδομένα με αποτέλεσμα να αυξηθεί η ικανότητα ταξινόμησης του.

Άντληση Δεδομένων από Twitter : Η συγκεκριμένη έρευνα χρησιμοποιεί ένα σύνολο δεδομένων το οποίο περιέχει tweets παραγόμενα από άνθρωπο και από τεχνητή νοημοσύνη.Σαν μελλοντική έρευνα πάνω στο συγκεκριμένο θέμα θα μπορούσε να γίνεται απευθείας άντληση δεδομένων από API’s του Twitter(X) και να αποθηκεύονται σε ένα αρχείο .csv . Έτσι θα γίνεται εκπαίδευση του μοντέλου σε νέα δεδομένα και θα μπορεί να οριστεί και ένα νέο αρχείο csv για την δοκιμή του μοντέλου.Με την δοκιμή του μοντέλου σε ένα νέο σύνολο δεδομένων θα πραγματοποιούνται διαφορετικές και μπορεί πιο συγκεκριμένες προβλέψεις.

Συμπερασματικά, η συγκεκριμένη διπλωματική παρουσίασε την δυνατότητα ενός μοντέλου μηχανικής μάθησης να διακρίνει μεταξύ ενός κειμένου το οποίο έχει παραχθεί από άνθρωπο και ενός κειμένου το οποίο έχει παραχθεί από τεχνητή νοημοσύνη.Οι μέθοδοι που ακολουθήθηκαν επέφεραν σημαντικές βελτιώσεις και εξελίξεις στο μοντέλο ταξινόμησης.Επιπρόσθετα, τα αποτελέσματα και οι στόχοι της διπλωματικής δίνουν την δυνατότητα για περαιτέρω μελλοντική έρευνα.Καθώς ο τομέας επεξεργασίας της φυσικής γλώσσας εξελίσσεται ραγδαία, η παρούσα διπλωματική θα μπορούσε να συνεισφέρει στο έργο εξάλειψης της παραπληροφόρησης και με μελλοντικές έρευνες θα έχει την ικανότητα να τελειοποιηθεί με σκοπό τον καλύτερο εντοπισμό κειμένων τα οποία παράγονται από τεχνητή νοημοσύνη.

**Βιβλιογραφία**