Επαλήθευση Συγγραφέα με χρήση προ-εκπαιδευμένων γλωσσικών μοντέλων

Authorship Verification using pre-trained Language Models

Από

Πετρόπουλος Παναγιώτης

Υποβάλλεται

για την εκπλήρωση των προϋποθέσεων λήψης

Μεταπτυχιακού Διπλώματος

στην «Τεχνητή Νοημοσύνη»

στο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Submitted

in partial fulfilment of the requirements for the degree of

Master of Artificial Intelligence

at the

UNIVERSITY OF PIRAEUS

May 2023

Συγγραφέας/Author Πετρόπουλος Παναγιώτης

ΔΠΜΣ «Τεχνητή Νοημοσύνη»/ II-MSc “Artificial Intelligence”

Μάιος 15, 2023

Έγινε αποδεκτό από/ Certified by.....

Ευστάθιος Σταματάτος /Efstathios Stamatatos
Καθηγητης /Professor
Επιπλέπων /Thesis Supervisor

Έγινε αποδεκτό από/ Certified by.....

Γεώργιος Βούρος /George Vouros
Καθηγητης /Professor
Μέλος Εξεταστικής Επιτροπής / Member of Examination Committee

Έγινε αποδεκτό από/ Certified by.....

Γεώργιος Πετάσης /George Petasis
Ερευνητής Β / Researcher B
Μέλος Εξεταστικής Επιτροπής / Member of Examination Committee

Επαλήθευση Συγγραφέα με χρήση προ-εκπαιδευμένων γλωσσικών μοντέλων

Authorship Verification using pre-trained Language Models

Από

Πετρόπουλος Παναγιώτης

Υποβλήθηκε στο ΔΠΜΣ «Τεχνητή Νοημοσύνη» την 15 Μαΐου 2023 ως υποχρέωση για την λήψη Μεταπτυχιακού Διπλώματος Σπουδών

Submitted to the II-MSc "Artificial Intelligence" on May 15, 2023, in partial fulfillment of the
requirements for the MSc degree

Abstract

In our everyday life, no one can dispute the necessity of artificial intelligence applications. These applications cover the largest to the smallest needs of modern humans. Knowledge, curiosity, security, and recognition are some of the basic human needs that people seek to satisfy through the internet. Social networking pages, chat, and blogs provide information and communication. There are also many incidents where we cannot verify the authenticity of a text as to its author. Authors who have left their mark in world literature can be easily recognizable. The difficult part is identifying the writer in the chaos of the worldwide web, and also for documents where the opinions of linguists and scientists diverge. It is easy to recognize texts from ancient or classical literature, but it is difficult to recognize in real-time the characteristics of an anonymous or forged writer. As the internet expands, the production of written language multiplies, and the field of artificial intelligence and an Author Verification system become increasingly necessary. It is no coincidence that many private or public enterprises and university units have integrated this field into their services.

Περίληψη

Στην καθημερινότητα μας, κανείς δεν μπορεί να αμφισβητήσει την αναγκαιότητα των εφαρμογών της τεχνητής νοημοσύνης. Οι εφαρμογές αυτές καλύπτουν από τις μεγαλύτερες ως τις μικρότερες ανάγκες του σύγχρονου ανθρώπου. Γνώση, περιέργεια, ασφάλεια, αναγνώριση είναι μερικές από τις βασικές ανθρώπινες ανάγκες που ζητάει να ικανοποιήσει ο άνθρωπος μέσα από το διαδίκτυο. Σελίδες κοινωνικής δικτύωσης, Chat, blogs παρέχουν πληροφόρηση και επικοινωνία. Πολλά είναι επίσης τα περιστατικά, όπου δεν μπορούμε να επικυρώσουμε την γνησιότητα κάποιου κειμένου ως προς τον συγγραφέα του. Οι συγγραφείς που έχουν αφήσει το στίγμα τους μέσα στην παγκόσμια βιβλιογραφία μπορεί να είναι εύκολα αναγνωρίσιμοι. Το δύσκολο είναι η αναγνώριση

του γράφοντος μέσα στο χάος του παγκόσμιου ιστού, αλλά και για έγγραφα όπου οι γνώμες των γλωσσολόγων και των επιστημόνων διίστανται. Είναι εύκολο να αναγνωριστούν κείμενα της παγκόσμιας αρχαίας ή κλασσικής λογοτεχνίας αλλά είναι δύσκολο να αναγνωριστούν σε αληθινό χρόνο τα χαρακτηριστικά ενός ανώνυμου ή ψευδεπίγραφου γράφοντος. Όσο το διαδίκτυο διευρύνεται, τόσο πολλαπλασιάζεται η παραγωγή γραπτού λόγου και τόσο πιο αναγκαίος καθίσταται ο τομέας της τεχνητής νοημοσύνης. Δεν είναι τυχαίο εξάλλου ότι, πολλές ιδιωτικές ή δημόσιες επιχειρήσεις και πανεπιστημιακές μονάδες έχουν εντάξει αυτόν τον τομέα στις υπηρεσίες που χρησιμοποιούν και προσφέρουν. Τα τελευταία χρόνια λοιπόν επιστήμονες που ασχολούνται με την τεχνητή νοημοσύνη, έχουν καταφέρει να δημιουργήσουν αυτοματοποιημένες εφαρμογές για την επαλήθευση συγγραφέων ενός ή πολλών κειμένων.

Επιβλέπων/Επιβλέπουσα: Σταματάτος Ευστάθιος
Ακαδημαϊκή Θέση: Καθηγητής

Ευχαριστίες/ Acknowledgments

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Σταματάτο για την πολύτιμη βοήθειά του και την άμεση ανταπόκρισή του σε κάθε πρόβλημα που προέκυπτε αλλά και για την υποστήριξή του και τα εύστοχα σχόλιά του στις στιγμές που παρέκκλινα από τον τελικό στόχο μου.

Το υλικό της Διπλωματικής αυτής εργασίας βασίζεται σε εργασία που πραγματοποιήθηκε και υποστηρίχθηκε από το Πανεπιστήμιο Πειραιώς και το Εθνικό κέντρο ερευνών και επιστημών Δημόκριτος στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών με τίτλο Μεταπτυχιακό στην Τεχνητή Νοημοσύνη. Οι απόψεις που εκφράζονται εδώ , τα ευρήματα και τα συμπεράσματα είναι αυτά του συγγραφέως και δεν εκφράζουν τις απόψεις του του Πανεπιστημίου Πειραιώς ή του Ινστ. Πληροφορικής και Τηλεπικοινωνιών του ΕΚΕΦΕ «Δημόκριτος».

Περιεχόμενα

Λίστα Εικόνων/ List of Figures

Λίστα Πινάκων/ List of Tables

1 Εισαγωγή

1.1 Επεξεργασία Φυσικής Γλώσσας – Natural Language Processing (NLP)

Το NLP αποτελεί ένα υπο-πεδίο της τεχνητής νοημοσύνης και πιο συγκεκριμένα χρησιμοποιείται στους τομείς της μηχανικής και βαθιάς μάθησης. Το NLP αναφέρεται στην εξόρυξη δεδομένων από κείμενα με χρήση τεχνικών text mining. Ο γενικότερος σκοπός είναι η ανάκτηση χρήσιμης πληροφορίας για το εκάστοτε μοντέλο με σκοπό την επίλυση κάποιου προβλήματος. Στις μέρες μας, οι προσεγγίσεις που υπάρχουν και προκύπτουν, δίνουν αρκετή σημασία στον τρόπο και το στυλ συγγραφής του κειμένου.

1.2 Ανάλυση Συγγραφέα – Authorship Analysis

Καθώς τα τελευταία χρόνια αυξάνεται συνεχώς η ανάγκη για απόκτηση πληροφοριών για κάποιον συγγραφέα ή κείμενο τόσο σε επίπεδο διαδικτύου, όσο και σε επίπεδο αρχαιολογικής επιστήμης, έχει προκύψει στον τομέα της μηχανικής και βαθιάς μάθησης το ερευνητικό πεδίο της ανάλυσης του κειμένου. Η ανάλυση των κειμένων είναι στατιστική μελέτη των γλωσσικών και υπολογιστικών χαρακτηριστικών των γραπτών εγγράφων των ατόμων. Το πρόβλημα του Authorship analysis μπορεί να κατηγοριοποιηθεί σε 4 βασικές υποκατηγορίες:

1. *Author Attribution*: Αναλύει και καθορίζει την πιθανότητα ενός συγκεκριμένου συγγραφέα να έχει γράψει ένα κομμάτι κειμένου, εξετάζοντας άλλα κείμενα που έχει συγγράψει ο ίδιος.
2. *Author Verification*: Αναλύει και καθορίζει την πιθανότητα 2 ή περισσότερα κείμενα να ανήκουν στον ίδιο συγγραφέα.
3. *Author Profiling*: Καθορίζει το προφίλ ή τα χαρακτηριστικά του δημιουργού ενός κειμένου. Αυτά τα χαρακτηριστικά περιλαμβάνουν φύλο και δημογραφικά στοιχεία, εκπαιδευτικό υπόβαθρο, προσωπικότητα, γλωσσική εξοικείωση κλπ.
4. *Similarity Detection*: Συγκρίνει πολλαπλά κομμάτια κειμένων και καθορίζει αν παράγονται ή όχι από έναν μόνο συγγραφέα χωρίς

απαραίτητα να προσδιορίζει τον συγγραφέα. Αυτή η κατηγορία του Authorship analysis χρησιμοποιείται κυρίως σε προβλήματα ανίχνευσης λογοκλοπής.

1.3 Επαλήθευση Συγγραφέα - Authorship Verification

Authorship Verification ορίζεται ως η εργασία (task) κατά την οποία ένα μοντέλο τεχνητής νοημοσύνης, αφού έχει εκπαιδευτεί είναι σε θέση να προσδιορίζει την πιθανότητα δύο ή περισσότερων κειμένων να ανήκουν στον ίδιο συγγραφέα. Αυτό μπορεί να επιτευχθεί με την ανάλυση και ανάκτηση χρήσιμων δεδομένων από κείμενα. Σύγχρονες εφαρμογές κατευθύνονται στην ανάκτηση του style συγγραφής για την επίλυση τους συγκεκριμένου προβλήματος.

Πολλές είναι επίσης οι εφαρμογές οι οποίες χρησιμοποιούν προ-εκπαιδευμένα γλωσσικά μοντέλα, τους λεγόμενους Transformers για την επίτευξη του στόχου. Η χρήση τέτοιων μοντέλων ξεκίνησε έπειτα από την δημοφιλή δημοσίευση με τίτλο «Attention is All you need» το 2017 [1], όπου εκεί «άνοιξε» το μονοπάτι για την δημιουργία των πρώτων Transformer. Το πρόβλημα της επαλήθευσης συγγραφέα αποτελεί ένα binary classification task, όπου ως False ορίζεται η πρόβλεψη ότι τα κείμενα δεν ανήκουν στον ίδιο συγγραφέα και ως True το αντίθετο.

1.4 Εφαρμογές του Authorship Verification

Η επαλήθευση συγγραφέα είναι μια τεχνική, η οποία έχει μια ευρεία γκάμα εφαρμογών σε διάφορους τομείς, συμπεριλαμβανομένων:

- Δικονομία (Forensics): Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για να προσδιορίσει το συγγραφέα ενός ύποπτου εγγράφου, όπως ενός απειλητικού γράμματος ή ενός απαγωγικού σημειώματος, πράγμα που μπορεί να βοηθήσει τις αρχές να αναγνωρίσουν και να συλλάβουν εγκληματίες.
- Δημοσιογραφία: Στη δημοσιογραφία, η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για να επαληθεύσει την αυθεντικότητα ενός άρθρου ειδήσεων ή για να ανιχνεύσει περιπτώσεις λογοκλοπής.

- Λογοτεχνία: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για να αναγνωρίσει τον πραγματικό συγγραφέα μιας λογοτεχνικής εργασίας, ιδιαίτερα σε περιπτώσεις όπου υπάρχει αμφισβήτηση ή αμφιβολία σχετικά με τη συγγραφή ενός συγκεκριμένου κειμένου.
- Γλωσσολογία: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για τη μελέτη των στυλιστικών και γλωσσικών χαρακτηριστικών ενός κειμένου και για την ανάλυση του πώς αυτά τα χαρακτηριστικά διαφέρουν ανάμεσα σε διαφορετικούς συγγραφείς και είδη γραφής.
- Ψηφιακή έρευνα: Στην εποχή της ψηφιακής τεχνολογίας, η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για την ανάλυση email, αναρτήσεων στα μέσα κοινωνικής δικτύωσης και άλλων μορφών ψηφιακής επικοινωνίας για να προσδιοριστεί ο συγγραφέας ενός συγκεκριμένου μηνύματος.
- Ιστορική έρευνα: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για να αναγνωρίσει τους συγγραφείς ιστορικών κειμένων, επιστολών και εγγράφων και για να φωτίσει σημαντικά ιστορικά γεγονότα και προσωπικότητες.
- Μάρκετινγκ: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί στον τομέα του μάρκετινγκ για να επαληθεύσει την αυθεντικότητα του περιεχομένου που δημοσιεύεται στο διαδίκτυο, όπως σε ιστοσελίδες, blogs και κοινωνικά δίκτυα. Αυτό μπορεί να βοηθήσει στην αναγνώριση και την πρόληψη της διασποράς ανεπιθύμητου περιεχομένου και ψευδών ειδήσεων.
- Δικαστικά: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί στο δικαστικό σύστημα για να προσδιοριστεί ο συγγραφέας ενός αμφισβητούμενου κειμένου, όπως μιας επιστολής ή μιας αναφοράς ως μια απόδειξη.
- Ιατρικός τομέας: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί στον ιατρικό τομέα για να προσδιορίσει τον συγγραφέα ενός ιατρικού κειμένου, όπως μιας ιατρικής έκθεσης ή μιας ιατρικής δημοσίευσης. Αυτό μπορεί να βοηθήσει στη διασφάλιση της ακρίβειας και της αξιοπιστίας των ιατρικών πληροφοριών.
- Πνευματική ιδιοκτησία: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί για να προστατεύσει τα πνευματικά δικαιώματα ενός

συγγραφέα ή μιας εταιρείας από παραβιάσεις, όπως παραποιήσεις ή αντιγραφή περιεχομένου χωρίς άδεια.

- Επιστημονική έρευνα: Η επαλήθευση συγγραφέα μπορεί να χρησιμοποιηθεί στην επιστημονική έρευνα για να προσδιοριστεί ο συγγραφέας ενός επιστημονικού κειμένου, όπως μιας διατριβής ή μιας δημοσίευσης σε επιστημονικό περιοδικό. Αυτό μπορεί να βοηθήσει στη διασφάλιση της ακρίβειας και της αξιοπιστίας της επιστημονικής έρευνας.

Συνολικά, η επαλήθευση συγγραφέα είναι ένα ισχυρό εργαλείο με ένα ευρύ φάσμα εφαρμογών, από την επιβολή του νόμου και την εγκληματολογία έως τη βιβλιογραφία και την ιστορική έρευνα.

1.5 Στόχος Εργασίας

Η επαλήθευση συγγραφέα (Authorship Verification) αποτελεί ένα σημαντικό και βασικό κομμάτι στο γενικότερο χώρο της ανάλυσης και αναγνώρισης συγγραφέα. Η παρούσα διπλωματική εργασία εστιάζει στην επίλυση του προβλήματος της επαλήθευσης συγγραφέα σε διάφορα είδη Dataset. Ιδιαίτερη έμφαση δόθηκε σε σώματα δεδομένων των παρακάτω κατηγοριών:

1. Μικρό Σώμα δεδομένων όπου το θέμα (topic) δεν είναι αυστηρό και διαφέρει ανάμεσα στα κείμενα (cross-topic).
2. Μεγάλο Σώμα δεδομένων όπου το θέμα (topic) αφορά συγκεκριμένα την επιστημονική φαντασία (σχόλια χρηστών από το www.fanfiction.net) και οι συγγραφείς του Test Dataset υπάρχουν στον Train Dataset (Close-Set).
3. Μεγάλο Σώμα δεδομένων όπου το θέμα (topic) αφορά συγκεκριμένα την επιστημονική φαντασία (σχόλια χρηστών από το www.fanfiction.net) και οι συγγραφείς του Test Dataset δεν υπάρχουν στον Train Dataset (Open-Set).
4. Μικρό Σώμα δεδομένων όπου το θέμα και το είδος κειμένων διαφέρει και οι συγγραφείς του Test Dataset δεν υπάρχουν στον Train Dataset. Πρόκειται επίσης για την επίλυση δια-τομεακής (Cross-Domain) επαλήθευσης συγγραφέα, όπου τα κείμενα έχουν διαφορετικά χαρακτηριστικά (π.χ. email vs essay ή text message vs business memo κλπ.). Αποτελεί ένα cross-DT Task (cross Discourse Types).

Δεδομένων των παραπάνω διαφορετικών Dataset έγινε χρήση του προ-εκπαιδευμένου μοντέλου BERT και παραλλαγής αυτού όπως το RoBERTa με σκοπό την εκτέλεση πειραμάτων, έτσι ώστε να ερευνηθεί κατά πόσο είναι εφικτό να επιτευχθεί ένα καλύτερο αποτέλεσμα από τις υπάρχουσες προσεγγίσεις σε αυτά τα σώματα δεδομένων. Οι Έρευνες και τα πειράματα κατευθύνθηκαν ως προς την χρήση του BERT και των Embeddings που παράγει αυτό το μοντέλο στο εκάστοτε Encode Layer.

2 Υπάρχουσες Προσεγγίσεις

2.1 Γενικότερες Προσεγγίσεις

Πολλές είναι οι προσεγγίσεις που έχουν γίνει για την επίλυση του Authorship Verification.

Άρθρα αναφέρουν [2] δύο βασικές κατηγορίες μοντέλων επαλήθευσης συγγραφέα: τα Intrinsic και τα Extrinsic μοντέλα. Τα Intrinsic μοντέλα παρέχουν μια απόφαση μόνο με βάση την ανάλυση των κειμένων σε ένα δεδομένο πρόβλημα επαλήθευσης, ενώ τα Extrinsic μοντέλα χρησιμοποιούν και εξωτερικά έγγραφα από άλλους συγγραφείς για να εκτιμήσουν την ομοιότητα των άγνωστων κειμένων με τα γνωστά κείμενα. Τα μοντέλα επαλήθευσης μπορούν επίσης να διαφοροποιηθούν ανάλογα με τον τύπο μάθησης που χρησιμοποιούν, με τα eager learning μοντέλα προσπαθώντας να εξάγουν ένα γενικό μοντέλο επαλήθευσης συγγραφέα με βάση το training set, και τα lazy learning μοντέλα που χειρίζονται κάθε περίπτωση επαλήθευσης ξεχωριστά.

Το 2017 επίσης πραγματοποιήθηκε μια προσέγγιση με βάση την μεθοδολογία των impostors [3]. Αυτή η μέθοδος χρησιμοποιεί ένα άλλο, εξωτερικό σώμα δεδομένων από άλλους συγγραφείς, μετατρέποντας το πρόβλημα από One-class σε Binary task. Αυτό το πετυχαίνει διαλέγοντας τυχαία κάθε φορά ένα set από συγγραφείς (impostors) με σκοπό να προβλέψει πότε ένα κείμενο ανήκει στον ίδιο ή όχι. Στην δημοσίευση του 2017 όμως αυτή η μέθοδος βελτιστοποιήθηκε και αντί για την τυχαία επιλογή των impostors έγινε χρήση της μέγιστης min-max similarity των κειμένων που τίθενται προς διερεύνηση για το αν ανήκουν στον

ίδιο συγγραφέα ή όχι. Τα πειράματα πραγματοποιήθηκαν στα σώματα δεδομένων του PAN 2014 και 2015. Παρακάτω τα αποτελέσματα.

	PAN14-DE	PAN14-DR	PAN14-EE	PAN14-EN	PAN14-GR	PAN14-SP	PAN15-DU	PAN15-EN	PAN15-GR	PAN15-SP
Khonji & Iraqi (2014)	0.913	**0.736**	0.590	0.750	0.889	**0.898**				
Gutierrez et al. (2015)							0.592	0.739	0.802	0.755
Original GI	0.947	0.660	0.618	0.649	0.772	0.604	0.667	**0.803**	0.656	0.785
Proposed-1	0.970	0.704	0.565	0.738	0.520	0.540	0.662	0.765	0.811	0.825
Proposed-2	0.901	0.698	0.655	0.634	0.860	0.772	0.595	0.786	0.742	0.802
Proposed-full	**0.976**	0.685	**0.762**	**0.767**	**0.929**	0.878	**0.709**	0.798	**0.844**	**0.851**

Πίνακας 1: Αποτελέσματα impostors method 2017.[3]

Φάνηκε λοιπόν ότι η βελτιωμένη μέθοδος των Impostors επέφερε πάρα πολύ καλά αποτελέσματα στα περισσότερα σώματα δεδομένων.

Μια από τις καλύτερες μεθόδους για την επίλυση του προβλήματος Authorship Verification είναι η εκμάθηση κατάλληλων αναπαραστάσεων. Μια τέτοια πολύ καλή προσέγγιση πραγματοποιήθηκε το 2017 [4] όπου μέσω της τεχνικής profile based (δηλ. συλλογή όλων των κειμένων ενός συγγραφέα) ορίστηκε ένας τρόπος εκμάθησης κατάλληλων αναπαραστάσεων που αντιστοιχούν στον εκάστοτε συγγραφέα, με την χρήση διαφορετικών τύπων χαρακτηριστικών όπως character-level n-grams, syntactic πληροφορία όπως Pos tags και επιλογή λέξεων που αφορούν το θέμα του κειμένου .

Άλλη προσέγγιση εστιάζει στην εκπαίδευση και το fine-tuning του προ-εκπαιδευμένου μοντέλου BERT σε Siamese αρχιτεκτονική με χρήση Contrastive Learning [5]. Συγκεκριμένα το σώμα δεδομένων που χρησιμοποιήθηκε κατά την εκπαίδευση είναι η μεγάλη εκδοχή του PAN 2020 και ως Test set χρησιμοποιήθηκε του 2021. Σε αυτήν την προσέγγιση έγινε χρήση του BERT για την εξαγωγή των Embeddings Vectors αφού είχαν γίνει re-sampling στα αρχικά ζευγάρια. Μετά το tokenization η κύρια μεθοδολογία διατηρούσε μόνο τα πρώτα 512 tokens από κάθε κείμενο. Τα αποτελέσματα στο Test set παρουσιάζονται παρακάτω:

Model	AUC	F1	c@1	F_0.5u	Brier	Overall
Final Model	0.8275	0.7911	0.7594	0.7257	0.8123	0.7832

Πίνακας 2: Αποτέλεσμα με χρήση BERT & Contrastive Learning PAN 2021.[5]

Επιπλέον, μια προσέγγιση που πραγματοποιήθηκε το 2014 εστιάζει στην συλλογή και συνένωση των κειμένων για κάθε συγγραφέα με σκοπό την συνολική είσοδο αυτών στο μοντέλο [6]. Χρησιμοποιώντας τον υπολογισμό αποστάσεων ανάμεσα σε Vectors που αφορούσαν τις συχνότητες εμφάνισης των n-grams στο κείμενο και προσδιορίζοντας ένα threshold για το πότε ένα ζευγάρι κειμένων είναι του ίδιου Author ή όχι επιτεύχθηκε ένα γενικό score της τάξεως του 0.845 (84.5%).

Τέλος, επειδή στις μέρες μας είναι απαραίτητη η δυνατότητα επεξήγησης του αποτελέσματος ενός Μοντέλου Τεχνητής Νοημοσύνης, έχουν δημοσιευθεί άρθρα, τα οποία προτείνουν μια εξηγήσιμη προσέγγιση για την επαλήθευση συγγραφέα χρησιμοποιώντας Contrastive Learning με τη βοήθεια του Attention μηχανισμού του εκάστοτε προ-εκπαιδευμένου γλωσσικού μοντέλου [7]. Μια από αυτές τις προσεγγίσεις χρησιμοποιεί Bi-LSTM για την εξαγωγή Embeddings των κειμένων αφού πρώτα έχουν υποστεί μια μικρή επεξεργασία ως προς τους περιέργους χαρακτήρες και τα URLS. Επίσης αφαιρέθηκαν τα σπάνια tokens για την αποφυγή της εκμάθησης του θέματος του κειμένου. Πριν την είσοδο του κειμένου και των χαρακτήρων από τα κείμενα στα στρώματα που αποτελούνται από Bi-LSTM προηγείται ένα CNN το οποίο αναλαμβάνει να «πιάσει» τα n-grams των tokens και των χαρακτήρων. Έπειτα με χρήση ενός Attention μηχανισμού το μοντέλο παράγει token Embeddings στα πρώτα επίπεδα, sentence Embeddings στα μεσαία, καταλήγοντας να εξάγει τα τελικά embeddings που αφορούν όλο το κείμενο της εισόδου. Αυτά τα Embeddings εισέρχονται σε μια μετρική ομοιότητας για την τελική απόφαση του μοντέλου. Για την δυνατότητα επεξήγησης της απόφασης του μοντέλου λαμβάνονται υπόψιν τα βάρη του Attention Μηχανισμού.

2.2 Διαγωνισμοί PAN at CLEF

Πρόκειται για διαγωνισμούς, όπου διάφορες επιστημονικές ομάδες από όλο τον κόσμο προσπαθούν να επιλύσουν ένα συγκεκριμένο Task στον τομέα του NLP[1].

[1] https://pan.webis.de/index.html

2.2.1 Authorship Verification Task PAN 2015

Το 2015 [8] στον συγκεκριμένο διαγωνισμό στο σώμα δεδομένων δεν υπήρχε κάποια αυστηρότητα ως προς το θέμα (topic), είδος (genre) και γλώσσα. Συγκεκριμένα στο σώμα δεδομένων μπορούσε κανείς να βρει:

1. Αγγλικά κείμενα: Cross-Topic Task
2. Ολλανδικά κείμενα: Cross-Genre Task
3. Ελληνικά κείμενα: Cross-Topic Task
4. Ισπανικά κείμενα: Cross-Genre Task

18 διαφορετικές υποβολές έγιναν στην πλατφόρμα αξιολόγησης του διαγωνισμού. Τα αποτελέσματα συγκεντρωτικά παρουσιάζονται στον παρακάτω πίνακα:

(a) Dutch

Team	FS	AUC	c@1	UP	Runtime
Moreau et al. [30]	**0.635**	**0.825**	**0.770**	0	08:09:35
Pacheco et al. [33]	0.624	0.822	0.759	30	00:05:08
Hürlimann et al. [14]	0.616	0.808	0.762	1	00:00:38
Maitra et al. [28]	0.518	0.759	0.683	4	02:32:48
Bartoli et al. [3]	0.518	0.751	0.689	1	00:07:01
Halvani [13]	0.455	0.709	0.642	8	00:00:09
Bagnall [2]	0.451	0.700	0.644	2	12:00:43
PAN15-ENSEMBLE	0.426	0.696	0.612	0	–
Gómez-Adorno et al. [10]	0.390	0.625	0.624	0	83:58:15
Sari and Stevenson [41]	0.381	0.613	0.621	4	00:02:04
Gutierrez et al. [11]	0.329	0.592	0.556	5	00:40:32
Vartapetiance and G. [49]	0.262	0.512	0.512	1	00:44:51
Pimas et al. [35]	0.262	0.508	0.515	0	00:02:27
PAN14-BASELINE-1	0.255	0.506	0.503	0	00:00:17
Castro-Castro et al. [5]	0.247	0.503	0.491	0	00:05:51
PAN13-BASELINE	0.242	0.506	0.479	0	00:00:47
Kocher and Savoy [21]	0.218	0.449	0.484	18	00:00:07
PAN14-BASELINE-2	0.191	0.422	0.452	16	00:02:10
Solórzano et al. [43]	0.153	0.397	0.385	4	00:10:25
Posadas-Durán et al. [36]	0.132	0.382	0.346	54	36:39:07
Nikolov et al. [31]	0.089	0.256	0.348	1	00:00:47
Mechti et al. [29]	0.000	0.500	0.000	165	–

(b) English

Team	FS	AUC	c@1	UP	Runtime
Bagnall [2]	**0.614**	**0.811**	**0.757**	3	21:44:03
Castro-Castro et al. [5]	0.520	0.750	0.694	0	02:07:20
Gutierrez et al. [11]	0.513	0.739	0.694	39	00:37:06
Kocher and Savoy [21]	0.508	0.738	0.689	94	00:00:24
PAN15-ENSEMBLE	0.468	0.786	0.596	0	–
Halvani [13]	0.458	0.762	0.601	25	00:00:21
Moreau et al. [30]	0.453	0.709	0.638	0	24:39:22
Pacheco et al. [33]	0.438	0.763	0.574	2	00:15:01
Hürlimann et al. [14]	0.412	0.648	0.636	5	00:01:46
PAN14-BASELINE-2	0.409	0.639	0.640	0	00:26:19
PAN13-BASELINE	0.404	0.654	0.618	0	00:02:44
Posadas-Durán et al. [36]	0.400	0.680	0.588	0	01:41:50
Maitra et al. [28]	0.347	0.602	0.577	10	15:19:13
Bartoli et al. [3]	0.323	0.578	0.559	3	00:20:33
Gómez-Adorno et al. [10]	0.281	0.530	0.530	0	07:36:58
Solórzano et al. [43]	0.259	0.517	0.500	0	00:29:48
Nikolov et al. [31]	0.258	0.493	0.524	16	00:01:36
Pimas et al. [35]	0.257	0.507	0.506	0	00:07:22
PAN14-BASELINE-1	0.249	0.537	0.464	159	00:01:11
Mechti et al. [29]	0.247	0.489	0.506	0	00:04:59
Sari and Stevenson [41]	0.201	0.401	0.500	0	00:05:47
Vartapetiance and G. [49]	0.000	0.500	0.000	500	–

(c) Greek

Team	FS	AUC	c@1	UP	Runtime
Bagnall [2]	**0.750**	0.882	**0.851**	5	10:07:49
Moreau et al. [30]	0.693	**0.887**	0.781	10	07:07:42
Kocher and Savoy [21]	0.631	0.822	0.768	20	00:00:11
Hürlimann et al. [14]	0.599	0.788	0.760	0	00:01:01
Gutierrez et al. [11]	0.581	0.802	0.725	5	00:28:32
PAN15-ENSEMBLE	0.537	0.779	0.690	0	–
Pacheco et al. [33]	0.517	0.773	0.670	3	00:02:01
Halvani [13]	0.493	0.767	0.643	9	00:00:17
Bartoli et al. [3]	0.458	0.698	0.657	1	00:07:45
Nikolov et al. [31]	0.454	0.709	0.640	0	00:01:01
PAN14-BASELINE-2	0.412	0.634	0.650	0	00:01:22
Castro-Castro et al. [5]	0.391	0.621	0.630	0	00:17:59
PAN13-BASELINE	0.384	0.641	0.600	0	00:01:46
Maitra et al. [28]	0.357	0.613	0.582	4	06:22:48
Gómez-Adorno et al. [10]	0.348	0.590	0.590	0	00:09:22
Solórzano et al. [43]	0.330	0.590	0.560	0	00:12:56
Pimas et al. [35]	0.230	0.480	0.480	0	00:03:58
Vartapetiance and G. [49]	0.212	0.460	0.460	0	00:36:30
PAN14-BASELINE-1	0.198	0.484	0.410	28	00:00:30
Mechti et al. [29]	0.000	0.500	0.000	100	–
Posadas-Durán et al. [36]	0.000	0.500	0.000	100	–
Sari and Stevenson [41]	0.000	0.500	0.000	100	–

(d) Spanish

Team	FS	AUC	c@1	UP	Runtime
Bartoli et al. [3]	**0.773**	**0.932**	**0.830**	0	00:09:16
Bagnall [2]	0.721	0.886	0.814	10	11:21:41
PAN15-ENSEMBLE	0.715	0.894	0.800	0	–
PAN14-BASELINE-2	0.683	0.823	0.830	0	00:04:03
Pacheco et al. [33]	0.663	0.908	0.730	0	00:04:23
Moreau et al. [30]	0.661	0.853	0.775	25	15:27:31
Hürlimann et al. [14]	0.539	0.739	0.730	0	00:01:29
Gutierrez et al. [11]	0.509	0.755	0.674	7	00:24:20
Sari and Stevenson [41]	0.485	0.724	0.670	0	00:03:48
Posadas-Durán et al. [36]	0.462	0.680	0.680	0	02:20:35
PAN14-BASELINE-1	0.443	0.692	0.640	0	00:00:45
Halvani [13]	0.441	0.704	0.627	23	00:00:14
PAN13-BASELINE	0.367	0.656	0.560	0	00:02:37
Kocher and Savoy [21]	0.366	0.650	0.564	20	00:00:22
Maitra et al. [28]	0.352	0.610	0.577	3	10:36:31
Vartapetiance and G. [49]	0.348	0.590	0.590	0	00:48:37
Castro-Castro et al. [5]	0.329	0.558	0.590	0	00:23:54
Gómez-Adorno et al. [10]	0.281	0.530	0.530	0	00:50:41
Pimas et al. [35]	0.240	0.490	0.490	0	00:04:12
Solórzano et al. [43]	0.218	0.454	0.480	0	00:11:18
Nikolov et al. [31]	0.095	0.280	0.340	0	00:01:09
Mechti et al. [29]	0.000	0.500	0.000	100	–

Πίνακας 3: Συγκεντρωτικά Αποτελέσματα PAN 2015.[8]

Η καλύτερη προσέγγιση στα Αγγλικά κείμενα είχε AUC Score 0.811 (81.1%) [9], χρησιμοποιώντας RNN όπου στην είσοδό του δεχόταν ακολουθίες από

χαρακτήρες και όχι ακολουθία από tokens. Τα κείμενα του κάθε συγγραφέα αντιμετωπίστηκαν ως διαφορετικά instances/samples (instance based προσέγγιση). Η ίδια μέθοδος επέφερε τα καλύτερα αποτελέσματα στον συγκεκριμένο διαγωνισμό και στα Ισπανικά (93.2) αλλά και στα Ελληνικά κείμενα (88.2%). Με αυτήν την προσέγγιση απέδειξε η συγκεκριμένη υποβολή ότι με ακολουθία χαρακτήρων ως είσοδο στο εκάστοτε μοντέλο μπορούν να επιτευχθούν πάρα πολύ καλά αποτελέσματα.

Στα Ολλανδικά κείμενα η καλύτερη προσέγγιση είχε AUC score 0.825 (82.5%) [10], όπου χρησιμοποιώντας SVM [11] και Decision trees και προσεγγίζοντας το πρόβλημα ως ένα regression κατάφερε να πετύχει το παραπάνω αποτέλεσμα. Αξίζει να σημειωθεί ότι η γενικότερη μεθοδολογία της συγκεκριμένης προσέγγισης ακολούθησε την λογική των Ensemble μοντέλων. Τέλος ως χαρακτηριστικά στην είσοδο του εκάστοτε μοντέλου χρησιμοποιήθηκαν POS tags για το κάθε sample του κάθε Authors (instance based προσέγγιση). Τέλος με την ίδια προσέγγιση η συγκεκριμένη υποβολή στον διαγωνισμό κατάφερε να πετύχει και το καλύτερο αποτέλεσμα ως προς το AUC score στα Ελληνικά κείμενα.

2.2.2 Authorship Verification Task PAN 2020

Το 2020 [12] στον συγκεκριμένο διαγωνισμό στο σώμα δεδομένων είχε ένα συγκεκριμένο θέμα (topic) και τα κείμενα περιείχαν ίδια χαρακτηριστικά. Υπήρχαν 2 σώματα δεδομένων ένα μεγάλο και ένα μικρό. Ήταν στην επιλογή του εκάστοτε διαγωνιζόμενου για το ποιο θα χρησιμοποιήσει. Πρόκειται για κείμενα από σχόλια χρηστών για ταινίες επιστημονικής φαντασίας όπου στο Test σώμα δεδομένων εμπεριέχονται ίδιοι συγγραφείς με αυτούς του Train (Close-Set).

13 υποβολές έγινα στον διαγωνισμό για το συγκεκριμένο task και τα συγκεντρωτικά αποτελέσματα παρουσιάζονται στον παρακάτω πίνακα.

Submission	AUC	c@1	F0.5u	F1	Overall
boenninghoff20-large	**0.969**	**0.928**	**0.907**	**0.936**	**0.935**
weerasinghe20-large	0.953	0.880	0.882	0.891	0.902
boenninghoff20-small	0.940	0.889	0.853	0.906	0.897
weerasinghe20-small	0.939	0.833	0.817	0.860	0.862
halvani20-small	0.878	0.796	0.819	0.807	0.825
kipnis20-small	0.866	0.801	0.815	0.809	0.823
araujo20-small	0.874	0.770	0.762	0.811	0.804
niven20-small	0.795	0.786	0.842	0.778	0.800
gagala20-small	0.786	0.786	0.809	0.800	0.796
araujo20-large	0.859	0.751	0.745	0.800	0.789
baseline (naive)	0.780	0.723	0.716	0.767	0.747
baseline (compression)	0.778	0.719	0.703	0.770	0.742
ordonez20-large	0.696	0.640	0.655	0.748	0.685
ikae20-small	0.840	0.544	0.704	0.598	0.672
faber20-small	0.293	0.331	0.314	0.262	0.300

Πίνακας 4: Συγκεντρωτικά Αποτελέσματα PAN 2020.[12]

Η καλύτερη προσέγγιση [13] πέτυχε γενικό score (Overall) 0.935 (93.5%) χρησιμοποιώντας την μεγάλη έκδοση του σώματος δεδομένων (Large). Στην συγκεκριμένη προσέγγιση έχουμε δύο βασικά βήματα κατά την εκπαίδευση:

1. Χρήση των LSTM νευρωνικών δικτύων σε μια Siamese αρχιτεκτονική με σκοπό την εξαγωγή Embeddings Vectors που αφορούν linguistic χαρακτηριστικά για τα Same Author και Different Author Pairs μέσω της διαδικασίας του Contrastive Learning υπολογίζοντας σε κάθε εποχή την Ευκλείδεια απόσταση των διανυσμάτων στον χώρο.
2. Deep Bayes Factor Scoring μοντέλο για τον προσδιορισμό των πιθανοτήτων και την επίλυση του binary Classification προβλήματος.

Για να αποφευχθεί το γεγονός ότι το μοντέλο μπορεί να μάθει χαρακτηριστικά που συσχετίζονται με το θέμα του κειμένου, η συγκεκριμένη μεθοδολογία ακολούθησε τρία βασικά βήματα:

1. Αντικατάσταση όλων των σπάνιων token και χαρακτήρων με κάποιο συγκεκριμένο αλφαριθμητικό. Αυτό διότι έχει αποδειχθεί ότι οι σπάνιες λέξεις και tokens συνήθως αφορούν το θέμα του κειμένου.
2. Προσθήκη της πληροφορίας για το θέμα του εκάστοτε κειμένου την οποία δίνει το σώμα δεδομένων ως prefix στα κείμενα.
3. Αναδιαμόρφωση των ζευγαριών από την αρχή με βάση τους συγγραφείς. Δεν διατηρήθηκαν δηλαδή τα αρχικά ζευγάρια (re-sampling).

2.2.3 Authorship Verification Task PAN 2021

Το 2021 [14] οι διαγωνιζόμενοι εκπαιδεύοντας τα μοντέλα τους με το αντίστοιχο σώμα δεδομένων του 2020 κλήθηκαν να κάνουν προβλέψεις σε κάποιο Test set στο οποίο οι συγγραφείς των κειμένων αυτού δεν υπήρχαν στο Train set (Open Set).

13 υποβολές έγινα στον διαγωνισμό για το συγκεκριμένο task και τα συγκεντρωτικά αποτελέσματα παρουσιάζονται στον παρακάτω πίνακα.

Team	Dataset	AUC	c@1	F_1	$F_{0.5u}$	Brier	Overall
boenninghoff21	large	**0.9869**	**0.9502**	**0.9524**	**0.9378**	**0.9452**	**0.9545**
embarcaderoruiz21	large	0.9697	0.9306	0.9342	0.9147	0.9305	0.9359
weerasinghe21	large	0.9719	0.9172	0.9159	0.9245	0.9340	0.9327
weerasinghe21	small	0.9666	0.9103	0.9071	0.9270	0.9290	0.9280
menta21	large	0.9635	0.9024	0.8990	0.9186	0.9155	0.9198
peng21	small	0.9172	0.9172	0.9167	0.9200	0.9172	0.9177
embarcaderoruiz21	small	0.9470	0.8982	0.9040	0.8785	0.9072	0.9070
menta21	small	0.9385	0.8662	0.8620	0.8787	0.8762	0.8843
rabinovits21	small	0.8129	0.8129	0.8094	0.8186	0.8129	0.8133
ikae21	small	0.9041	0.7586	0.8145	0.7233	0.8247	0.8050
unmasking21	small	0.8298	0.7707	0.7803	0.7466	0.7904	0.7836
tyo21	large	0.8275	0.7594	0.7911	0.7257	0.8123	0.7832
naive21	small	0.7956	0.7320	0.7856	0.6998	0.7867	0.7600
compressor21	small	0.7896	0.7282	0.7609	0.7027	0.8094	0.7581
futrzynski21	large	0.7982	0.6632	0.8324	0.6682	0.7957	0.7516
liaozhihao21	small	0.4962	0.4962	0.0067	0.0161	0.4962	0.3023

Πίνακας 5: Συγκεντρωτικά Αποτελέσματα PAN 2021[14]

Την καλύτερη προσέγγιση όπως και το 2020 την είχε πάλι η ίδια ομάδα [15] με γενικό score (overall) 0.9545 (95.45%).

Σε αυτήν την προσέγγιση η ομάδα επέκτεινε την μεθοδολογία του 2020 και πρόσθεσε ένα επιπλέον βήμα για την ανίχνευση των ζευγαριών που δεν απαντήθηκαν από το μοντέλο (out-of-distribution – O2D2 detector). Επομένως στο τέλος η αρχιτεκτονική και η τελική προσέγγιση αποτελούσε μια Ensemble μεθοδολογία όπου τα επιμέρους μοντέλα ήταν το μοντέλο του 2020 και το μοντέλο O2D2.

2.2.4 Authorship Verification Task PAN 2022

Το 2022 [16] για ένα μικρό σώμα δεδομένων όπου περιείχε διαφορετικά είδη κειμένων (essays,emails,text message & business memos) χωρίς να μιλούν για κάποιο συγκεκριμένο θέμα και οι συγγραφείς του Test set να μην υπάρχουν στο Train αρκετές ομάδες προσπάθησαν να επιλύσουν το Authorship Verification Task.

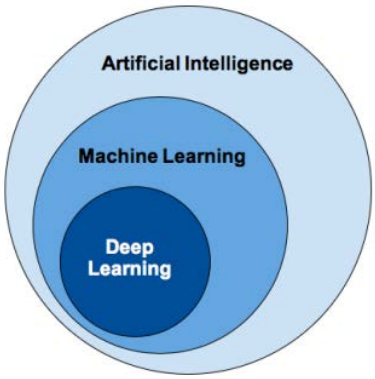
7 υποβολές έγιναν στην πλατφόρμα αξιολόγησης του διαγωνισμού και παρουσιάζονται παρακάτω τα αποτελέσματα συγκεντρωτικά.

Participant	AUROC	c@1	F_1	$F_{0.5u}$	Brier	Overall
BASELINE-CNGDIST22	0.546	0.496	0.669	0.542	0.749	**0.600**
NAJAFI22	**0.598**	**0.571**	0.576	**0.571**	0.618	0.587
GALICIA22	0.512	0.499	0.628	0.544	0.741	0.585
JINLI22	0.577	0.557	0.581	0.563	0.589	0.573
BASELINE-COMPRESSOR22	0.541	0.493	0.570	0.478	**0.750**	0.566
LEI22	0.539	0.539	0.399	0.488	0.539	0.501
YIHUIYE22	0.542	0.526	0.398	0.461	0.565	0.499
HUANG22	0.519	0.519	0.196	0.328	0.519	0.416
CRESPOSANCHEZ22	0.500	0.500	0	0	0.748	0.350

Πίνακας 6: Συγκεντρωτικά αποτελέσματα PAN 2022.[16]

Η Καλύτερη προσέγγιση πέτυχε 0.598 (59.8%) AUC score [17] πράγμα που δείχνει την δυσκολία του συγκεκριμένου Dataset. Σε αυτή την προσέγγιση χρησιμοποιήθηκε το προ-εκπαιδευμένο γλωσσικό μοντέλο T5 [18] για την εξαγωγή των Embeddings από τα κείμενα χρησιμοποιώντας επιπλέον χαρακτηριστικά και πληροφορία για να βοηθηθεί το μοντέλο. Αυτές οι πληροφορίες είναι n-grams και pos tags. Για την παραγωγή των n-grams εφαρμόστηκε μια αρχιτεκτονική με CNN νευρωνικά δίκτυα. Επίσης δεν αφαιρέθηκε τίποτα από τα κείμενα όπως τα emoticons καθώς χρησιμοποιήθηκαν ως επιπλέον χαρακτηριστικά. Τέλος με την βοήθεια του Attention μηχανισμού του μοντέλου T5 προσπάθησαν να εξάγουν και πληροφορία για το κάθε token. Όλες αυτές οι πληροφορίες και ο συνδυασμός αυτών κατάφερε να επιφέρει το καλύτερο αποτέλεσμα στον διαγωνισμό, το οποίο όμως είναι αρκετά χαμηλό, αποδεικνύοντας την δυσκολία του συγκεκριμένου σώματος δεδομένων.

3 Μηχανική και Βαθιά Μάθηση



3.1 Μηχανική Μάθηση – Machine Learning

Η Μηχανική Μάθηση είναι μια υποκατηγορία της τεχνητής νοημοσύνης που ασχολείται με τον σχεδιασμό, την ανάπτυξη και την εφαρμογή αλγορίθμων που επιτρέπουν στις μηχανές να "μαθαίνουν" από δεδομένα, χωρίς να χρειάζεται να προγραμματιστούν ρητά.

Η διαδικασία της Μηχανικής Μάθησης ξεκινά με τη συλλογή δεδομένων, τα οποία χρησιμοποιούνται στη συνέχεια για να εκπαιδεύσουν ένα μοντέλο. Έπειτα το μοντέλο αυτό χρησιμοποιείται για να κάνει προβλέψεις ή να λαμβάνει αποφάσεις βάσει νέων δεδομένων.

Υπάρχουν δύο βασικά είδη Μηχανικής Μάθησης:

1. Supervised Learning (Επιβλεπόμενη Μάθηση): Στην επιβλεπόμενη μάθηση, το μοντέλο εκπαιδεύεται σε ένα σύνολο δεδομένων που περιλαμβάνει ετικέτες (labels) για κάθε παράδειγμα. Το μοντέλο μαθαίνει να προβλέπει τις ετικέτες για νέα δεδομένα. Για παράδειγμα, ένα μοντέλο επιβλεπόμενης μάθησης θα μπορούσε να εκπαιδευτεί να αναγνωρίζει εικόνες που περιέχουν σκύλους και γάτες.

2. Unsupervised Learning (Μη Επιβλεπόμενη Μάθηση): Στη μη επιβλεπόμενη μάθηση, το μοντέλο εκπαιδεύεται σε ένα σύνολο δεδομένων που δεν έχει ετικέτες. Το μοντέλο αναζητά μοτίβα ή δομές στα δεδομένα.

Για την επίτευξη της μάθησης, τα μοντέλα χρησιμοποιούν διάφορους αλγορίθμους. Αυτοί οι αλγόριθμοι μπορούν να αναπαραστήσουν τα δεδομένα σε διάφορες μορφές, όπως διανύσματα ή γραφήματα, και να βελτιστοποιήσουν τις παραμέτρους του μοντέλου για να μεγιστοποιήσουν την ακρίβεια των προβλέψεων.

Η Μηχανική Μάθηση εφαρμόζεται σε πολλούς τομείς, όπως η αναγνώριση προτύπων, οι αυτόματες μεταφράσεις, ο έλεγχος της ποιότητας των προϊόντων, η αναγνώριση φωνής και εικόνας.

Στη διαδικασία της εκπαίδευσης, τα μοντέλα Μηχανικής Μάθησης χρησιμοποιούν τα δεδομένα εκπαίδευσης για να βελτιώσουν την ακρίβεια των προβλέψεων. Κατά τη διάρκεια της εκπαίδευσης, τα μοντέλα μπορούν να υποστούν υπερπαραμετροποίηση, η οποία αφορά την επιλογή των παραμέτρων του μοντέλου που επηρεάζουν την ακρίβεια των προβλέψεων.

Όταν το μοντέλο εκπαιδεύεται στο σύνολο των δεδομένων εκπαίδευσης, μπορεί να ελεγχθεί στο σύνολο δεδομένων ελέγχου (Validation Set), για να διαπιστωθεί αν το μοντέλο είναι σε θέση να γενικεύσει σε νέα δεδομένα και να δώσει ακριβείς προβλέψεις.

Συμπερασματικά, η Μηχανική Μάθηση είναι ένα ευρέως χρησιμοποιούμενο εργαλείο για την ανάλυση δεδομένων και τη δημιουργία προβλέψεων σε διάφορους τομείς.

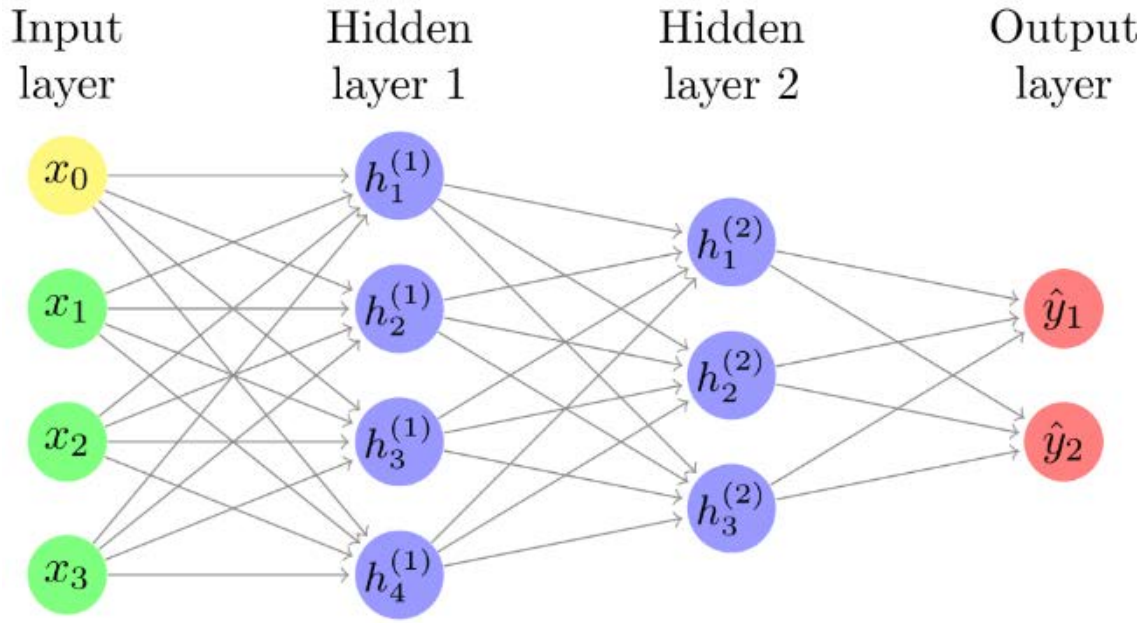
3.2 Βαθιά Μάθηση – Deep Learning

Η βαθιά μάθηση (Deep Learning) αποτελεί μια υποκατηγορία της τεχνητής νοημοσύνης που αναφέρεται στη χρήση βαθιών νευρωνικών δικτύων για την εκμάθηση και την αναγνώριση προτύπων από δεδομένα. Αυτά τα δίκτυα αποτελούνται από πολλαπλά επίπεδα, κάθε ένα από τα οποία επεξεργάζεται και αφαιρεί χρήσιμες πληροφορίες από τα δεδομένα εισόδου. Καθώς η επεξεργασία γίνεται σε πολλά επίπεδα, το δίκτυο μαθαίνει και αναγνωρίζει πολύπλοκα πρότυπα, πράγμα που το καθιστά κατάλληλο για προβλήματα όπως η αναγνώριση εικόνων, η φωνητική αναγνώριση και η επεξεργασία φυσικής

γλώσσας (NLP). Για να μπορέσουν τέτοιου είδους νευρωνικά δίκτυα να μάθουν τα πρότυπα με σκοπό να επιλύσουν κάποια συγκεκριμένη εργασία, απαιτείται συνήθως μεγάλη ποσότητα δεδομένων και υψηλή υπολογιστική ισχύς.

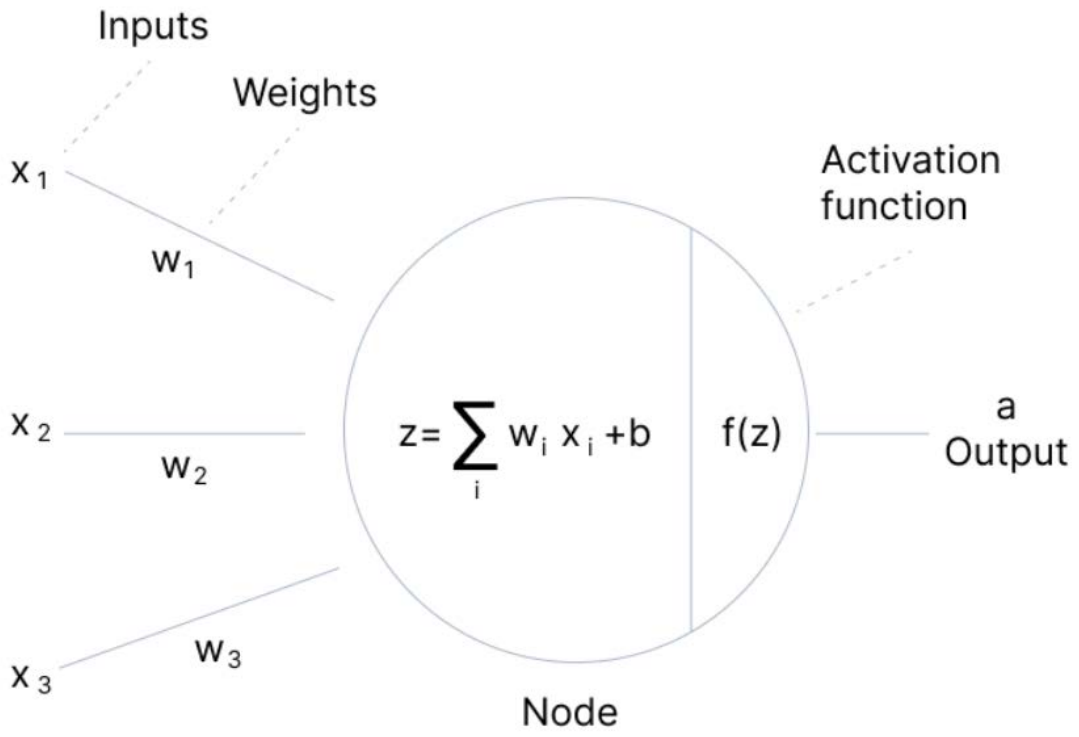
3.2.1 Απλό Τεχνητό Νευρωνικό Δίκτυο

Ένα απλό τεχνητό νευρωνικό δίκτυο είναι ένα σύστημα που αποτελείται από πολλαπλά στοιχεία επεξεργασίας που είναι εμπνευσμένα από τη λειτουργία των βιολογικών νευρώνων. Το νευρωνικό δίκτυο αποτελείται από διασυνδεδεμένα επίπεδα (layers) νευρώνων, και κάθε νευρώνας λειτουργεί ως μια μικρή μονάδα επεξεργασίας. Οι νευρώνες λαμβάνουν εισόδους από άλλους νευρώνες ή από εξωτερικές πηγές και παράγουν έξοδο με βάση την είσοδο που λαμβάνουν. Η είσοδος σε κάθε νευρώνα υπολογίζεται ως ένας συνδυασμός των εισόδων από άλλους νευρώνες ή από εξωτερικές πηγές, και η έξοδος του νευρώνα υπολογίζεται με μια μη γραμμική συνάρτηση ενεργοποίησης.



Εικόνα 1: Βασική αρχιτεκτονική Νευρωνικού Δικτύου.

Κάθε νευρώνας λαμβάνοντας είσοδο από εξωτερικές πηγές ή από άλλους νευρώνες υπολογίζει μια γραμμική συνάρτηση με βάση τα βάρη και την προκατάληψη (bias).



Εικόνα 2: Βασική Λειτουργία ενός Νευρώνα.

Πολλές φορές κυρίως σε προβλήματα ταξινόμησης που χρειαζόμαστε κατανομή πιθανότητας, η γραμμικότητα που προκύπτει από τους υπολογισμούς του κάθε νευρώνα δεν βοηθάει στην επίλυση του προβλήματος. Γι' αυτό το λόγο, πριν την έξοδο του νευρώνα και μετά τον υπολογισμό του αθροίσματος των βαρών και της προκατάληψης, για κάθε είσοδο, εφαρμόζεται μια μη γραμμική συνάρτηση. Οι πιο γνωστές από αυτές είναι:

- Softmax: Σκοπός αυτής της συνάρτησης είναι να μετατρέψει μια σειρά αριθμών σε κατανομή πιθανοτήτων που αθροίζουν στο 1.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}, \text{για } i = 1, 2, ..., n.$$

- Sigmoid: Σκοπός αυτής της συνάρτησης είναι να μετατρέψει μια σειρά αριθμών σε πιθανότητες που δεν αθροίζουν στο 1 και είναι ανεξάρτητες μεταξύ τους.

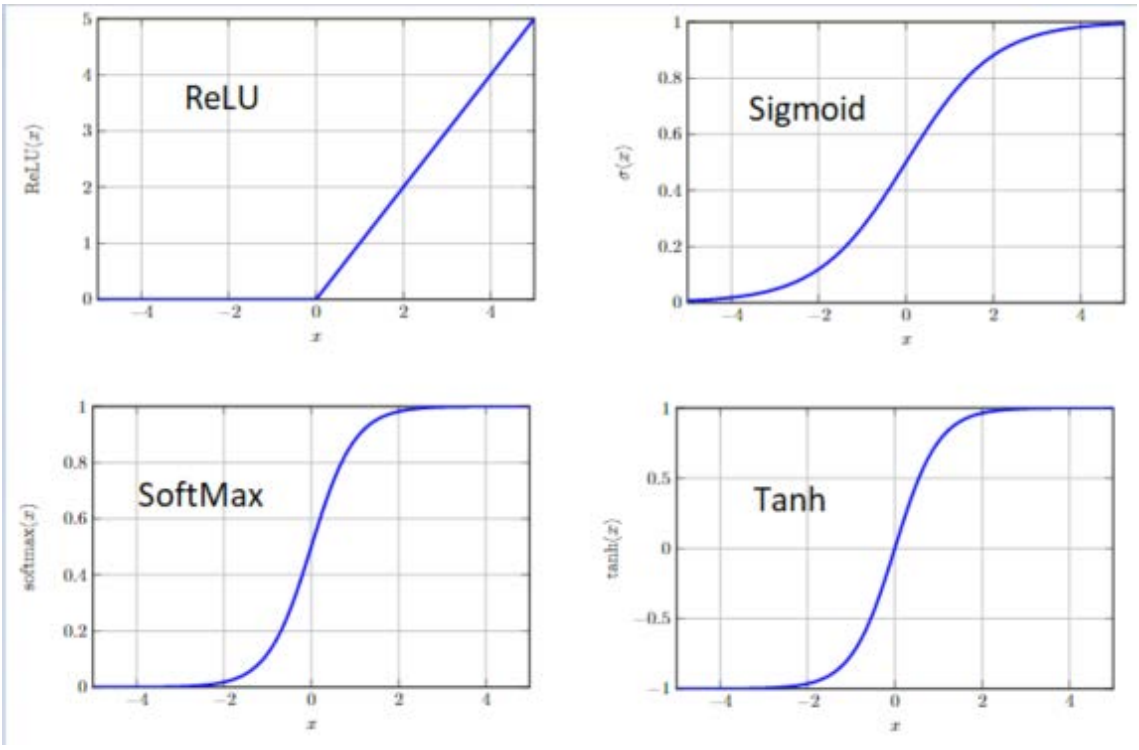
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- Relu: Σκοπός αυτής της συνάρτησης είναι να μετατρέπει τις αρνητικές τιμές σε μηδέν. Χρησιμοποιείται κυρίως στα ενδιάμεσα επίπεδα για την αποφυγή του overfitting.

$$\sigma(z) = \max(0, z)$$

- Tanh: Σκοπός αυτής της συνάρτησης είναι να εξάγει μια μη γραμμικότητα διατηρώντας όμως τα θετικά και αρνητικά πρόσημα των τιμών που θα εισέλθουν στην είσοδό της.

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$



Εικόνα 3: Βασικές Συναρτήσεις Ενεργοποίησης.

Για να μπορέσει ένα νευρωνικό δίκτυο να εκπαιδευτεί χρειάζεται ένας αλγόριθμος όπου θα βοηθήσει στην διαδικασία της οπισθοδρόμησης του λάθους (back propagation). Αυτοί οι αλγόριθμοι βοηθάνε το εκάστοτε μοντέλο να βελτιωθεί. Είναι γνωστοί ως Optimizers. Μερικοί γνωστοί και ευρέως χρησιμοποιούμενοι από αυτούς είναι:

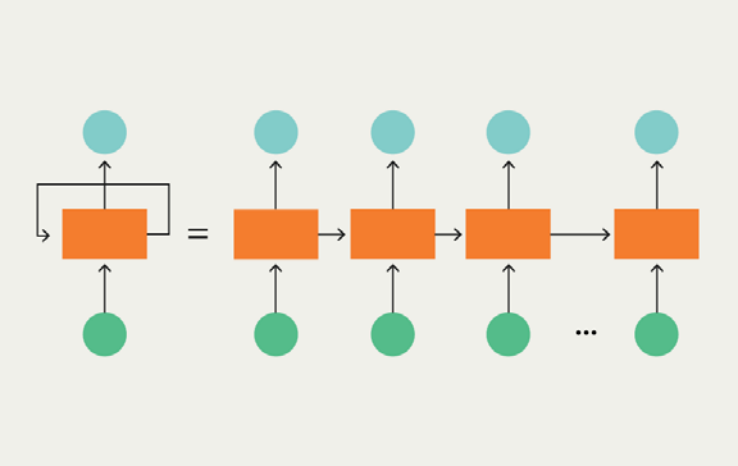
- Stochastic Gradient Descent (SGD): Είναι ένας αλγόριθμος βελτιστοποίησης που βασίζεται στον αλγόριθμο Gradient Descent (GD), ο οποίος αναζητά το ελάχιστο μιας συνάρτησης κόστους. Στην περίπτωση

του GD η ενημέρωση και βελτιστοποίηση των παραμέτρων του μοντέλου γίνεται για κάθε παράδειγμα (sample) ξεχωριστά. Στην περίπτωση του SGD επιλέγονται τυχαία ομάδες (batches) από δείγματα για να γίνει αυτή η βελτιστοποίηση. Αυτό λύνει το πρόβλημα της πολυπλοκότητας ειδικά όταν έχουμε πάρα πολλά δεδομένα. Αυτοί οι αλγόριθμοι βελτιστοποίησης έχουν μια υπερ παράμετρο που ονομάζεται learning rate. Για το οποίο είναι σημαντικό να βρεθεί πειραματικά η κατάλληλη τιμή ώστε να αποφευχθεί το Overfitting και underfitting.

- adaptive moment estimation (Adam): Αποτελεί μια παραλλαγή του SGD με το πλεονέκτημα ότι δοκιμάζει πολλαπλά και διαφορετικά learning rates. Είναι ένας συνδυασμός των αλγορίθμων *Adaptive Gradient Algorithm* και του *Root Mean Square Propagation (RMSPProp)*.

Το απλό νευρωνικό δίκτυο μπορεί να χρησιμοποιηθεί για προβλήματα ταξινόμησης, παλινδρόμησης, αναγνώρισης προτύπων και πρόβλεψης. Έχει χρησιμοποιηθεί επιτυχώς σε πολλούς τομείς, όπως η αναγνώριση φωνής, η αναγνώριση προτύπων σε εικόνες και η επεξεργασία φυσικής γλώσσας. Η αρχιτεκτονική αυτή που περιέχει πάνω από δύο κρυμμένα επίπεδα ονομάζεται Βαθύ Νευρωνικό Δίκτυο.

3.2.2 RNN



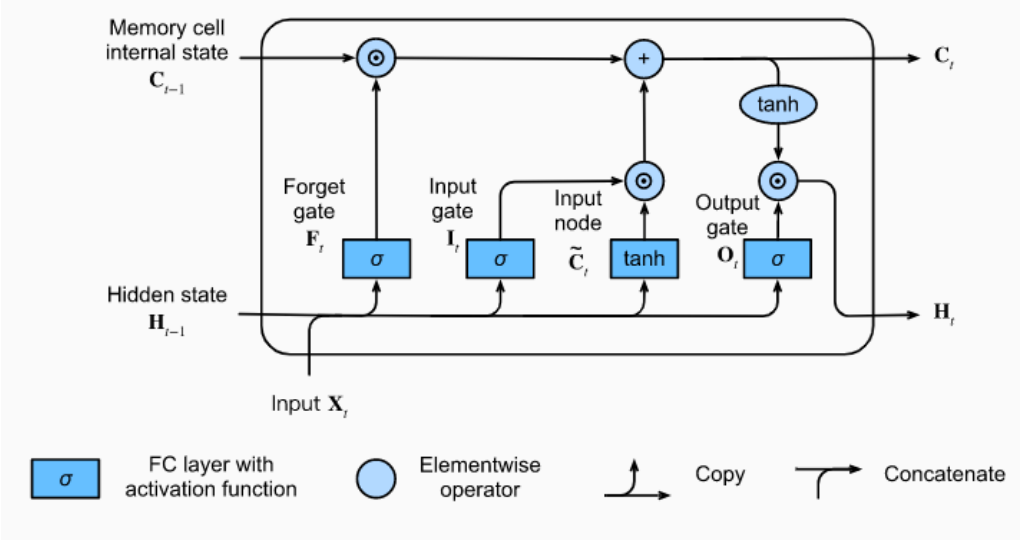
Εικόνα 4: Αρχιτεκτονική RNN.

Το RNN (Recurrent Neural Network) είναι ένα είδος τεχνητού νευρωνικού δικτύου που έχει σχεδιαστεί ειδικά για να επεξεργάζεται ακολουθίες δεδομένων, όπως κείμενο, ήχο και χρονοσειρές. Τα RNN συνδέουν τους νευρώνες μεταξύ τους σε μια αλυσίδα, όπου η έξοδος του ενός νευρώνα αντιστοιχεί στην είσοδο του επόμενου. Αυτή η δομή επιτρέπει στα RNN να διατηρούν μια κατάσταση μνήμης, που κρατάει πληροφορίες από τα προηγούμενα στοιχεία της ακολουθίας, και να τις χρησιμοποιούν για την παραγωγή των επόμενων στοιχείων. Τα RNN έχουν τη δυνατότητα να επεξεργάζονται ακολουθίες δεδομένων διαφορετικού μήκους και να διατηρούν πληροφορία σχετικά με το παρελθόν σε μια κρυφή κατάσταση. Αυτό τα καθιστά ιδιαίτερα χρήσιμα για την επεξεργασία ακολουθιών όπου η σειρά των στοιχείων έχει σημασία.

Τα RNN έχουν εφαρμογές σε πολλούς τομείς, όπως η αναγνώριση φωνής, η αναγνώριση χειρόγραφων χαρακτήρων, η αναγνώριση αντικειμένων σε εικόνες και η παραγωγή κειμένου.

Ένα από τα κύρια προβλήματα που αντιμετωπίζουν τα RNN είναι το γνωστό πρόβλημα του Vanishing Gradients ή το Exploding Gradients. Κατά τη διάρκεια της εκπαίδευσης, οι παράγωγοι (gradients) μπορεί να γίνουν πολύ μικροί (Vanishing Gradients) ή πολύ μεγάλοι (Exploding Gradients), κάνοντας την εκπαίδευση δύσκολη ή αδύνατη με αποτέλεσμα να παθαίνουν πολύ εύκολα overfitting ή underfitting. Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι η χρήση συστημάτων RNN με ειδικές αρχιτεκτονικές, όπως τα LSTM (Long Short-Term Memory).

3.2.3 LSTM



Εικόνα 5: Αρχιτεκτονική LSTM.[2]

Το LSTM (Long Short-Term Memory) είναι ένα είδος RNN που χρησιμοποιείται συνήθως για την επεξεργασία σειρών, χρονοσειρών, ακολουθίες κειμένων και την αναγνώριση προτύπων σε αυτές.

Το LSTM διατηρεί μια εσωτερική κατάσταση που επιτρέπει στο δίκτυο να "θυμάται" τις προηγούμενες πληροφορίες και να λαμβάνει αποφάσεις με βάση αυτές τις πληροφορίες. Αυτό είναι χρήσιμο όταν οι ακολουθίες έχουν πολύπλοκες δομές ή διακυμάνσεις στο χρόνο που είναι δύσκολο να αναγνωριστούν από απλά νευρωνικά δίκτυα. Επιπλέον αυτό το είδος νευρωνικού δικτύου, αποτελείται από μια αλυσίδα επαναλαμβανόμενων μονάδων (recurrent units) που διαθέτουν τρεις διαφορετικές πύλες (gates) που ελέγχουν ποια πληροφορία πρέπει να διατηρηθεί, ποια πληροφορία πρέπει να απορριφθεί και ποια πληροφορία πρέπει να ενημερωθεί. Η απόφαση που λαμβάνεται στις πύλες εξαρτάται από την είσοδο του δικτύου και την εσωτερική κατάσταση των προηγούμενων χρονικών βημάτων.

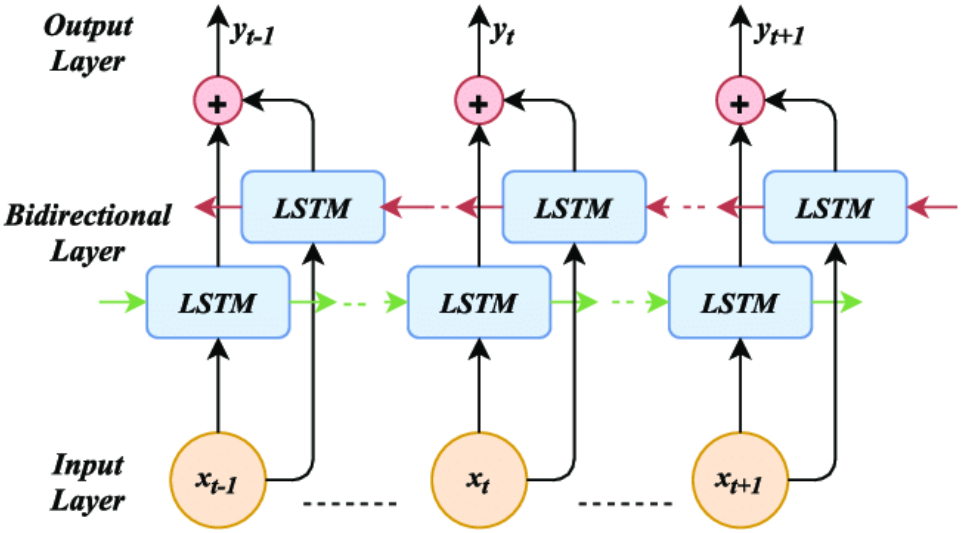
Συνολικά, το LSTM μπορεί να μάθει να επεξεργάζεται μακροπρόθεσμες εξαρτήσεις μεταξύ των στοιχείων μιας σειράς χρονοσειρών ή ακολουθιών λέξεων ή χαρακτήρων στην περίπτωση που τα δεδομένα είναι κείμενα. Αυτό επιτρέπει

[2] https://d2l.ai/chapter_recurrent-modern/lstm.html

στο δίκτυο να προβλέπει την επόμενη τιμή στη σειρά, βάσει των προηγούμενων τιμών και της σχέσης τους μεταξύ τους.

Το LSTM έχει εφαρμογές σε πολλούς τομείς, όπως η αναγνώριση φωνής, η αναγνώριση κειμένου, η μηχανική μετάφραση, η πρόβλεψη τιμών μετοχών, η πρόβλεψη καιρού και πολλές άλλες.

3.2.4 Bi-LSTM



Εικόνα 6: Αρχιτεκτονική Bi-LSTM. [19]

Το Bi-LSTM (Bidirectional LSTM) είναι μια παραλλαγή του LSTM, που διαθέτει δύο συστήματα LSTM που λειτουργούν ταυτόχρονα στην είσοδο του δικτύου, ένα από την αριστερή πλευρά και ένα από τη δεξιά πλευρά. Κάθε σύστημα αναλαμβάνει να επεξεργαστεί την είσοδο σε αντίθετη κατεύθυνση, δηλαδή το ένα σύστημα αναλαμβάνει να επεξεργαστεί την είσοδο από τα αριστερά προς τα δεξιά και το άλλο σύστημα αναλαμβάνει να επεξεργαστεί την είσοδο από τα δεξιά προς τα αριστερά.

Η ιδέα πίσω από το Bi-LSTM είναι ότι καθώς κάθε σύστημα αναλαμβάνει να επεξεργαστεί την είσοδο σε αντίθετη κατεύθυνση, το δίκτυο μπορεί να εξάγει περισσότερη πληροφορία από τις ακολουθίες δεδομένων. Στην περίπτωση της πρόβλεψης τιμών χρηματιστηρίου, για παράδειγμα, το Bi-LSTM μπορεί να αξιοποιήσει πληροφορία από προηγούμενες και μελλοντικές τιμές των μετοχών. Στην περίπτωση του κειμένου, ομοίως μπορεί να αξιοποιήσει πληροφορία για την εκάστοτε λέξη, καθώς ιδιαίτερα στα κείμενα η κάθε διαφορετική λέξη αποκτά

διαφορετικό νόημα ανάλογα με την θέση της στο κείμενο αλλά και το είδος του κειμένου.

3.2.5 Embeddings

Στην επεξεργασία φυσικής γλώσσας (NLP), τα Embeddings είναι αριθμητικές αναπαραστάσεις λέξεων ή φράσεων που αποτυπώνουν το σημασιολογικό νόημα του κειμένου. Τα Embeddings δημιουργούνται χρησιμοποιώντας αλγόριθμους μηχανικής ή βαθιάς μάθησης που αναλύουν μεγάλες ποσότητες δεδομένων κειμένου και δημιουργούν μια διανυσματική αναπαράσταση (Vectors) κάθε λέξης με βάση το περιεχόμενό της και το νόημά της στην εκάστοτε πρόταση.

Η βασική ιδέα πίσω από τις ενσωματώσεις λέξεων είναι η αντιστοίχιση κάθε λέξης στο λεξιλόγιο σε ένα διάνυσμα υψηλής διάστασης σε έναν συνεχή χώρο. Η θέση του διανύσματος σε αυτό το διάστημα συλλαμβάνει το νόημα της λέξης, με παρόμοιες λέξεις να αντιπροσωπεύονται από διανύσματα που είναι κοντά μεταξύ τους και ανόμοιες λέξεις να αντιπροσωπεύονται από διανύσματα που απέχουν πολύ μεταξύ τους.

Τα Embeddings μπορούν να χρησιμοποιηθούν ως είσοδος σε αλγόριθμους μηχανικής και βαθιάς μάθησης για διάφορες εργασίες NLP, όπως ταξινόμηση κειμένου, ανάλυση συναισθημάτων και μηχανική μετάφραση. Μπορούν επίσης να χρησιμοποιηθούν για την εκτέλεση ανάλυσης σημασιολογικής ομοιότητας και για τη δημιουργία συστάσεων για σχετικό περιεχόμενο.

Υπάρχουν αρκετοί δημοφιλείς αλγόριθμοι για τη δημιουργία Embeddings, συμπεριλαμβανομένων των Word2Vec, GloVe και fastText. Αυτοί οι αλγόριθμοι διαφέρουν ως προς την προσέγγισή τους στην αναπαράσταση λέξεων και στις υπολογιστικές τους απαιτήσεις, αλλά όλοι στοχεύουν να συλλάβουν το νόημα των λέξεων με τρόπο που είναι χρήσιμος για εργασίες NLP. Στις μέρες μας, δεν είναι λίγοι οι επιστήμονες και οι ερευνητές που προτιμούν να χρησιμοποιήσουν κάποιο προ-εκπαιδευμένο γλωσσικό μοντέλο με σκοπό να εξάγουν Embeddings από αυτό.

3.2.6 Προ-Εκπαιδευμένο Γλωσσικό Μοντέλο

Ένα προ-εκπαιδευμένο γλωσσικό μοντέλο είναι ένας τύπος μοντέλου τεχνητής νοημοσύνης που έχει εκπαιδευτεί σε μεγάλες ποσότητες δεδομένων κειμένου για να μάθει τα μοτίβα και τις σχέσεις μεταξύ λέξεων, φράσεων και προτάσεων. Πρόκειται για μοντέλα που ακολουθούν την αρχιτεκτονική ενός Transformer. Αυτά τα μοντέλα χρησιμοποιούν μια διαδικασία που ονομάζεται μάθηση χωρίς επίβλεψη (unsupervised learning) για να αναλύσουν τη γλώσσα και να δημιουργήσουν μια αναπαράσταση του κειμένου (Embeddings) με τρόπο που επιτρέπει στο μοντέλο να δημιουργήσει νέο κείμενο ή να εκτελέσει άλλες εργασίες που σχετίζονται με τη γλώσσα.

Τα προ-εκπαιδευμένα γλωσσικά μοντέλα μπορούν να χρησιμοποιηθούν για μια ποικιλία εργασιών επεξεργασίας φυσικής γλώσσας (NLP), όπως ανάλυση συναισθήματος, ταξινόμηση κειμένου, απάντηση ερωτήσεων και αυτόματη μετάφραση. Μπορούν επίσης να βελτιωθούν σε συγκεκριμένες εργασίες με μικρότερο όγκο δεδομένων για να βελτιώσουν την απόδοσή τους σε αυτές τις εργασίες. Αυτή η διαδικασία ονομάζεται fine-tuning.

Μερικά δημοφιλή προ-εκπαιδευμένα μοντέλα γλώσσας περιλαμβάνουν το BERT (Bidirectional Encoder Representations from Transformers), το GPT (Generative Pre-training Transformer) και το RoBERTa (Robustly Optimized BERT Approach). Αυτά τα μοντέλα έχουν επιτύχει state-of-the-art αποτελέσματα σε ένα ευρύ φάσμα εργασιών NLP και έχουν γίνει βασικό συστατικό πολλών συστημάτων και εφαρμογών NLP. Δεν είναι λίγες οι εφαρμογές όπου χρησιμοποιούνται τέτοια μοντέλα για transfer learning με σκοπό την επίλυση ενός task.

3.2.7 Transformer

Ένας Transformer είναι ένας τύπος αρχιτεκτονικής νευρωνικού δικτύου που χρησιμοποιείται στην επεξεργασία φυσικής γλώσσας (NLP), αλλά και σε άλλου είδους δεδομένα, όπως η εικόνα, για την επεξεργασία διαδοχικών δεδομένων εισόδου, όπως το κείμενο. Για πρώτη φορά εισήχθη στην δημοσίευση «Attention Is All You Need» των Vaswani et al. το 2017 και έκτοτε έχει γίνει βασικό συστατικό πολλών σύγχρονων μοντέλων NLP.

Ένας Transformer αποτελείται από έναν κωδικοποιητή (Encoder) και έναν αποκωδικοποιητή (Decoder). Ο Encoder παίρνει μια ακολουθία από tokens, όπως

λέξεις ή χαρακτήρες, και παράγει μια ακολουθία από Embeddings που αποτυπώνουν το νόημα της εισόδου. Ο Decoder λαμβάνει τις αναπαραστάσεις (Embeddings) και δημιουργεί μια ακολουθία από tokens εξόδου, όπως λέξεις ή χαρακτήρες, με βάση την είσοδο και ένα σύνολο παραμέτρων που έχει μάθει ή θα μάθει κατά την διαδικασία του Fine-tuning σε ένα συγκεκριμένο task.

Η βασική καινοτομία του Transformer είναι η χρήση του self-attention μηχανισμού, ενός μηχανισμού που επιτρέπει στο μοντέλο να σταθμίζει τη σημασία διαφορετικών τμημάτων της ακολουθίας εισόδου κατά τη δημιουργία της ακολουθίας εξόδου. Τους επιτρέπει αρχικά να ξεχωρίζουν τα πραγματικά tokens της ακολουθίας από αυτά του padding. Δηλαδή το μοντέλο με βάση αυτόν τον μηχανισμό δεν λαμβάνει υπόψιν του τα padding tokens. Ο self-attention μηχανισμός επιτρέπει στο μοντέλο να καταγράφει εξαρτήσεις μεγάλης εμβέλειας μεταξύ των tokens εισόδου και εξόδου, κάτι που είναι ιδιαίτερα σημαντικό σε εργασίες NLP όπου η σημασία μιας λέξης μπορεί να εξαρτάται από την θέση στην οποία εμφανίζεται.

Οι Transformers έχουν επιτύχει state-of-the-art αποτελέσματα σε ένα ευρύ φάσμα εργασιών NLP, συμπεριλαμβανομένης της μοντελοποίησης γλώσσας, της μηχανικής μετάφρασης και της ταξινόμησης κειμένων. Μερικά δημοφιλή μοντέλα που βασίζονται σε μετασχηματιστές περιλαμβάνουν το BERT, το GPT και το T5.

3.2.8 BERT Model

Το BERT [20] Είναι ένα προηγμένο μοντέλο Βαθιάς μηχανικής μάθησης που χρησιμοποιείται για την κατανόηση φυσικής γλώσσας. Έχει εκπαιδευτεί στο “masked language modeling” Task και στο «next sentence prediction» Task.

- masked language modeling: Αποτελεί ένα πρόβλημα στον τομέα της Επεξεργασία φυσικής γλώσσας κατά το οποίο το εκάστοτε μοντέλο θα πρέπει να είναι σε θέση να προβλέπει μια λέξη ή token εντός του κειμένου, το οποίο όμως είναι κρυμμένο (masked). Η διαδικασία του masking είναι σταθερή και προκαθορισμένη κατά την εκπαίδευση.
- next sentence prediction: Αποτελεί ένα πρόβλημα στον τομέα της Επεξεργασία φυσικής γλώσσας κατά το οποίο το εκάστοτε μοντέλο θα

πρέπει να είναι σε θέση να καταλαβαίνει από ένα ζευγάρι προτάσεων, ένα η δεύτερη πρόταση είναι συνέχεια της προηγούμενης.

Το BERT χρησιμοποιείται για πολλές εφαρμογές φυσικής γλώσσας, όπως η αναγνώριση ονομαστικών οντοτήτων, η αναγνώριση συναισθήματος, η ανάλυση συντακτικών δομών και η μετάφραση μηχανής. Είναι ικανό να κατανοήσει το περιβάλλον της φράσης και τις συσχετίσεις μεταξύ των λέξεων, παρέχοντας έναν πλούσιο αναπαραστατικό χώρο (Embeddings) για την κατανόηση της φυσικής γλώσσας.

Το μοντέλο αυτό είναι ένας Transformer που αποτελείται από 1 Initial Embeddings Layer, 12 Encode Layers και 12 Attention Layers (για τον attention μηχανισμό). Το BERT δημιουργήθηκε το 2018 και από τότε έχουν προκύψει αρκετές παραλλαγές αυτού όπως το RoBERTa. Επίσης στο βασικό μοντέλο BERT μπορεί κανείς να βρει εκδόσεις με κάποια επιπλέον Layers (Classification Heads) που βοηθούν σε Classification προβλήματα. Ένα από αυτά είναι το Pooling Layer όπου ως έξοδο παράγει μη γραμμικά Embeddings Vectors από το CLS token (Classification token) της τελευταίας κρυφής κατάστασης του συνολικού μοντέλου (last hidden Layer). Η μη γραμμικότητα προέρχεται από το γεγονός ότι το Pooling Layer περιέχει στην έξοδό του την συνάρτηση ενεργοποίησης tanh. Το μήκος του Embeddings Vector που παράγει το μοντέλο BERT είναι 768 στην μικρή του έκδοση (base) και 1024 στην μεγάλη (Large). Έχουν γίνει αρκετές έρευνες όπου ελέγχουν την πληροφορία που παράγει κάθε Encode Layer[21], αλλά ακόμα δεν είναι σίγουρο με ακρίβεια για το τι παράγει τελικά κάθε Layer.

3.2.9 BERT Tokenizer

Αποτελεί ένα sub-module του μοντέλου BERT και χρησιμοποιείται για την προ-επεξεργασία δεδομένων κειμένου μετατρέποντας κείμενο σε αριθμητικά διακριτικά που μπορούν να γίνουν κατανοητά από το μοντέλο BERT. Ο συγκεκριμένος Tokenizer χρησιμοποιεί την τεχνική word Piece tokenization [22], η οποία είναι μια μέθοδος δημιουργίας tokens χωρίζοντας τις λέξεις σε sub words. Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη για τον χειρισμό λέξεων εκτός λεξιλογίου και σπάνιων λέξεων, καθώς μπορεί να τις αναλύσει σε πιο κοινές υπολέξεις. Για παράδειγμα η πρόταση «I love Natural Language Processing» έπειτα από το word piece tokenization θα γίνει ["i", "love", "natural", "language", "process", "##ing"].

Η διαδικασία του tokenization περιλαμβάνει τα ακόλουθα βήματα:

1. Ανάλογα με το ποια έκδοση BERT χρησιμοποιηθεί (cased ή uncased) μετατρέπει τα γράμματα του κειμένου σε πεζή μορφή. Στην περίπτωση της Cased εκδοχής τα αφήνει ως έχει.
2. Μετατρέπει τα tokens σε αριθμητικές τιμές (index) για να μπορεί να τα αναγνωρίσει το BERT.

Για την επίλυση οποιουδήποτε task χρειάζεται η ακολουθία από tokens που θα εισέλθει στο μοντέλο να είναι σταθερού μήκους. Για παράδειγμα 512 μήκος. Για αυτόν τον λόγο ο BERT Tokenizer μετά το tokenization εάν η ακολουθία που προέκυψε είναι λιγότερη από ένα συγκεκριμένο μήκος (π.χ. 512) κάνει padding με μηδενικά. Η τιμή του μηδέν ως padding index ακολουθείτε σε όλα τα κείμενα και είναι σταθερή.

Για να μπορέσει όμως το μοντέλο να καταλάβει που βρίσκονται τα πραγματικά tokens και όχι οι padding χαρακτήρες, καθώς αυτοί δεν προσφέρουν κάποια πληροφορία στην επίλυση ενός task, παράγονται και οι αριθμητικές τιμές για τον attention μηχανισμό. Αυτές οι τιμές είναι μια ακολουθία από τον αριθμό 1 μέχρι και το τελευταίο token της ακολουθίας το οποίο βρίσκεται πριν το πρώτο padding token αν υπάρχει και μια ακολουθία από μηδέν για τα padding tokens.

Τέλος το μοντέλο BERT έχει εκπαιδευτεί στο να δέχεται στην είσοδο του τα λεγόμενα Special Tokens. Τα βασικότερα από αυτά είναι:

- [SEP] token: Πρόκειται για ένα ειδικό token που βοηθάει το μοντέλο να ξεχωρίζει τις διαφορετικές προτάσεις στην είσοδο του. Συνήθως έχει νόημα να χρησιμοποιηθεί όταν πρόκειται να χρησιμοποιηθεί ο Cross-Attention Μηχανισμός. Αυτό το token όμως είναι υποχρεωτικό να χρησιμοποιηθεί ακόμα και σε μονή πρόταση στην είσοδο, δηλαδή στην χρήση του Self-Attention και Cross-Attention μηχανισμού. Αναδεικνύει πότε τελειώνει μια ακολουθία από tokens ή μια πρόταση. Ο index αυτού του token είναι η τιμή 102.
- [CLS] token: Πρόκειται για ένα ειδικό token που βοηθάει το μοντέλο σε Classification προβλήματα καθώς τα Embeddings αυτών γίνονται fine tuning κατά την φάση του Training. Αποτελεί την αρχή της ακολουθίας. Είναι επίσης υποχρεωτικό token και η τιμή του Index αυτού είναι το 101.

- [PAD] token: Χρησιμοποιείται εάν και εφόσον χρειάζεται να συμπληρωθεί το μέγιστο μήκος ακολουθίας που έχουμε ορίσει. Η τιμή του index αυτού του token είναι 0.

3.2.10 RoBERTa Model

Το RoBERTa (Robustly Optimized BERT Approach)[23] είναι ένα μοντέλο νευρωνικών δικτύων που βασίζεται στην αρχιτεκτονική του BERT (Bidirectional Encoder Representations from Transformers) με πρόσθετες βελτιώσεις στον τρόπο εκπαίδευσης και στην αναπαράσταση του κειμένου. Σε αντίθεση με το BERT κατά τη διάρκεια της εκπαίδευσης το masking των tokens γίνεται με δυναμικό και τυχαίο τρόπο.

Πιο συγκεκριμένα, το RoBERTa χρησιμοποιεί μια σειρά από βελτιώσεις στη διαδικασία εκπαίδευσης του BERT, όπως αυξημένο μέγεθος του batch, ανακατεμένα δείγματα (shuffling), διαφορετικό τρόπο επιλογής των δειγμάτων εκπαίδευσης και άλλες βελτιώσεις στην αναζήτηση υπερπαραμέτρων (hyperparameter tuning). Αυτές οι βελτιώσεις βελτιώνουν την απόδοση του μοντέλου σε μια ποικιλία από εργασίες επεξεργασίας φυσικής γλώσσας.

Επιπλέον, το RoBERTa επεκτείνει το BERT με μια μεγαλύτερη ποικιλία από δεδομένα εκπαίδευσης. Το μεγάλο σύνολο δεδομένων εκπαίδευσης, σε συνδυασμό με τις βελτιώσεις στη διαδικασία εκπαίδευσης, βοηθούν το RoBERTa να αναπαράγει καλύτερα το σημασιολογικό (semantic) περιεχόμενο του κειμένου και να επιτυγχάνει καλύτερα αποτελέσματα σε διάφορες εργασίες επεξεργασίας φυσικής γλώσσας.

Όπως και το BERT, το RoBERTa χρησιμοποιεί μια αρχιτεκτονική Transformers και είναι ένα μοντέλο που δέχεται ως είσοδο ένα κείμενο και παράγει ως έξοδο μια αναπαράσταση του κειμένου (embeddings) που περιέχει πληροφορία για το σημασιολογικό περιεχόμενό του.

Το RoBERTa έχει επιτύχει εξαιρετικά αποτελέσματα σε πολλές εργασίες επεξεργασίας φυσικής γλώσσας, συμπεριλαμβανομένης της αναγνώρισης ονοματισμένων οντοτήτων (NER), της αναγνώρισης σημασιολογικών σχέσεων μεταξύ λέξεων και της παραγωγής κειμένου.

3.2.11 RoBERTa Tokenizer

Αποτελεί ένα sub-module του μοντέλου RoBERTa και χρησιμοποιείται για την προ-επεξεργασία δεδομένων κειμένου μετατρέποντας κείμενο σε αριθμητικά διακριτικά που μπορούν να γίνουν κατανοητά από το μοντέλο RoBERTa. Ο συγκεκριμένος Tokenizer χρησιμοποιεί την τεχνική Byte Pair Encoding (BPE) [24] ,η οποία είναι μια τεχνική συμπίεσης δεδομένων που αντικαθιστά τα πιο συχνά ζεύγη χαρακτήρων με ένα νέο token, δημιουργώντας σταδιακά ένα λεξιλόγιο μονάδων υπολέξεων (sub words units) που μπορούν να αντιπροσωπεύουν οποιαδήποτε λέξη στο κείμενο εισόδου. Ο Tokenizer λειτουργεί διαιρώντας πρώτα το κείμενο εισαγωγής σε tokens με βάση τα κενά και τα σημεία στίξης. Αυτά τα διακριτικά στη συνέχεια χωρίζονται περαιτέρω σε μονάδες υπολέξεων χρησιμοποιώντας τον αλγόριθμο BPE. Ο Tokenizer διατηρεί ένα λεξιλόγιο μονάδων υπολέξεων, όπου κάθε υπολέξη αντιπροσωπεύεται από έναν ακέραιο αριθμό (index).

Όταν κωδικοποιεί ένα κομμάτι κειμένου, ο tokenizer εφαρμόζει πρώτα ένα βασικό tokenization για να χωρίσει το κείμενο σε tokens επιπέδου λέξης, στη συνέχεια εφαρμόζει BPE σε κάθε token σε επίπεδο λέξης για να το χωρίσει σε tokens σε επίπεδο υπολέξεων και τέλος μετατρέπει κάθε token σε επίπεδο υπολέξεων στον αντίστοιχο ακέραιο δείκτη (index) του στο λεξιλόγιο του tokenizer.

Τέλος το μοντέλο RoBERTa έχει εκπαιδευτεί στο να δέχεται στην είσοδο του τα λεγόμενα Special Tokens. Τα βασικότερα από αυτά είναι:

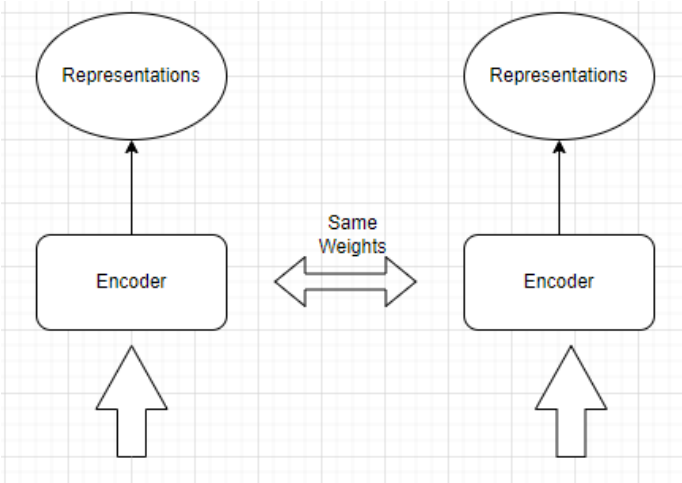
- </s> token: Πρόκειται για ένα ειδικό token που βοηθάει το μοντέλο να ξεχωρίζει τις διαφορετικές προτάσεις στην είσοδο του. Συνήθως έχει νόημα να χρησιμοποιηθεί όταν πρόκειται να χρησιμοποιηθεί ο Cross-Attention Μηχανισμός. Αυτό το token όμως είναι υποχρεωτικό να χρησιμοποιηθεί ακόμα και σε μονή πρόταση στην είσοδο, δηλαδή στην χρήση του Self-Attention μηχανισμού. Αναδεικνύει πότε τελειώνει μια ακολουθία από tokens ή μια πρόταση. Ο index αυτού του token είναι η τιμή 2.
- <s> token: Πρόκειται για ένα ειδικό token που βοηθάει το μοντέλο σε Classification προβλήματα καθώς τα Embeddings αυτών γίνονται fine

tuning κατά την φάση του Training. Αποτελεί την αρχή της ακολουθίας. Είναι επίσης υποχρεωτικό token και η τιμή του Index αυτού είναι το 0.

- <pad> token: Χρησιμοποιείται εάν και εφόσον χρειάζεται να συμπληρωθεί το μέγιστο μήκος ακολουθίας που έχουμε ορίσει. Η τιμή του index αυτού του token είναι 1.

3.2.12 Siamese Αρχιτεκτονική Νευρωνικού Δικτύου

Η Siamese αρχιτεκτονική ονομάζεται η δομή ενός νευρωνικού δικτύου κατά την οποία σε ένα γενικό μοντέλο όπου υπάρχει μια και ενιαία διαδικασία forward pass και back propagation, έχουμε εντάξει το ίδιο αντίγραφο (instance) ενός μοντέλου «δίπλα-δίπλα» έτσι ώστε να τρέχουν παράλληλα. Στην Siamese αρχιτεκτονική, κάθε αντίγραφο του νευρωνικού δικτύου λαμβάνει ένα δείγμα και εφαρμόζει ένα σύνολο επεξεργασιών για να εξάγει ένα σύνολο χαρακτηριστικών. Στη συνέχεια, τα χαρακτηριστικά αυτά συγκρίνονται μεταξύ τους, συνήθως με χρήση μιας συνάρτησης απόστασης ή ομοιότητας, για να προκύψει μια αριθμητική τιμή της ομοιότητας των δύο δειγμάτων. Είναι σημαντικό να αναφερθεί ότι κατά την δημιουργία μια τέτοιας αρχιτεκτονικής δεν πρέπει να δημιουργούνται νέα αντίγραφα ενός μοντέλου και τα βάρη των αντιγράφων να είναι τα ίδια. Μια βασική δομή μιας Siamese αρχιτεκτονικής παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 7: Βασική Siamese Αρχιτεκτονική.

Οι Siamese αρχιτεκτονικές χρησιμοποιούνται συνήθως για τον χειρισμό δεδομένων και την επίλυση ενός προβλήματος που απαιτούν σύγκριση και αντιστοίχιση.

4 Μεθοδολογία

Για την επίλυση του Authorship Verification Task δύο βασικές τεχνικές χρησιμοποιήθηκαν ως προς την εκπαίδευση του μοντέλου. Αυτές οι τεχνικές παρουσιάζονται παρακάτω.

4.1 Contrastive Learning

Η διαδικασία του Contrastive Learning αποτελεί μια τεχνική η οποία έχει αποδειχθεί ότι βελτιώνει την απόδοση των μοντέλων, χρησιμοποιώντας την αντίθεση των δειγμάτων μεταξύ τους για την εκμάθηση κοινών και μη χαρακτηριστικών ανάμεσα στα δείγματα.

Αυτή η τεχνική αποτελεί μια supervised μέθοδο η οποία ξεκίνησε να εφαρμόζεται σε vision προβλήματα και image classification αλλά τα τελευταία χρόνια έχει τεράστια απήχηση και στον τομέα του NLP. Ειδικότερα όταν πρέπει να δημιουργηθεί ένα μοντέλο που αφορά Verification δυο ή παραπάνω δεδομένων.

Βασικός Στόχος αυτής της μεθόδου είναι η εκπαίδευση ενός νευρωνικού δικτύου, ώστε να αναγνωρίζει τη διαφορά μεταξύ δυο δειγμάτων που ανήκουν στην ίδια κατηγορία (positive pair) και δύο δειγμάτων που ανήκουν σε διαφορετικές κατηγορίες (negative pair).

Κατά την διαδικασία της εκπαίδευσης το μοντέλο λαμβάνει στην είσοδό του 2 δείγματα από τα δεδομένα και προσπαθεί να εξάγει αναπαραστάσεις που αντιπροσωπεύουν το κοινό στοιχείο των δύο εισόδων και να τα απομονώσει από τα διαφορετικά στοιχεία τους.

Κατά αυτόν τον τρόπο το μοντέλο αφού έχει εκπαιδευτεί θα είναι σε θέση όταν εισέρχονται στην είσοδό του δύο κείμενα του ίδιου συγγραφέα να τα προβλέψει ως True ενώ ως False στην αντίθετη περίπτωση. Η διαδικασία του Contrastive Learning συνήθως απαιτεί μια Siamese αρχιτεκτονική.

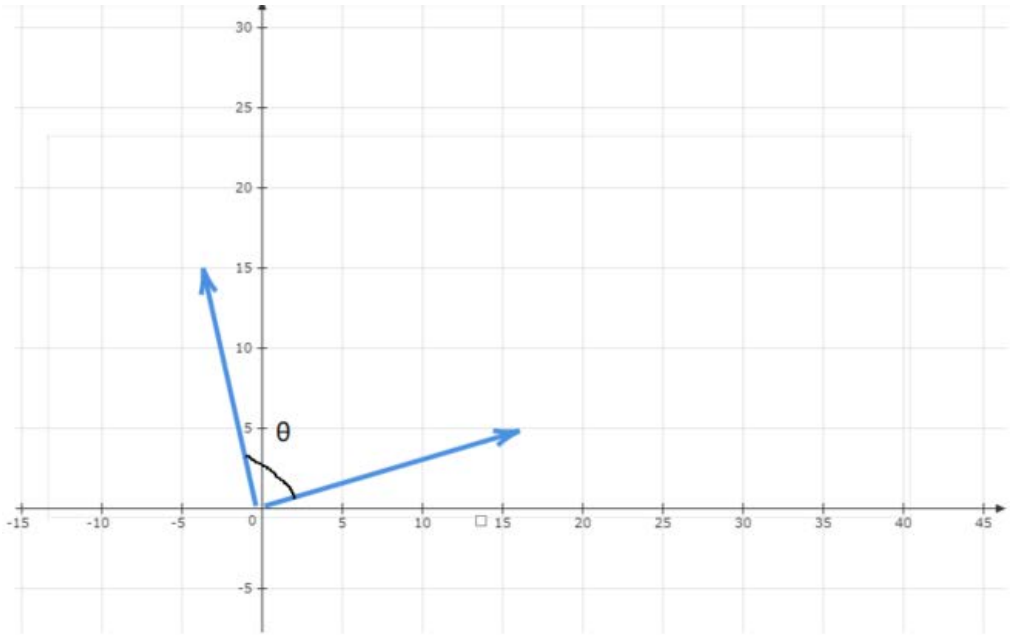
4.1.1 Διαδικασία Εκπαίδευσης με Contrastive Learning

Η διαδικασία της εκπαίδευσης ακολουθεί τα παρακάτω βήματα:

1. Είσοδος στο μοντέλο ενός ζευγαριού από τα δείγματα.
2. Εξαγωγή Αναπαραστάσεων.
3. Εφαρμογή κάποιας Distance ή similarity συνάρτησης για τον υπολογισμό της ομοιότητας του ζευγαριού.
4. Υπολογισμός Loss function.
5. Back Propagation.

Για να γίνει κατανοητή η παραπάνω διαδικασία ας δώσουμε ένα παράδειγμα.

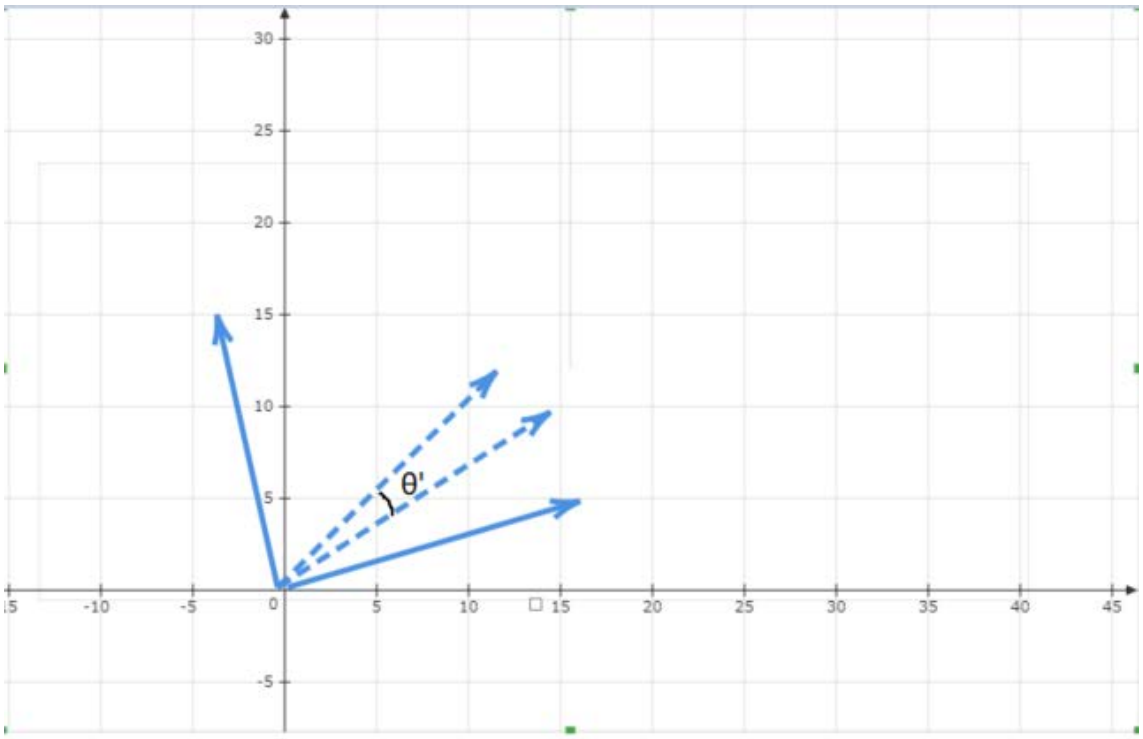
Έστω ότι το μοντέλο μας στην αρχή της εκπαίδευσης παράγει αναπαραστάσεις σε μορφή Διανυσμάτων (Vectors) στον πολυδιάστατο χώρο. Για την απλούστευση του παραδείγματος θα χρησιμοποιήσουμε τις 2 διαστάσεις μόνο. Άρα προκύπτει το παρακάτω σχήμα κατά το πρώτο forward pass από το μοντέλο.



Εικόνα 8: Αρχικοί Embeddings Vectors έπειτα από Forward Pass.

Έστω ότι τα 2 Vectors αυτά αντιστοιχούν σε ένα ζευγάρι κειμένων τα οποία προέρχονται από τον ίδιο συγγραφέα. Σύμφωνα με τον βασικό στόχο του Contrastive Learning θα πρέπει αυτά τα δύο Διανύσματα να συγκλίνουν το ένα προς το άλλο. Δηλαδή να μικρύνει η Ευκλείδεια απόσταση μεταξύ τους ή η γωνία που σχηματίζουν (Cosine Similarity).

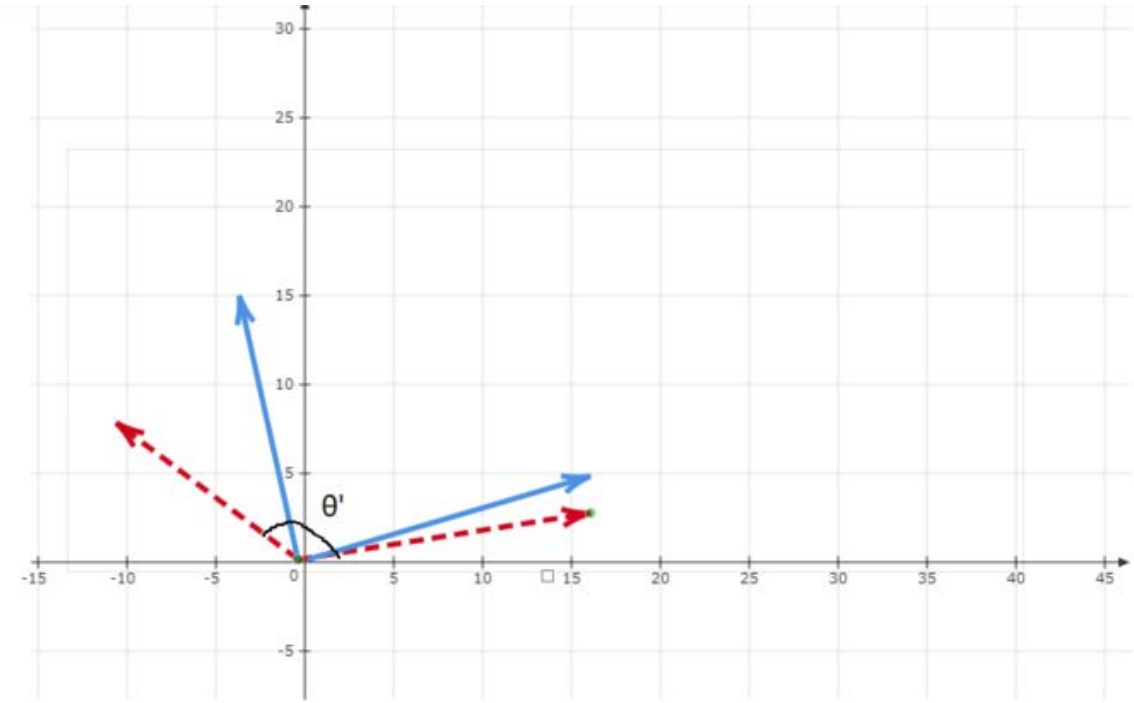
Έτσι λοιπόν μετά την διαδικασία του back propagation και το update των Βαρών του Μοντέλου θα πρέπει να προκύψει το παρακάτω σχήμα.



Εικόνα 9: Embeddings Vectors (ίδιου συγγραφέα) έπειτα από εφαρμογή Contrastive Loss και Back Propagation.

Παρατηρώντας κανείς τα διανύσματα με τις διακεκομμένες γραμμές μπορεί να δει ότι μετακινήθηκαν στον χώρο και σύγκλιναν μεταξύ τους.

Σύμφωνα με τα παραπάνω λοιπόν ένα ιδανικό μοντέλο για τα κείμενα του ίδιου συγγραφέα θα πρέπει να εξάγει διανύσματα όπου η γωνία μεταξύ τους θα είναι 0. Άρα η Cosine Similarity θα έχει τιμή 1. Αντίστοιχα για τα κείμενα διαφορετικών συγγραφέων θα πρέπει να συμβεί το αντίθετο και η Cosine Similarity θα πρέπει να έχει τιμή -1. Παρακάτω και η σχετική αναπαράσταση στους άξονες 2 διαστάσεων (κόκκινα διανύσματα).



Εικόνα 10:Embeddings Vectors (διαφορετικού συγγραφέα - κόκκινα βέλη) έπειτα από εφαρμογή Contrastive Loss και Back Propagation.

4.1.2 Contrastive Loss

Αναζητώντας κανείς την υπάρχουσα βιβλιογραφία στο διαδίκτυο μπορεί να βρει πολλές Loss Functions που σχετίζονται με το Contrastive Learning. Οι πιο βασικές παρουσιάζονται παρακάτω.

1. *Max Margin Contrastive Loss [25]:* είναι ένας τύπος loss function που χρησιμοποιείται συνήθως στην βαθιά μάθηση, ειδικά σε μοντέλα εκπαίδευσης για εκμάθηση ομοιότητας.

   Ο στόχος αυτής της συνάρτησης είναι η εκμάθηση ενός χώρου αναπαράστασης στον οποίο παρόμοια παραδείγματα χαρτογραφούνται το ένα κοντά στο άλλο και ανόμοια παραδείγματα χαρτογραφούνται μακριά. Η συνάρτηση αυτή το επιτυγχάνει αυτό «τιμωρώντας» το μοντέλο όταν παρόμοια παραδείγματα αντιστοιχίζονται σε μεγάλη απόσταση και όταν ανόμοια παραδείγματα χαρτογραφούνται κοντά.

   Η Loss υπολογίζεται με βάση ζεύγη παραδειγμάτων, όπου το ένα παράδειγμα θεωρείται το θετικό παράδειγμα και το άλλο θεωρείται το αρνητικό παράδειγμα. Το θετικό παράδειγμα είναι ένα παράδειγμα που είναι παρόμοιο με το άλλο παράδειγμα του ζεύγους, ενώ το αρνητικό παράδειγμα είναι ένα παράδειγμα που είναι ανόμοιο με το άλλο

παράδειγμα. Η Max Margin Contrastive Loss αποτελείται από δύο όρους: έναν όρο που αποτελεί την hinge loss[3] και έναν όρο περιθωρίου (margin). Η hinge loss μετρά την απόσταση μεταξύ των θετικών και αρνητικών παραδειγμάτων στον χώρο αναπαράστασης και τιμωρεί το μοντέλο εάν αυτή η απόσταση είναι πολύ μικρή. Ο όρος περιθωρίου ορίζει μια ελάχιστη απόσταση που πρέπει να διατηρεί το μοντέλο μεταξύ των θετικών και αρνητικών παραδειγμάτων. Η Contrastive loss μπορεί να βελτιστοποιηθεί χρησιμοποιώντας αλγόριθμους gradient descent, όπως stochastic gradient descent (SGD), για την ενημέρωση (update) των παραμέτρων και βαρών του μοντέλου που ελαχιστοποιούν την τιμή της Loss. Ελαχιστοποιώντας την Loss το μοντέλο μαθαίνει έναν χώρο αναπαράστασης στον οποίο παρόμοια παραδείγματα αντιστοιχίζονται το ένα κοντά στο άλλο και ανόμοια παραδείγματα χαρτογραφούνται μακριά, γεγονός που μπορεί να βελτιώσει την απόδοση ενός μοντέλου για μια συγκεκριμένη εργασία που βασίζεται στην ομοιότητα.

$$L(E_1, E_2, Y) = (Y)\frac{1}{2}D(E_1, E_2)^2 + (1 - Y)\frac{1}{2}\{max(0, m - D(E_1, E_2))\}^2$$

D: Απόσταση των 2 Embeddings Vectors ή Cosine Similarity.
E_1 & E_2 : Embeddings Vectors.
m: margin
Y: target label

2. *Modified Contrastive Loss with margins:* Αποτελεί μια παραλλαγή τη Max Margin Contrastive Loss. Σε αυτήν την συνάρτηση loss δεν υπάρχει μόνο μια παράμετρος margin αλλά δύο. Το ένα margin αντιστοιχεί και επιδρά μόνο στα θετικά ζευγάρια, εκεί δηλαδή που δύο κείμενα γράφτηκαν από τον ίδιο συγγραφέα, και το άλλο στα αρνητικά. Πρακτικά μπαίνει ένα άνω και κάτω όριο στο αποτέλεσμα της Cosine Similarity ή κάποια απόστασης. Εάν τα αποτελέσματα των μετρικών αυτών είναι μεγαλύτερα ή ίσα από το margin των θετικών τότε τα ζευγάρια αυτά δεν

[3] https://en.wikipedia.org/wiki/Hinge_loss

συνυπολογίζονται στην loss οπότε για αυτά η Loss είναι μηδέν. Ομοίως και για τα αρνητικά ζευγάρια.

$$L(E_1, E_2, Y) = (Y)\frac{1}{2}\{max\left(D(E_1, E_2) - m_p\right)^2 + (1-Y)\frac{1}{2}\left\{max(0, m_n - D(E_1, E_2))^2\right\}$$

D: Απόσταση των 2 Embeddings Vectors ή Cosine Similarity.

E_1 & E_2 : Embeddings Vectors.

m_p: margin θετικών (positive)

m_n: margin αρνητικών (negative)

Y: target label

3. *Triplet Loss:* γ συγκεκριμένη συνάρτηση Loss λειτουργεί όπως οι παραπάνω Contrastive loss με την διαφορά ότι δέχεται τριάδες από δείγματα. Συγκεκριμένα δεδομένου ενός δείγματος αναφοράς (anchor) ανακτούμε ένα δεύτερο δείγμα, το οποίο αντιστοιχεί στο θετικό δείγμα, με σκοπό να δημιουργηθεί ένα Positive pair και ένα τρίτο δείγμα, το οποίο αντιστοιχεί στο αρνητικό, με σκοπό να δημιουργηθεί το negative pair. Στην triplet loss, ο στόχος είναι να μειωθεί η απόσταση μεταξύ του anchor και του θετικού και να αυξηθεί μεταξύ του anchor και του αρνητικού.

$$L(A, E_p, E_n) = max\left\{0, \left(D(A, E_p)^2 - D(A, E_n)^2 + m_n \right) \right\}$$

D(A, Ep): Απόσταση ή ομοιότητα μεταξύ Anchor και Positive

D(A, En): Απόσταση ή ομοιότητα μεταξύ Anchor και Negative

m_n: margin αρνητικού

4. *InfoNCE Loss ή NT-Xent Loss:* είναι ένας τύπος loss function που χρησιμοποιείται συνήθως στην βαθιά μάθηση, ειδικά σε μοντέλα εκπαίδευσης για εκμάθηση ομοιότητας.

   Ο στόχος αυτής της συνάρτησης είναι η εκμάθηση ενός χώρου αναπαράστασης στον οποίο παρόμοια παραδείγματα χαρτογραφούνται το ένα κοντά στο άλλο και ανόμοια παραδείγματα χαρτογραφούνται μακριά. Η NT-Xent loss βασίζεται στη σύγκριση των χαρακτηριστικών εξόδου δύο εκδόσεων του ίδιου δείγματος (positive pair) και της εξόδου διαφορετικών δειγμάτων (negative pair). Πιο συγκεκριμένα, τα δείγματα εισάγονται στο νευρωνικό δίκτυο και παράγονται δύο χαρακτηριστικά εξόδου (embeddings) τα οποία συγκρίνονται με τη χρήση της κλασικής Cross-Entropy loss. Η βασική ιδέα της NT-Xent loss είναι να εφαρμοστεί

μια κλιμάκωση στην έξοδο του δικτύου προκειμένου να ενισχυθεί η διαφοροποίηση μεταξύ των θετικών και αρνητικών ζευγαριών. Η κλιμάκωση επιτυγχάνεται με τη χρήση ενός παράγοντα κλίμακας (scale factor) που αυξάνει την επιλογή των ακραίων τιμών των χαρακτηριστικών εξόδου, ενώ παράλληλα μειώνει την επίδραση των πιο "soft" τιμών (soft margins). Αυτό οδηγεί στον αποκλεισμό της πληροφορίας που μπορεί να μοιράζονται τα αρνητικά ζευγάρια στην επιλογή του χαρακτηριστικού διανύσματος. Η NT-Xent loss επίσης χρησιμοποιεί έναν temperature scaler για να αυξήσει την επίδραση των απομακρυσμένων ζευγαριών στην απώλεια. Ο scales αναπροσαρμόζει την κατανομή των αποστάσεων των χαρακτηριστικών εξόδου, καθιστώντας πιο εμφανείς τα απομακρυσμένα ζευγάρια.

$$L(X) = \frac{1}{N} \cdot \sum_{n=1}^{N} -\log \frac{\exp X_{n,n}}{\sum_{i=1}^{N} \exp X_{n,i}}$$

N: αριθμός δειγμάτων

X: Το εσωτερικό γινόμενο των Embeddings Vectors, πολλαπλασιασμένο επί τον temperature scaler τ.

$$X = E \cdot E'^{T} \cdot \tau$$

4.2 Απλό Classification Task

Το πρόβλημα του Authorship Verification, αφορά ένα Binary Classification πρόβλημα. Για αυτόν τον λόγο εκτελέστηκαν και πειράματα με πολλαπλές αρχιτεκτονικές με την χρήση της CrossEntropy Loss. Η επίλυση ενός Classification προβλήματος αποτελεί μια εργασία με γνωστά βήματα σε σχέση με αυτή του Contrastive Learning.

4.3 Threshold Finder για το Classification

Αναζητώντας κανείς τις βιβλιογραφίες στο διαδίκτυο μπορεί να βρει προβλήματα Classification, όπου αφού βρεθούν Similarities ή Dissimilarities μεταξύ δύο κειμένων εκτελούν μια διαδικασία εύρεσης κατώτατου ορίου (Threshold) για την ομοιότητα.

Στην συγκεκριμένη διπλωματική εργασία έπειτα από το Contrastive Learning για την επίλυση του Authorship Verification εκτελέστηκε η διαδικασία εύρεσης ενός Threshold με βάση το Validation set ως εξής:

1. Εξαγωγή Embeddings Vectors για τα δύο κείμενα ενός ζευγαριού.
2. Υπολογισμό Cosine Similarity
3. Grid search ώστε να βρεθεί με πολλαπλά πειράματα το καλύτερο νούμερο από το οποίο και πάνω έχουμε τα θετικά ζευγάρια.
4. Κάτω από αυτό έχουμε τα αρνητικά.

   Επειδή όμως χρειαζόμαστε και πιθανότητες, αυτές μπορούν να υπολογιστούν με βάση την ομοιότητα στην οποία εφαρμόζουμε κανονικοποίηση ώστε να έχει τιμές μεταξύ 0 και 1, επειδή χρειαζόμαστε πιθανότητες. Σε κάθε άλλη περίπτωση υπολογίζοντάς 1-similarity θα έχουμε την πιθανότητα του αρνητικού ζευγαριού.

   Επιπλέον τρόπος εύρεσης ορίου είναι να βρεθούν 2 Thresholds για τα θετικά και τα αρνητικά αντίστοιχα με την ίδια διαδικασία ή με βάση τα margins της Contrastive Loss. Τα ζευγάρια τα οποία έχουν Cosine Similarity με τιμές μεταξύ των 2 Thresholds, μπορούν να αφεθούν αναπάντητες ή να έχουν πιθανότητα 0.5 στο τελικό αποτέλεσμα εάν και εφόσον μας το επιτρέπει το πρόβλημα που πάμε να επιλύσουμε.

4.4 Χρήση των δεδομένων

Για την εκπαίδευση του εκάστοτε μοντέλου εφαρμόστηκαν επιγραμματικά οι παρακάτω τεχνικές:

- Προ-επεξεργασία των κειμένων.
- Tokenization με χρήση του Tokenizer του εκάστοτε Προ-εκπαιδευμένου γλωσσικού μοντέλου.
- Chunking κειμένων.

Αναζητώντας κανείς στην βιβλιογραφία στο διαδίκτυο, θα δει ότι για την επίλυση προβλημάτων όπως το Authorship Verification ή το Authorship Attribution υπάρχουν δυο βασικές τεχνικές κατά τις οποίες μπορεί κανείς να επεξεργαστεί και να δημιουργήσει την μορφή των δεδομένων που θα εισέλθουν ως είσοδο στο εκάστοτε μοντέλο. Αυτές είναι:

1. *Profile Based:* αποτελεί μια μεθοδολογία κατά την οποία γίνεται συλλογή και συνένωση όλων των κειμένων ενός συγγραφέα με σκοπό να εξαχθούν χαρακτηριστικά για το στυλ συγγραφής του.
2. *Instance Based:* αποτελεί μια μεθοδολογία κατά την οποία κάθε διαφορετικό κείμενο (sample) ενός συγγραφέα εισέρχεται στην είσοδο του μοντέλου ξεχωριστά. Αυτή η μεθοδολογία εφαρμόζεται κυρίως σε μοντέλα όπου υπάρχει περιορισμός στο μήκος ακολουθίας που μπορούν να δεχτούν ως είσοδο.

Το βασικό πρόβλημα των προ-εκπαιδευμένων γλωσσικών μοντέλων είναι το μέγιστο μήκος ακολουθίας που μπορούν να δεχτούν στην είσοδό τους. Συγκεκριμένα τα μοντέλα που χρησιμοποιήθηκαν (BERT & RoBERTa) στις βασικές εκδόσεις τους μπορούν να δεχτούν μέχρι 512 μήκος ακολουθίας έπειτα από το tokenization. Για την επίλυση του προβλήματος αυτού επιλέχθηκε η τεχνική του chunking. Δηλαδή το σπάσιμο του tokenized κειμένου σε μικρότερα κείμενα (chunks) μέχρι μήκος ακολουθίας 512 tokens. Επομένως πρόκειται για χρήση της τεχνικής Instance Based, καθώς υπάρχει ο περιορισμός της μέγιστης ακολουθίας στην είσοδο. Αναλυτικότερα παρακάτω.

4.4.1 Προ-επεξεργασία κειμένων

Το κάθε διαφορετικό σώμα δεδομένων χρήζει και διαφορετικής μεταχείρισης και επεξεργασίας. Συνήθως χρησιμοποιώντας τα προ-εκπαιδευμένα γλωσσικά μοντέλα δεν απαιτείται κάποια ιδιαίτερη επεξεργασία στα κείμενα, καθώς αυτά τα μοντέλα έχουν εκπαιδευτεί στο να μαθαίνουν να πιάνουν το νόημα ενός κειμένου και άρα να παράγουν Contextualized Embeddings. Σκοπός ενός Επαληθευτή είναι η αναγνώριση συγκεκριμένων patterns, έτσι ώστε να μπορεί να ξεχωρίζει εάν ένα ζευγάρι κειμένων είναι γραμμένο από τον ίδιο ή όχι συγγραφέα.

Για την επίλυση των προβλημάτων ακολουθήθηκαν δυο τεχνικές ξεχωριστά και ο συνδυασμός αυτών.

1. Αντικατάσταση όλων των αριθμητικών ψηφίων με το ψηφίο 1, διατηρώντας όμως την μορφολογία του κειμένου. Για παράδειγμα έστω ότι μέσα σε ένα κείμενο μπορεί έχουμε τα παρακάτω:

a. Ημερομηνία της μορφής 9/21/2023. Κάποιο άλλο άτομο θα μπορούσε να την γράψει αλλιώς. Για την ανίχνευση του στυλ συγγραφής ιδανικά θα θέλαμε να κρατήσουμε την μορφολογία αυτής της ημερομηνίας. Επομένως έπειτα από την αντικατάσταση αυτή θα γίνει 1/11/1111.
b. Κάποιος αριθμός πλήθους όπως 1500. Με την ίδια λογική θα γίνει 1111.

2. Αντικατάσταση των Named Entities με κάποιο συγκεκριμένο της υπάρχουσας συλλογής. Τα Entities αυτά μπορεί να ποικίλουν. Στην συγκεκριμένη εργασία έγινε αντικατάσταση μόνο των οντοτήτων που αντιστοιχούν σε ανθρώπους (κύρια ονόματα, ιδιότητες κλπ.) και σε γεωγραφικές περιοχές (πόλη, χώρα κλπ.). Για παράδειγμα:
   a. Αντικατάσταση του Ονόματος Elizabeth με κάποιο όνομα από την υπάρχουσα συλλογή κειμένων, το οποίο ο Tokenizer του εκάστοτε μοντέλου το αφήνει στην αρχική του μορφή χωρίς να το σπάσει σε Word pieces επειδή δεν ανήκει στο λεξιλόγιό του. Π.χ. το όνομα David.
   b. Αντικατάσταση κάποιας περιοχής με την λέξη Mexico.

   Κατά αυτόν τον τρόπο όλα τα Entities τύπου PERSON έχουν μια συγκεκριμένη τιμή David μέσα σε όλα τα κείμενα της εκάστοτε συλλογής. Το ίδιο ισχύει και για της γεωγραφικές περιοχές, οι οποίες λαμβάνουν την σταθερή τιμή Mexico. Η χρήση συγκεκριμένων τιμών προέκυψε πειραματικά έπειτα από tokenization.
3. Συνδυασμός των παραπάνω.

4.4.2 Tokenization

Για τα πειράματα στην συγκεκριμένη εργασία χρησιμοποιήθηκαν οι παρακάτω Tokenizers.

1. Bert Tokenizer
2. RoBerta Tokenizer

4.4.3 Chunking

Πρόκειται για μια τεχνική data augmentation, κατά την οποία από ένα μεγάλο κείμενο προκύπτουν πολλά μικρότερα κείμενα.

Σε αυτήν την τεχνική μπορεί κανείς να εφαρμόσει πολλαπλές μεθοδολογίες όπως:

- Chunking με overlapping μεταξύ των μικρότερων κειμένων. Δηλαδή το κάθε chunk να ξεκινάει από το τέλος του προηγούμενου chunk. Να υπάρχει επομένως μια επικάλυψη μεταξύ των chunks. Κατά αυτόν τον τρόπο διατηρούμε την συνοχή του κειμένου και το νόημά του αλλά επίσης διατηρούμε ίδιο μήκος ακολουθίας και την ομοιομορφία των δεδομένων που θα εισέλθουν στο μοντέλο χωρίς να κάνουμε χρήση των pad tokens.
- Chunking χωρίς overlapping. Κατά αυτόν τον τρόπο το κάθε κείμενο κόβεται έπειτα από ένα συγκεκριμένο μήκος, με αποτέλεσμα να προκύψουν πολλά chunks ανεξάρτητα μεταξύ τους. Σε αυτήν την μέθοδο χρειάζεται padding κυρίως στα τελευταία chunks.
- Chunking έπειτα από ανακατάταξη (shuffle) των προτάσεων. Κατά αυτόν τον τρόπο προηγείται μια διαδικασία ανίχνευσης και εξαγωγής προτάσεων (sentencization). Έπειτα η συλλογή των προτάσεων αυτών ανακατεύεται και επαναδημιουργείται το αρχικό κείμενο. Τέλος εφαρμόζεται η τεχνική του chunking είτε με overlapping είτε χωρίς. Με αυτήν την μέθοδο χάνεται η συνοχή του κειμένου και το νόημα αυτού.

Για τα πειράματα στην παρούσα εργασία η τεχνική του chunking εφαρμόστηκε έπειτα από το tokenization του εκάστοτε μοντέλου. Τα σημεία στα οποία θα σπάσει το κάθε κείμενο σε μικρότερα κομμάτια ορίζεται με βάση το μέγιστο όριο ακολουθίας που ορίζουμε στο BERT. Άρα η επιλογή των chunks δεν προκύπτει χειροκίνητα αλλά με βάση το BERT. Για παράδειγμα εάν ορίσουμε στο BERT να δέχεται 512 μήκος ακολουθίας θα σπάσει το κάθε κείμενο σε κομμάτια των 512 μήκους. Δηλαδή εάν ένα κείμενο έχει 1300 μήκος λίστας από tokens τότε αυτό θα σπάσει στα παρακάτω chunks:

- Πρώτο chunk μήκους 512 tokens (μέχρι το 512 token στη σειρά).
- Δεύτερο chunk μήκους 512 tokens (από το 513 στην σειρά token μέχρι 1025)
- Τρίτο chunk από 1026 στην σειρά μέχρι το τέλος.

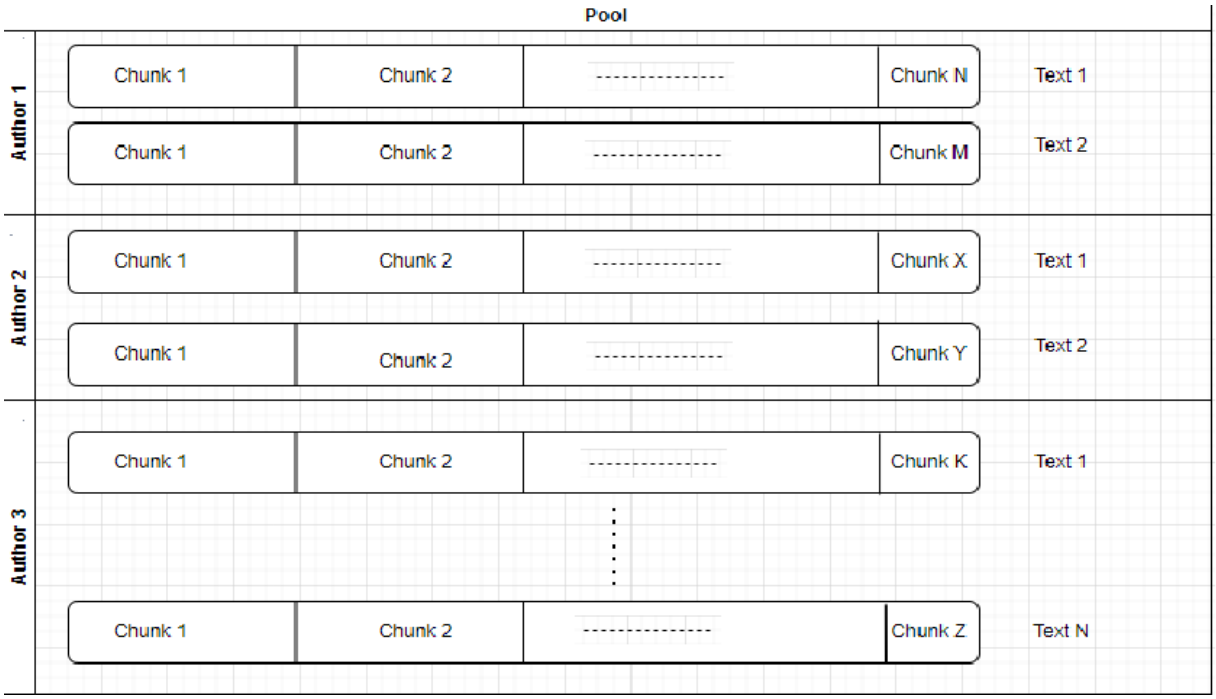
Με βάση το παραπάνω παράδειγμα θα μπορούσε να πει κανείς ότι το τρίτο chunk δεν έχει ίδιο μήκος με τα άλλα. Εφόσον κάποιο chunk δεν έχει ίδιο μήκος χάνουμε την ομοιομορφία των δεδομένων που εισέρχονται στο

μοντέλο. Άρα για αυτά τα chunks μπορούμε να εφαρμόσουμε την τεχνική του padding ή του overlapping.

4.4.4 Δημιουργία ζευγαριών κειμένων

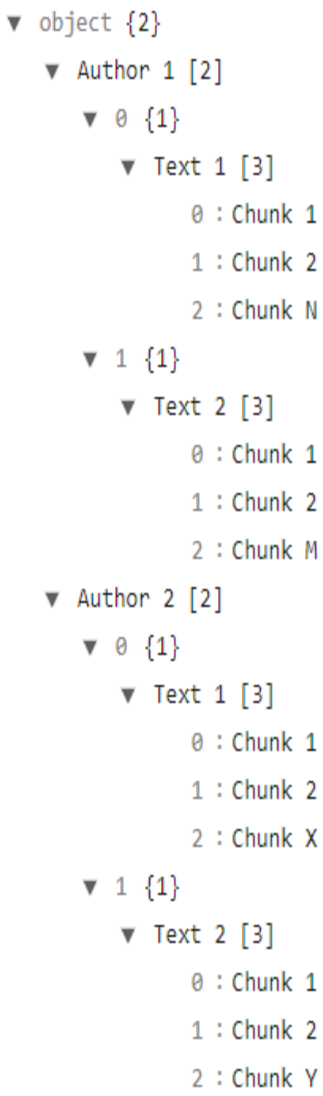
Για το fine tuning μοντέλων όπως το BERT απαιτείται μεγάλος όγκος δεδομένων εφαρμόστηκε η τεχνική του chunking των κειμένων για το data augmentation και τα ζευγάρια επαναπροσδιορίστηκαν από την αρχή.

Για κάθε σώμα δεδομένων από αυτά που χρησιμοποιήθηκαν υπάρχει η πληροφορία για το αν προέρχονται από τον ίδιο συγγραφέα ή όχι (target Label). Επίσης υπάρχει η πληροφορία για τα μοναδικά IDs των συγγραφέων. Έτσι λοιπόν για κάθε συγγραφέα συλλέχθηκαν τα κείμενα του και έπειτα εφαρμόστηκε η τεχνική του chunking. Οπότε προέκυψε μια δομή δεδομένων όπου περιείχε για κάθε συγγραφέα τα διαφορετικά chunks του ανά κείμενο. Ενδεικτικά παρακάτω η μορφή της δομής αυτής.



Εικόνα 11: Δομή Δεδομένων αποθήκευσης των κομματιών κειμένων (Chunks) σε μορφή πίνακα.

Παρακάτω και σε δενδροειδή μορφή για να γίνει καλύτερα κατανοητό.



Εικόνα 12: Δομή Δεδομένων αποθήκευσης των κομματιών κειμένων (Chunks) σε Δενδροειδή μορφή.

Με βάση την παραπάνω εικόνα, για κάθε Author έχουν αποθηκευτεί ξεχωριστά τα chunks για κάθε κείμενο. Δηλαδή δεν συλλέχθηκαν όλα τα chunks για κάθε συγγραφέα σε μια δομή, αντιθέτως κρατήθηκαν ξεχωριστά. Ο λόγος που ακολουθήθηκε αυτός ο τρόπος είναι γιατί κατά την ανάκτηση 2 chunks για την δημιουργία του ζευγαριού δεν θα θέλαμε να παρθούν 2 chunks από το ίδιο κείμενο, έτσι ώστε το μοντέλο να μην είναι biased ως προς το θέμα του κειμένου. Είναι απόλυτα λογικό 2 chunks του ίδιου κειμένου να αναφέρονται στο ίδιο θέμα ιδίως όταν έχει επιλεχθεί η μέθοδος του overlapping. Αυτός ο κίνδυνος αυξάνεται περισσότερο λόγω της χρήσης του BERT, καθώς αυτό το μοντέλο παράγει

Contextualize Embeddings τα οποία περιέχουν και πληροφορία για το νόημα το κειμένου.

Για την δημιουργία λοιπόν των ζευγαριών επιλέγονται κάθε φορά 1 τυχαίο chunk από 2 διαφορετικά κείμενα όπου είναι εφικτό για τον κάθε συγγραφέα. Έτσι δημιουργούμε τα ζευγάρια όπου το target Label τους είναι True. Άρα ανήκουν στον ίδιο συγγραφέα.

Για την δημιουργία των ζευγαριών διαφορετικών συγγραφέων επιλέγονται κάθε φορά 1 τυχαίο chunk από κείμενα διαφορετικών συγγραφέων.

4.4.5 Δημιουργία του Batch

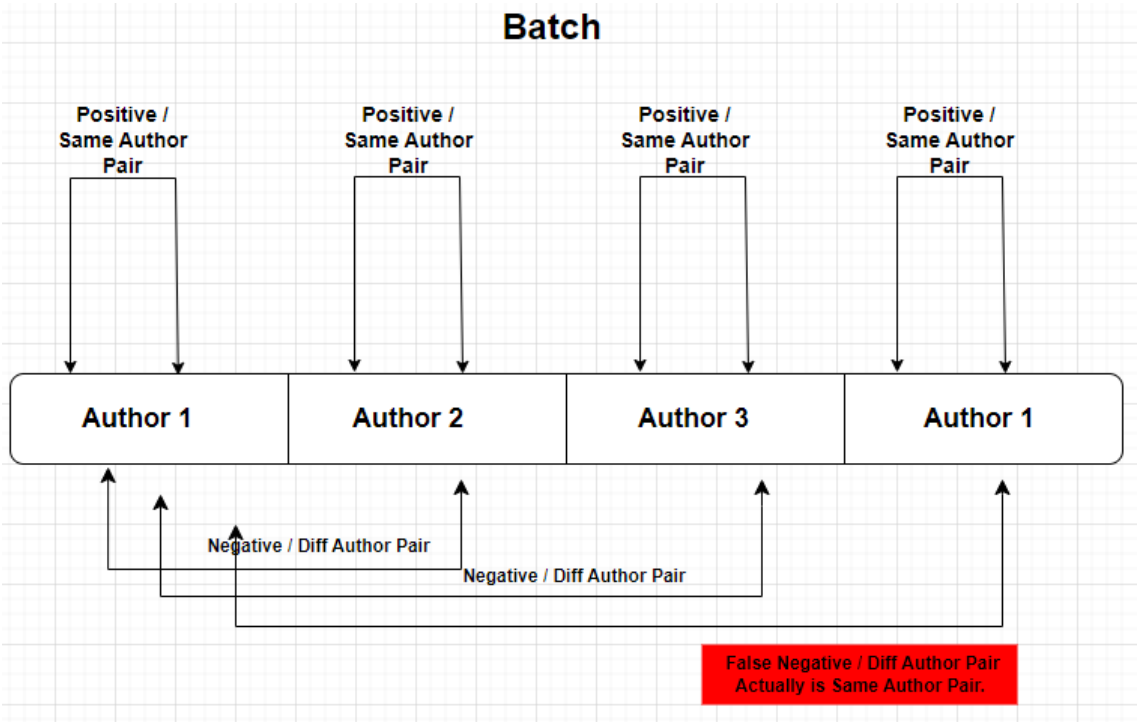
Κατά την διαδικασία του Training και του Validation θα πρέπει να συγκρίνονται ζευγάρια του ίδιου αλλά και διαφορετικών συγγραφέων. Για την δημιουργία τέτοιων ζευγαριών σε ένα batch κάθε φορά βοηθάει η χρήση της Loss.

Η Loss που χρησιμοποιήθηκε στο Contrastive Learning, εμπεριέχει και υπολογίζει το εσωτερικό γινόμενο για κάθε συνδυασμό εντός του Batch.

Το Batch δημιουργείτε αποκλειστικά από ζευγάρια με target Label True. Αυτό που θα πρέπει να σημειωθεί σε αυτό το σημείο και είναι αρκετά σημαντικό είναι ότι δεν μπορούμε να έχουμε στο ίδιο Batch 2 διαφορετικά ζευγάρια του ίδιου συγγραφέα. Είναι κάτι που θα πρέπει να αποφευχθεί για να μην υπάρχουν τα λεγόμενα False Negatives, καθώς όπως προαναφέρθηκε η Loss υπολογίζει το εσωτερικό γινόμενο για όλου τους συνδυασμούς εντός του Batch. Αυτό σημαίνει ότι προκύπτει ένας πίνακας (matrix) NxN όπου η κύρια διαγώνιός του είναι τα ζευγάρια με target Label True και όλα τα υπόλοιπα στοιχεία στον πίνακα έχουν target Label False. Για να γίνει καλύτερα αντιληπτό αυτό ας δώσουμε ένα παράδειγμα:

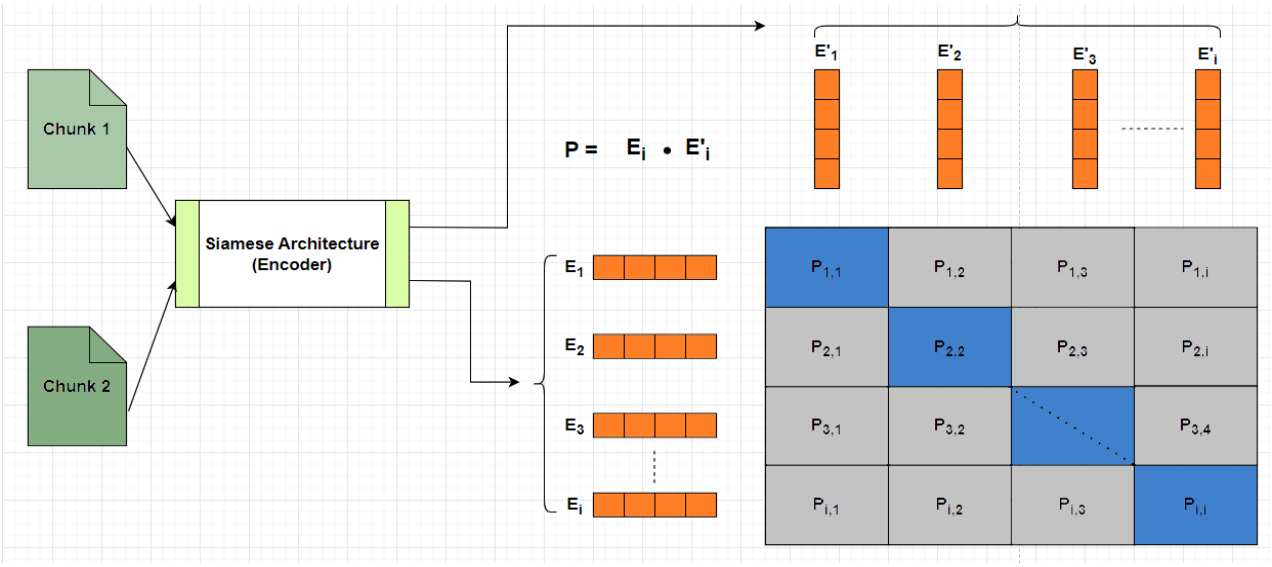
Έστω ότι το Batch Size είναι 64. Άρα θα πρέπει να επιλεχθούν τυχαία 2 chunks από 32 Authors ($32 * 2 = 64$). Ο λόγος που είναι αυστηρός ο τρόπος επιλογής των δεδομένων είναι η μη ύπαρξη False Negative εντός του Batch. Δηλαδή έστω ότι έχουμε σύνολο 50 Authors. Εάν επιλεχθούν όλοι οι Authors σε ένα batch και αρκετά chunks τότε με βάση την παρακάτω εικόνα (Εικόνα 14: Υπολογισμός Dot Product.) και τους υπολογισμούς όλων των συνδυασμών εντός του batch, για ίδιους συγγραφείς, θα έχουμε και ζευγάρι που δεν αντιστοιχεί σε negative (Target Label False), ενώ στην πραγματικότητα θα είναι Positive. Άρα θα υπάρχουν εντός

του batch κάποια False Negatives. Παρακάτω ακολουθεί παράδειγμα ως απεικόνιση ενός Batch για να γίνει κατανοητή η παραπάνω περιγραφή.



Εικόνα 13: Απεικόνιση ενός Batch με False Negative.

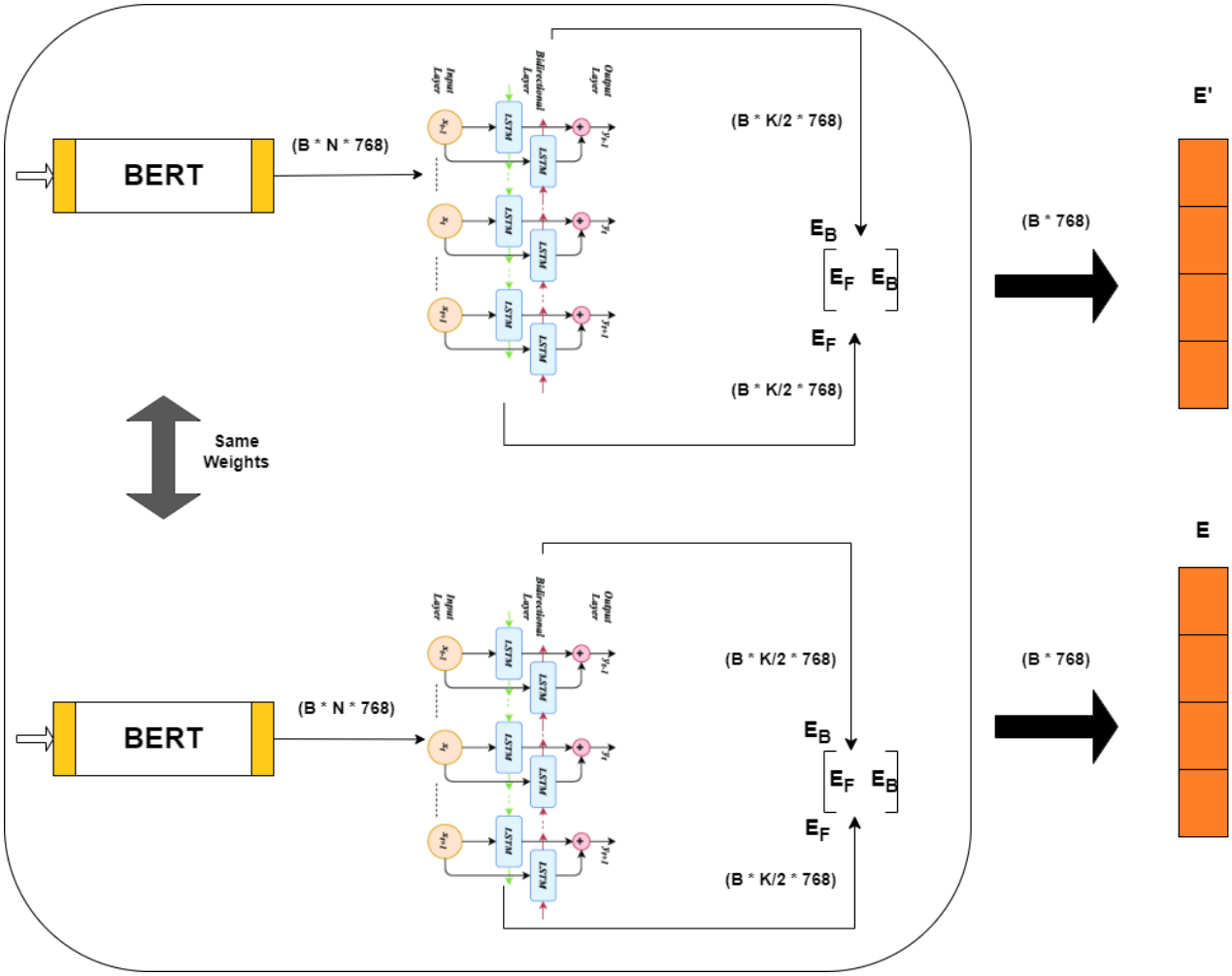
Αφού λοιπόν έχουν δημιουργηθεί σωστά τα batches, ξεκινάει η διαδικασία της εκπαίδευσης. Βασική λειτουργία αυτής της διαδικασίας είναι η loss function και το backpropagation. Η διαδικασία υπολογισμού του εσωτερικού γινομένου για όλους τους συνδυασμούς εντός του batch απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 14: Υπολογισμός Dot Product.

Οτιδήποτε εντός της κύριας διαγώνιου του πίνακα του εσωτερικού γινομένου αποτελεί το αποτέλεσμα της Cosine Similarity για τα ζευγάρια ίδιου συγγραφέα, ενώ στις υπόλοιπες θέσεις του πίνακα έχουμε το αποτέλεσμα της Cosine Similarity για τα ζευγάρια διαφορετικών συγγραφέων.

4.5 Αρχιτεκτονική του Μοντέλου Contrastive Learning

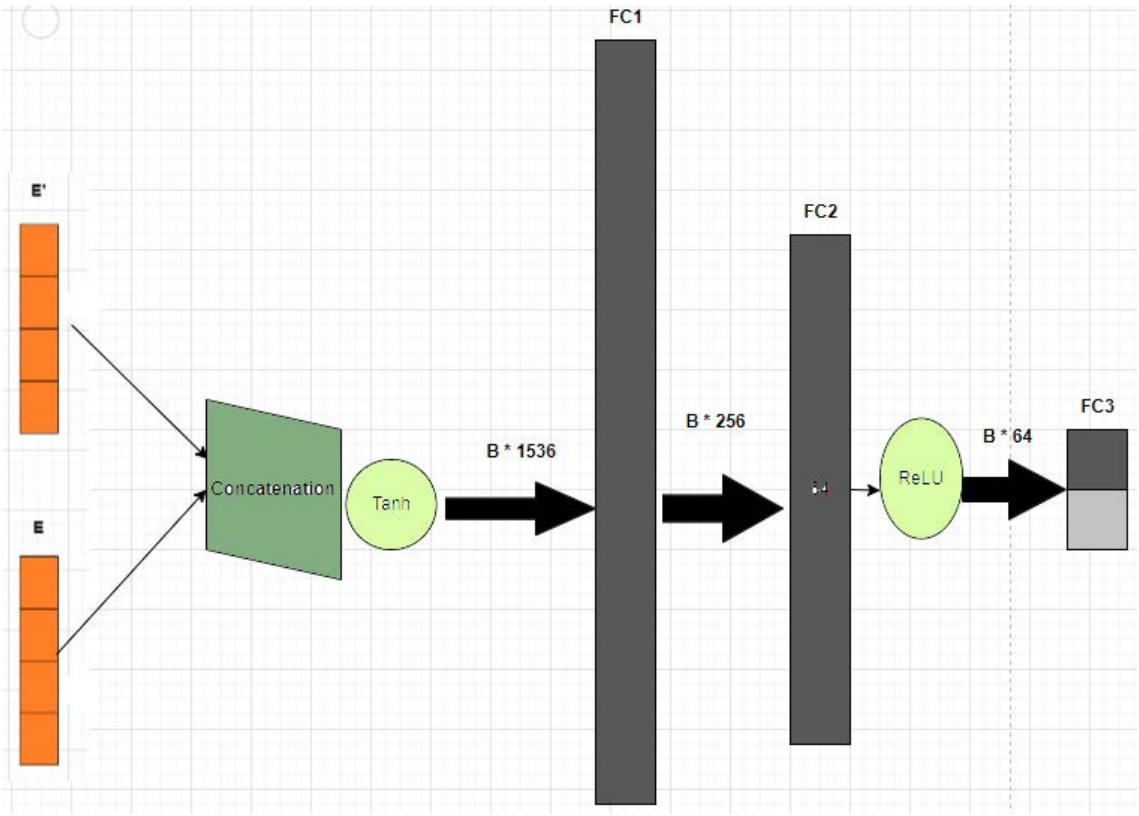


Εικόνα 15: Siamese Αρχιτεκτονική του Μοντέλου που εκπαιδεύτηκε.

Όπως απεικονίζεται στην παραπάνω εικόνα σε ένα ενιαίο μοντέλο υπάρχει 1 αντίγραφο του ίδιου BERT Μοντέλου 2 φορές. Το κάθε υπο-μοντέλο της Siamese αρχιτεκτονικής παράγει Embeddings σε μορφή Vectors (E και E'). Ξεκινώντας από τα BERT, παράγονται τα Embeddings Vectors από το BERT όπου το κάθε

Vector έχει διαστάσεις (Batch Size, N, 768), όπου N το μήκος ακολουθίας που θα δεχτεί στην είσοδό του το μοντέλο. Έπειτα αυτά τα Vectors εισέρχονται σε Bi-LSTM με σκοπό να μάθουν να πιάνουν το στυλ συγγραφής. Από το Bi-LSTM γίνεται εξαγωγή και συνένωση των τελευταίων κρυφών καταστάσεων και από τις δύο κατευθύνσεις και άρα καταλήγουμε να έχουμε ένα Vector με σταθερά Embeddings ως προς τις διαστάσεις (batch size,768) από κάθε υπο-μοντέλο.

Έπειτα από το Contrastive Learning για την ταξινόμηση κάθε ζευγαριού εκπαιδεύτηκε ένα δεύτερο μοντέλο το οποίο λαμβάνει ως είσοδο την συνένωση των Embeddings του μοντέλου από το Contrastive Learning και εκτελεί Classification εργασία. Η αρχιτεκτονική του μοντέλου αυτή παρουσιάζεται παρακάτω. Ο λόγος που χρησιμοποιείται η Tanh στην αρχή, αφορά το γεγονός ότι θέλουμε να διατηρήσουμε τα πρόσημα των Embeddings Vectors που προέρχονται από την διαδικασία του Contrastive Learning.

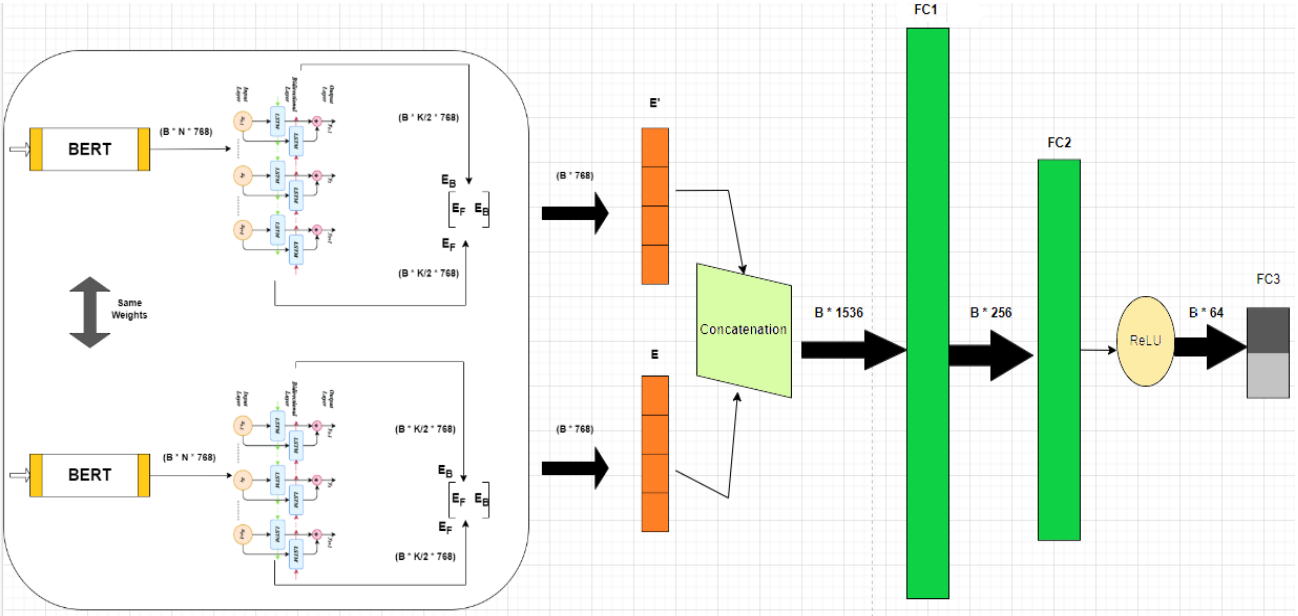


Εικόνα 16: Αρχιτεκτονική Μοντέλου για το Classification μετά από το Contrastive Learning.

Επομένως η διαδικασία της εκπαίδευσης από την αρχή έως το τέλος που αφορά την επίλυση του προβλήματος Authorship Verification ακολουθεί 2 βήματα.

1. Εκπαίδευση μιας Siamese αρχιτεκτονικής με σκοπό να εξάγει κατάλληλα embeddings για τα όμοια και ανόμοια ζευγάρια, χρησιμοποιώντας Contrastive Learning.
2. Εκπαίδευση ενός απλού μοντέλου που δέχεται τα ως είσοδο τα Embeddings του μοντέλου από το βήμα 1 και εκτελεί ένα Classification task με CrossEntropy Loss.

4.6 Αρχιτεκτονική του Classification Μοντέλου



Εικόνα 17: Αρχιτεκτονική Classification Μοντέλου.

Επομένως η διαδικασία της εκπαίδευσης από την αρχή έως το τέλος που αφορά την επίλυση του προβλήματος Authorship Verification σε αυτή την περίπτωση αρχιτεκτονικής ακολουθεί ένα και μόνο βήμα με χρήση CrossEntropy Loss. Εκπαιδεύεται δηλαδή μια Siamese αρχιτεκτονική όπου οι δύο έξοδοί της (E και E') συνενώνονται με σκοπό να εισέλθουν σε κάποια Fully Connected επίπεδα για να εφαρμοστεί η διαδικασία μιας Classification εργασίας.

5 Πειράματα και Αποτελέσματα

5.1 Περιγραφή Δεδομένων

Τα δεδομένα αποτελούνται από κείμενα που έχουν συγγράψει διάφορα άτομα διαφόρων ηλικιών και κοινωνικών ομάδων. Αφού συλλέχθηκαν για τον διαγωνισμό PAN δημιουργήθηκαν με τέτοιο τρόπο ώστε να έχουμε ζευγάρια κειμένων για την εξυπηρέτηση και την επίλυση του προβλήματος Authorship Verification. Στην εκάστοτε συλλογή μπορεί κανείς να δει ζευγάρια διαφορετικών συγγραφέων αλλά και ζευγάρια ίδιων συγγραφέων.

5.1.1 PAN 2015

Για το συγκεκριμένο πρόβλημα του 2015[4] η συλλογή αποτελείται από 4 διαφορετικά σώματα δεδομένων από διαφορετικές γλώσσες (Ολλανδικά, Αγγλικά, Ελληνικά και Ισπανικά) Όπως φαίνεται στον παρακάτω πίνακα. Κάθε σώμα δεδομένων περιέχει 100 διαφορετικά ζευγάρια κειμένων.

LANGUAGE	TYPE	CODE
Dutch	Cross-genre	DU
English	Cross-topic	EN
Greek	Cross-topic	GR
Spanish	Cross-genre	SP

Πίνακας 7: Κατανομή Δεδομένων PAN 2015[4].

5.1.2 PAN 2020

Για το συγκεκριμένο πρόβλημα του 2020[5] η συλλογή αποτελείται από 2 σώματα δεδομένων ένα μεγάλο και ένα μικρό με ζεύγη που αναζητήθηκαν και

[4] https://pan.webis.de/clef15/pan15-web/authorship-verification.html

[5] https://pan.webis.de/clef20/pan20-web/author-identification.html#synopsis

ανακτήθηκαν από το fanfiction.net. Αφού συλλέχθηκαν όλα τα κείμενα επεξεργάστηκαν, με αποτέλεσμα ο μέσος όρος του μήκους κάθε κειμένου να είναι περίπου 21.000 χαρακτήρες. Έτσι δημιουργήθηκαν ζευγάρια κειμένων παρέχοντας όμως πληροφορία και για την θεματική ενότητα (fandom).

```
1. {"id": "6cced668-6e51-5212-873c-717f2bc91ce6", "fandoms": ["Fandom 1", "Fandom 2"], "pair": ["Text 1...", "Text 2..."]}
2. {"id": "ae9297e9-2ae5-5e3f-a2ab-ef7c322f2647", "fandoms": ["Fandom 3", "Fandom 4"], "pair": ["Text 3...", "Text 4..."]}
3. ...
```

Εικόνα 18: PAN 2020 Dataset format.[5]

Για το μεγάλο σύνολο δεδομένων, 148.000 ίδιου συγγραφέα (SA) και 128.000 ζευγάρια διαφορετικών συγγραφέων (DA) επιλέχθηκαν. Τα SA ζευγάρια περιλαμβάνουν 41.000 συγγραφείς από τους οποίους τουλάχιστον 4 και όχι περισσότεροι από 400 έχουν γράψει στο ίδιο fandom. Συνολικά επιλέχθηκαν 1.600 fandoms και κάθε μεμονωμένος συγγραφέας έχει γράψει σε τουλάχιστον 2, αλλά όχι περισσότερα από 6 fandoms. Το μικρό σώμα δεδομένων είναι ένα υποσύνολο του μεγάλου με 28.000 ζεύγη ίδιου συγγραφέα και 25.000 διαφορετικών συγγραφέων από τα ίδια 1.600 fandoms, αλλά με μειωμένο αριθμός συγγραφέων (περίπου 52000).

Το test σώμα δεδομένων αποτελείται από την ίδια μορφή ζευγαριών με του ίδιους συγγραφείς που υπάρχουν στο train set με την διαφορά ότι τα κείμενα διαφέρουν ανάμεσα στο test και το train σώμα δεδομένων. Τέλος το πλήθος των ζευγαριών κατανέμεται ως εξής:

- 10.000 ζευγάρια ίδιου συγγραφέα.
- 6900 ζευγάρια διαφορετικών συγγραφέων.

Με βάση αυτά τα σώματα δεδομένων το πρόβλημα αποτελεί μια Closet-set εργασία καθώς οι συγγραφείς που βρίσκονται στο σώμα δεδομένων εκπαίδευσης εμπεριέχονται και στο test σώμα δεδομένων.

5.1.3 PAN 2021

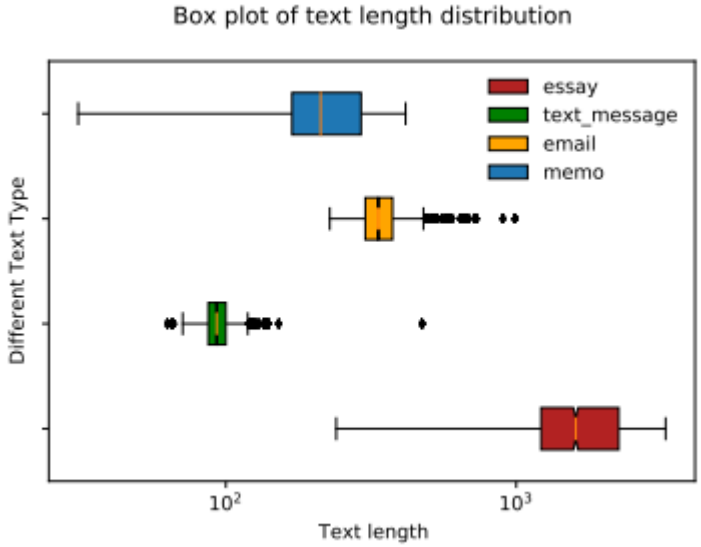
Για το συγκεκριμένο πρόβλημα του 2021 η συλλογή αποτελείται από τα ίδια σώματα δεδομένων με αυτά του 2020 με την διαφορά ότι στο test σώμα δεδομένων έχουμε open-set εργασία, καθώς οι συγγραφείς του σώματος εκπαίδευσης δεν εμπεριέχονται στο test σώμα δεδομένων.

5.1.4 PAN 2022

Το σώμα δεδομένων του 2022[6] αποτελείται από 12264 ζευγάρια. Τα κείμενα προέρχονται από συγγραφείς ηλικίας 18-22 και από 4 διαφορετικά είδη κειμένων:

- Essays
- Emails
- Text messages
- Business memos

Το συγκεκριμένο σώμα δεδομένων είναι αρκετά δύσκολο καθώς τα κείμενα μεταξύ τους διαφέρουν ως προς το θέμα το ύφος το μέγεθος και διάφορα τέτοια χαρακτηριστικά. Παρακάτω παρουσιάζεται ένα Box Plot για το μέσο μήκος χαρακτήρων του εκάστοτε είδους κειμένου.



Εικόνα 19: Box Plot για την σύγκριση των μεγεθών των διαφορετικών τύπων κειμένων[26].

Δεδομένου των χαρακτηριστικών του σώματος δεδομένων αλλά και το μέγεθος αυτού, καθώς είναι αρκετά μικρό, θα πρέπει αν ακολουθηθεί κάποια έξυπνη μεθοδολογία, η οποία θα αξιοποιεί και το είδος του κειμένου.

[6] https://pan.webis.de/clef22/pan22-web/author-identification.html

5.2 Contrastive Learning

Για την διαδικασία του Contrastive Learning χρησιμοποιήθηκαν κάποιοι βασικοί και σταθεροί υπερπαράμετροι σε όλα τα πειράματα όπως παρουσιάζονται στον παρακάτω πίνακας. Τα καλύτερα αποτελέσματα επιτεύχθηκαν χρησιμοποιώντας την modified contrastive loss.

Dataset	Optimizer	Negative Margin	Positive Margin	Batch Size	Chunk Size
PAN 2015	AdamW	0	1	32	128
PAN 2020	AdamW	0	1	64	512
PAN 2021	AdamW	0	1	64	512
PAN 2022	AdamW	0	1	16	128 στα text_messages και emails και 256 στα essays και memos.

Πίνακας 8: Βασικοί υπερπαράμετροι για το Contrastive Learning.

Όλα τα πειράματα εκτελέστηκαν στο Google Colab με την ισχυρότερη GPU (NVIDIA V100, 40GB vRAM). Ο κώδικας είναι διαθέσιμος στο github[7]

Πριν ξεκινήσει η διαδικασία έγινε διαχωρισμός των δεδομένων σε Train και Validation σώματα δεδομένων, έτσι ώστε να έχουμε Open-set εργασία.

Συγκεκριμένα:

- PAN 2015: Επειδή το σώμα δεδομένων ήταν μικρό τα ποσοστά διαχωρισμού σε train και validation είναι 85% και 15% αντίστοιχα.
- PAN 2020 & 2021: Αντίστοιχα τα ποσοστά είναι 80% και 20%
- PAN 2022: Επειδή το σώμα δεδομένων ήταν μικρό τα ποσοστά διαχωρισμού σε train και validation είναι 85% και 15% αντίστοιχα.

[7] https://github.com/icsd13152/Thesis_MSC_in_AI_AuthorVerif

Για λόγους υπολογιστικής ισχύος και κόστους στο Google Colab[8] ήταν εξαιρετικά χρονοβόρο να εκπαιδευτεί κάποιο μοντέλο με όλα τα δεδομένα στο σώμα δεδομένων PAN 2020 και ειδικά στις περιπτώσεις όπου δεν έγινε κάποιο freeze. Για αυτόν τον λόγο, η δυσκολία αυτή μετατράπηκε σε μια ιδέα για επιπλέον πειράματα ως προς το πλήθος των δεδομένων που θα ληφθούν υπόψιν. Έπειτα από την δημιουργία των Batches με τον τρόπο που παρουσιάζεται στο κεφάλαιο 4.4.5 υπήρχε η δυνατότητα αύξησης ή μείωσης των συγγραφέων που θα λάβουν μέρος στο training.

Κατά την διαδικασία της εκπαίδευσης και της αξιολόγησης έγιναν πάρα πολλά πειράματα χρησιμοποιώντας τα προ-εκπαιδευμένα γλωσσικά μοντέλα *BERT-base-uncase, BERT-base-cased* και *Roberta-base*.

Επιγραμματικά τα πειράματα που εκτελέστηκαν παρουσιάζονται στους παρακάτω πίνακες:

ID	Μεθοδολογία PAN 2015 με χρήση BERT Uncased	Epochs
1	Freeze μέχρι το 8ο Encode Layer του BERT και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.	1. Siamese αρχιτεκτονική Contrastive Learning: 3 εποχές. 2. Δεύτερο βήμα εκπαίδευσης Classification: 10 εποχές.
2	Freeze όλων των Encode Layers του BERT και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Τα embeddings έγιναν εξαγωγή από το Last Hidden Layer του BERT.	1. Siamese αρχιτεκτονική Contrastive Learning: 5 εποχές. 2. Δεύτερο βήμα εκπαίδευσης Classification: 13 εποχές.

Πίνακας 9: Βασική λίστα πειραμάτων PAN 2015 BERT Uncased.

[8] https://colab.research.google.com/drive/1q4Vusbi5dtd5wM0Y0809iooZZK0l70PQ#scrollTo=oVfLShVA7VEo

Για το σώμα δεδομένων του PAN 2015 πειραματικά προέκυψε ότι το καλύτερο learning rate για τον AdamW optimizer είναι το 0.00002 (2e-5). Επίσης Έγινε χρήση μόνο των αγγλικών κειμένων.

ID	Μεθοδολογία PAN 2022 με χρήση BERT Uncased
1	Unfreeze όλου του BERT, αντικατάσταση των αριθμών με συγκεκριμένο ψηφίο (1), αντικατάσταση των Emoticons/Emojis με ένα συγκεκριμένο ψηφίο (2), μετατροπή των πιο συχνών συντομογραφιών στην μορφή όπου έχουμε όλες τις λέξεις. Εξαγωγή Embeddings από την συνένωση των CLS (πρώτου token) Embeddings από το 3ο έως 12ο Encode Layer.

Πίνακας 10: Βασική λίστα πειραμάτων PAN 2022 BERT Uncased.

Για το σώμα δεδομένων του PAN 2022 πειραματικά προέκυψε ότι το καλύτερο learning rate για τον AdamW optimizer είναι το 0.00005 (5e-5). Επίσης επειδή το συγκεκριμένο σώμα δεδομένων είναι αρκετά δύσκολο, καθώς εμπεριέχει κείμενα από διαφορετικά είδη κειμένων (text messages,memos,essays,emails), αποφασίστηκε να «σπάσει» σε μικρότερα σώματα με την παρακάτω λογική:

- Ένα σώμα δεδομένων που αποτελείται μόνο από email και text messages, καθώς είναι περίπου ίδια σε μέγεθος και εμπεριέχουν περίπου ίδιου τύπου ειδικών χαρακτήρων, συμβόλων και λέξεων όπως τα Emoticons/Emojis και Slang λέξεις.
- Ένα σώμα δεδομένων που αποτελείται μόνο από memos και essays, καθώς είναι περίπου ίδια σε μέγεθος.

Επομένως έχουμε 2 διαφορετικά μοντέλα και για την τελική απόφαση στο test σώμα δεδομένων εκτελούνται ανάλογα με τους τύπους κειμένου που θα εισέλθουν στην είσοδο και για τα κείμενα για τα οποία τα μοντέλα δεν γνωρίζουν συνδυασμό (π.χ. essay vs. emails) προκύπτει ο μέσος όρος των προβλέψεων αυτών των μοντέλων. Δηλαδή μια διαδικασία Ensemble.

Σε αντίθεση με τα υπόλοιπα σώματα δεομένων για τα PAN 2020 και 2021 εκτελέστηκαν αρκετά πειράματα καθώς είναι μεγάλα σώματα δεδομένων και έτσι μπορούμε να βγάλουμε καλύτερο συμπέρασμα για την μεθοδολογία σε συνδυασμό με τα άλλα σώματα.

Το learning rate για όλα τα πειράματα ήταν το 0.00002 (2e-5).

Τα παρακάτω προ-εκπαιδευμένα μοντέλα χρησιμοποιήθηκαν.

- BERT Uncased
- BERT Cased
- RoBERTa

Για όλα τα παρακάτω πειράματα 1 εποχή στο training του Contrastive learning ήταν αρκετή να μειώσει κατά πολύ την τιμή της loss.

ID	Μεθοδολογία PAN 2020 & 2021 με χρήση BERT uncased
1	Freeze όλων των Encode Layers του BERT κρατώντας λιγότερο από τους μισούς συγγραφείς (περίπου 11.000 από τους 42000 του train) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Τα embeddings έγιναν εξαγωγή από το Last Hidden Layer του BERT.
2	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας τους μισούς συγγραφείς (περίπου 21000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.
3	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 28000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.
4	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 30000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.
5	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 35000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.
6	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 30000) και αντικατάσταση των αριθμών με συγκεκριμένο ψηφίο και των Named Entities με συγκεκριμένες ονοματολογίες. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.
7	Freeze όλων των Encode Layers του BERT κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 35000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Τα embeddings έγιναν εξαγωγή από το Last Hidden Layer του BERT.

Πίνακας 11: Βασική λίστα πειραμάτων PAN 2020 & 2021 BERT Uncased.

ID	Μεθοδολογία PAN 2020 & 2021 με χρήση BERT cased
1	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 35000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.

Πίνακας 12: Βασική λίστα πειραμάτων PAN 2020 & 2021 BERT Cased.

ID	Μεθοδολογία PAN 2020 & 2021 με χρήση RoBERTa
1	Freeze μέχρι το 8ο Encode Layer του RoBERTa κρατώντας πάνω από τους μισούς συγγραφείς (περίπου 30000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.

Πίνακας 13: Βασική λίστα πειραμάτων PAN 2020 & 2021 RoBERTa.

5.3 Απλό Classification Task

Σε αυτού του είδους την μορφή εκπαίδευσης το πείραμα εκτελέστηκε για 5 εποχές.

ID	Μεθοδολογία PAN 2020 & 2021 με χρήση BERT uncased
1	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας τους μισούς συγγραφείς (περίπου 30000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.

Πίνακας 14: Βασική λίστα πειραμάτων Classification PAN 2020 & 2021 BERT Uncased.

Στο παρακάτω πείραμα το μοντέλων εκπαιδεύτηκε για 7 εποχές.

ID	Μεθοδολογία PAN 2015 με χρήση BERT uncased
1	Freeze μέχρι το 8ο Encode Layer του BERT κρατώντας τους μισούς συγγραφείς (περίπου 21000) και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers. Chink size 128.

Πίνακας 15 : Βασική λίστα πειραμάτων Classification PAN 2015 BERT Uncased.

5.4 Evaluation

Για τη αξιολόγηση των μοντέλων ακολουθήθηκε η λογικά του Open-set, δηλαδή οι συγγραφείς του test σώματος δεδομένων δεν περιέχονται στο Train. Κατά αυτόν τον τρόπο πριν ξεκινήσει η διαδικασία έγινε διαχωρισμό των δεδομένων σε Train και Validation σώματα δεδομένων, έτσι ώστε να έχουμε Open-set εργασία.

Συγκεκριμένα:

- PAN 2015: Επειδή το σώμα δεδομένων ήταν μικρό τα ποσοστά διαχωρισμού σε train και validation είναι 85% και 15% αντίστοιχα.
- PAN 2020 & 2021: Αντίστοιχα τα ποσοστά είναι 80% και 20%
- PAN 2022: Επειδή το σώμα δεδομένων ήταν μικρό τα ποσοστά διαχωρισμού σε train και validation είναι 85% και 15% αντίστοιχα.

Τα chunks του κάθε συγγραφέα σε κάθε εποχή επιλέγονται τυχαία με τέτοιο τρόπου που να μην ανακτηθούν 2 chunks (1 ζευγάρι) που ανήκουν στο ίδιο κείμενο. Η παραπάνω διαδικασία ακολουθήθηκε διότι δεν θα έπρεπε το εκάστοτε μοντέλο να μάθει αναπαραστάσεις που αφορούν το θέμα του κειμένου. Επειδή χρησιμοποιούνται Transformers οι οποίοι παράγουν Contextualized Embeddings το να μάθει το μοντέλο να προβλέπει με βάση το θέμα και όχι το στυλ συγγραφής δεν θα μπορούσε να επιφέρει καλά αποτελέσματα στο πρόβλημα του Authorship Verification. Σημαντικό επίσης είναι να αναφερθεί ότι η επιλογή της Open set εργασίας ακολουθήθηκε για να δυσκολευτεί το μοντέλο περισσότερο. Για κάθε μοντέλο εκτελέστηκαν 3 runs στο test set με σκοπό να είμαστε σε θέση να γνωρίζουμε και την τυπική απόκλιση (std) του Accuracy και του F1 score. Οι τιμές των τυπικών αποκλίσεων αναγράφονται στο κάθε πίνακα με την μορφή (+/- XXX).

Τα αποτελέσματα των πειραμάτων επιτεύχθηκαν στα αντίστοιχα test set του εκάστοτε διαγωνισμού. Με βάση αυτό, τα κείμενα κάθε ζευγαριού στο test set χωρίστηκαν σε chunks και για την τελική απόφαση υπολογίζονται τα averaged Embeddings του κάθε chunk για κάθε κείμενο του ζευγαριού. Τελικώς έχουμε text Embeddings. Ένας επιπλέον τρόπος είναι η σύγκριση όλων των συνδυασμών των chunks μεταξύ τους ανά ζευγάρια και ως τελικό αποτέλεσμα είναι το average αυτών. Αυτό εννοείται εφαρμόζεται στο Contrastive Learning όπου θα έχουμε τόσα cosine similarities scores όσοι και οι συνδυασμοί των ζευγαριών από

chunks. Επομένως για την τελική απόφαση στο τεστ σώμα δεδομένων ακολουθούμε 1 βασικό βήμα:

1. Chunking των κειμένων διατηρώντας όμως τα αρχικά ζευγάρια.

Έπειτα μπορεί κανείς να επιλέξει έναν από τους παρακάτω βασικούς τρόπους της τελικής απόφασης:

1. Παραγωγή Embeddings για κάθε chunk από το κάθε κείμενο ενός ζευγαριού. Υπολογισμό Average Embeddings Vector από τα chunks του κάθε κειμένου, οδηγώντας σε Average Embeddings Vector του κάθε κειμένου από το ζευγάρι. Άρα 2 Vectors για κάθε κείμενο. Τέλος εφαρμογή Cosine Similarity για το τελικό prediction.
2. Παραγωγή Embeddings για κάθε chunk από το κάθε κείμενο ενός ζευγαριού. Υπολογισμό Cosine Similarity για κάθε chunk του πρώτου κειμένου με όλα τα chunks του δεύτερου κειμένου ενός ζευγαριού. Τέλος υπολογισμό Average Cosine Similarity.
3. Συνδυασμός των 1 και 2.

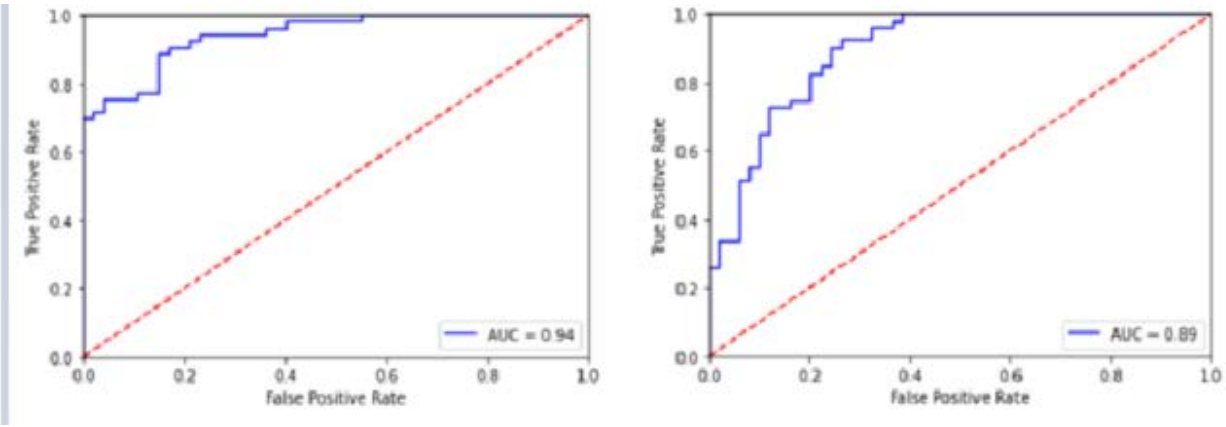
Εννοείται αντί για Average στο νούμερο 1 μπορεί κανείς να υπολογίσει το άθροισμα των Embeddings. Αυτό όμως θέλει προσοχή καθώς μπορεί να οδηγήσει σε μεγάλα αθροίσματα ή ένας συγκεκριμένος Vector να αποτελείται από μεγάλους αριθμούς, με αποτέλεσμα να αλλοιώσει το τελικό prediction της Cosine Similarity. Ένας πιθανός τρόπος αντιμετώπισης είναι το normalization των Embeddings Vectors πριν τον υπολογισμό του αθροίσματος, με σκοπό να μορφοποιηθούν στην ίδια κλίμακα.

5.5 Αποτελέσματα Contrastive Learning

Τα αποτελέσματα που παρουσιάζονται παρακάτω επιτεύχθηκαν στα Test σώματα δεδομένων. Σε αυτά δηλαδή που δεν λήφθηκαν υπόψιν κατά την διαδικασία του Train και Validation.

5.5.1 Αποτελέσματα PAN 2015

Με βάση τα πειράματα του πίνακα (Πίνακας 9) τα αποτελέσματα παρουσιάζονται παρακάτω (αριστερά το πείραμα του πίνακα με ID 1 και δεξιά το πείραμα με ID 2):



Εικόνα 20: AUC & ROC Curves PAN 2015.

ID	Accuracy (%)	AUC score (%)	F1-score (%)
1	**87.2 (+/-0.02)**	**94**	**85.6 (+/-0.03)**
2	82.06 (+/- 0.03)	89	82.9 (+/- 0.03)

Πίνακας 16: Accuracy, AUC, F1-score PAN 2015.

Στο σώμα δεδομένων του 2015 καλύτερο αποτέλεσμα επιτεύχθηκε με τις παρακάτω βασικές ενέργειες:

1. Freeze μέχρι το 8ο Encode Layer του BERT uncased και αντικατάσταση μόνο των αριθμών με συγκεκριμένο ψηφίο. Εξαγωγή Embeddings από το άθροισμα των 4 τελευταίων Encode Layers.
2. Μήκος chunk 128 καθώς πρόκειται για μικρά κείμενα.
3. Μέγεθος Batch 32.
4. Χρήση της προσέγγισης με 2 φάσεις στην διαδικασία της εκπαίδευσης:
    a. Siamese αρχιτεκτονική Contrastive Learning: 3 εποχές.
5. Δεύτερο βήμα εκπαίδευσης Classification: 10 εποχές.

Στον παρακάτω πίνακα παρουσιάζονται το AUC score της παρούσας διπλωματικής εργασίας σε σύγκριση με κάποιες προσεγγίσεις όπου πέτυχαν πάρα πολύ καλά αποτελέσματα.

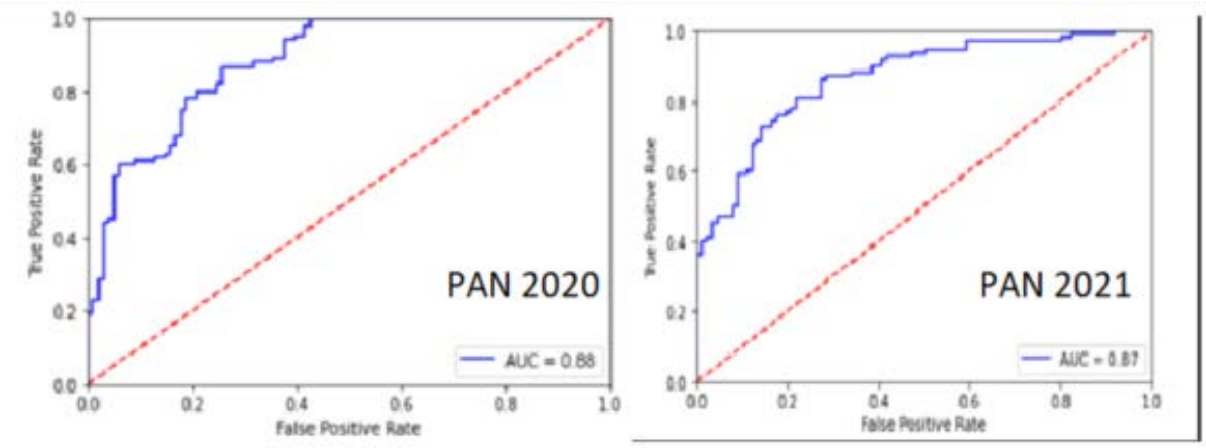
Approach	**AUC PAN 2015**
Enhanced Impostors Method[3]	79.8
Bagnall[9]	81.1
Δική μου προσέγγιση	**94**

Πίνακας 17: Σύγκριση με την καλύτερη προσέγγιση του 2015 (Αγγλικά κείμενα).

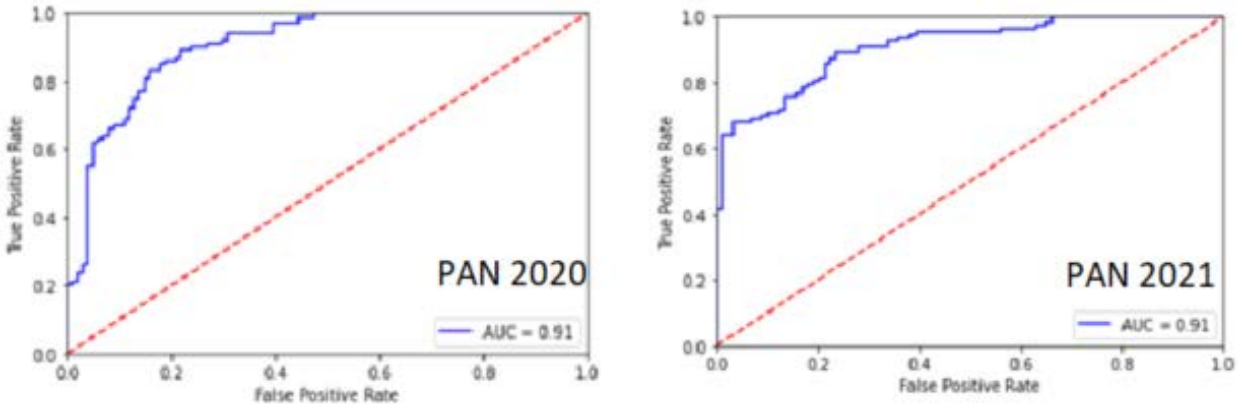
Με βάση τα αποτελέσματα τα καλύτερα scores ξεπεράστηκαν από την μεθοδολογία της παρούσας διπλωματικής εργασίας και μάλιστα κατά αρκετά μεγάλο ποσοστό. Αυτό μας οδηγεί στο συμπέρασμα ότι η χρήση του BERT και Bi-LSTM σε Contrastive Learning τεχνική μπορεί να επιφέρει πάρα πολύ καλά αποτελέσματα ακόμα και με λίγα δεδομένα. Ακόμα και το δεύτερο πείραμα για αυτό το Dataset που πραγματοποιήθηκε για τις ανάγκες της παρούσας διπλωματικής κατάφερε να ξεπεράσει τις καλύτερες προσεγγίσεις.

5.5.2 Αποτελέσματα PAN 2020 & PAN 2021

Παρακάτω ακολουθούν τα πειράματα για τα σώματα δεδομένων του 2020 και 2021 με βάση τα ID πειραμάτων του πίνακα (Πίνακας 11).

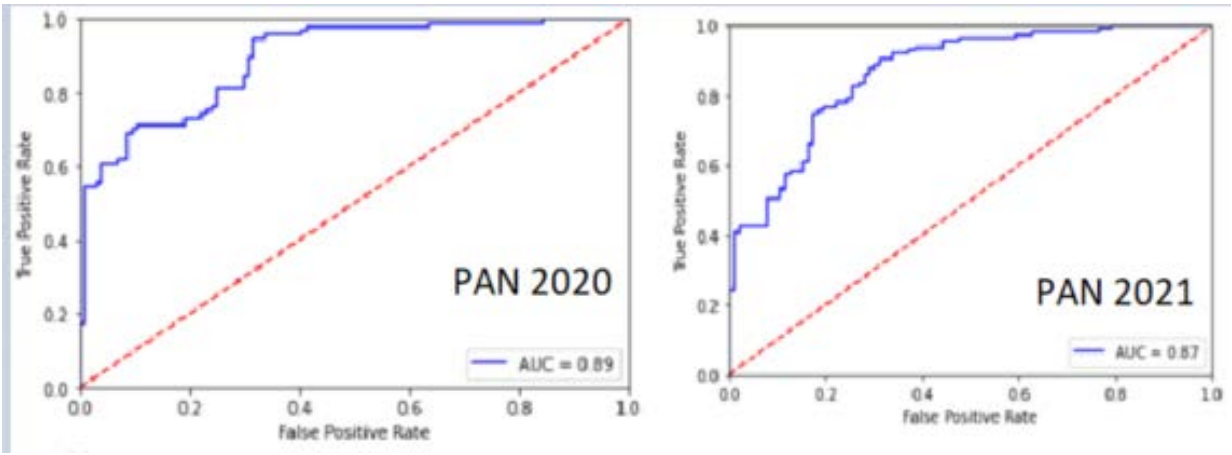


Εικόνα 21: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 1.

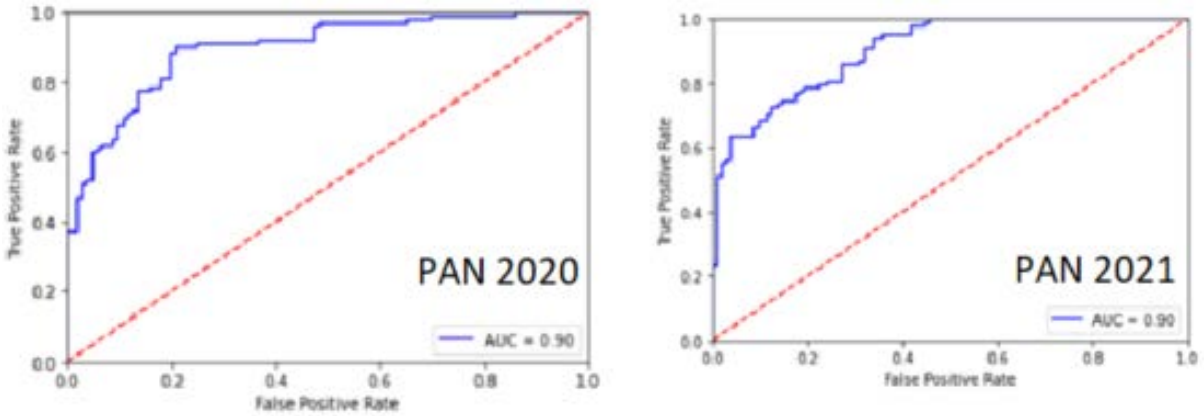


Εικόνα 22: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 7.

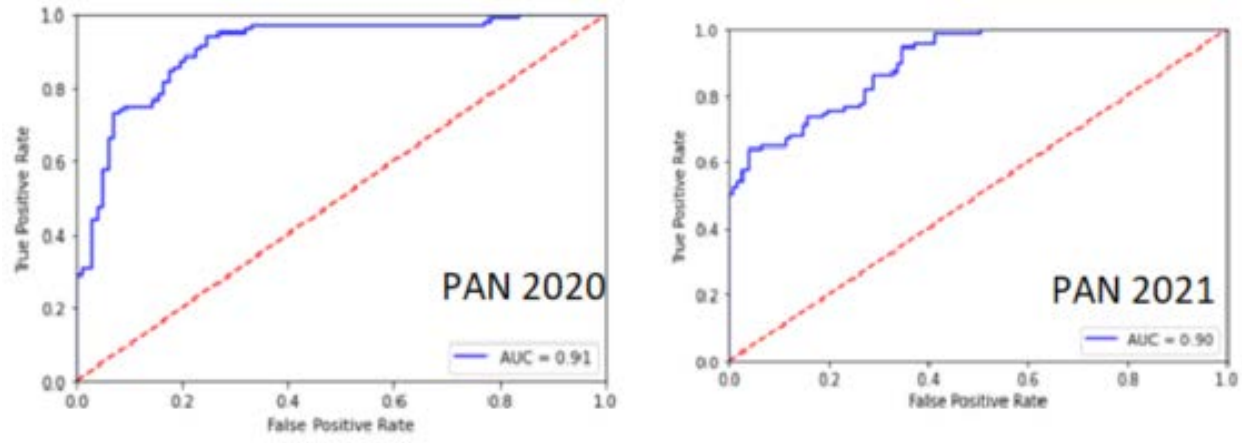
Σε αντίθεση με τα παραπάνω πειράματα όπου το BERT ήταν freeze σε όλα τα επίπεδα και άρα η διαδικασία του back propagation και η βελτιστοποίηση των βαρών δεν εκτελούνταν για τα επίπεδα του BERT, στα παρακάτω αποτελέσματα τα πειράματα έγινα με freeze του BERT μέχρι το 8ο Encode Layer. Δηλαδή για τα 4 τελευταία επίπεδα του BERT ακολουθείται η διαδικασία του back propagation.



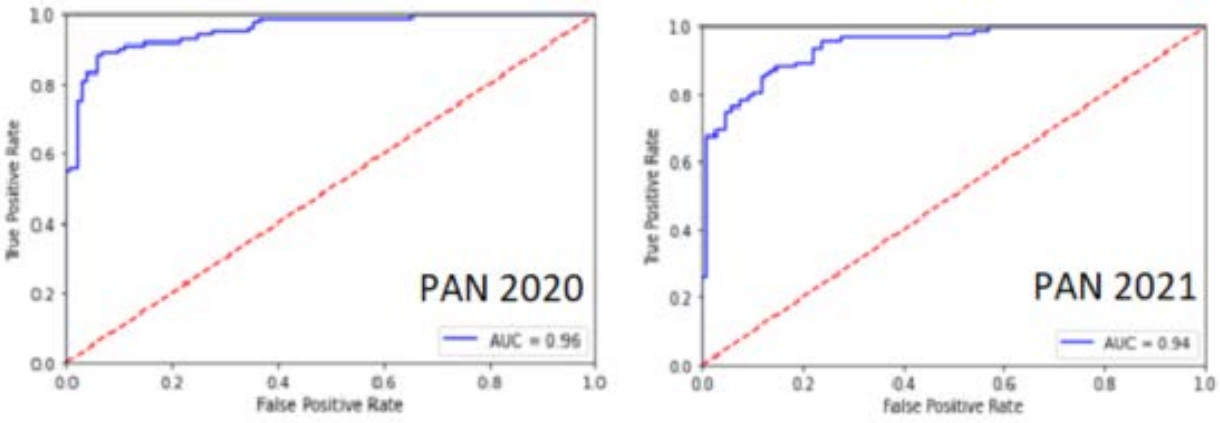
Εικόνα 23: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 2.



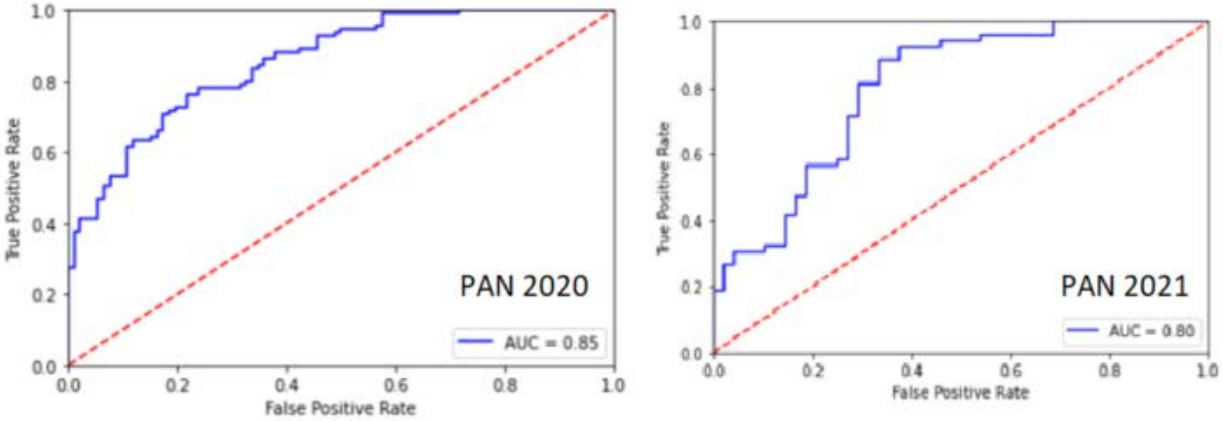
Εικόνα 24: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 3.



Εικόνα 25: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 4.



Εικόνα 26: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 5.



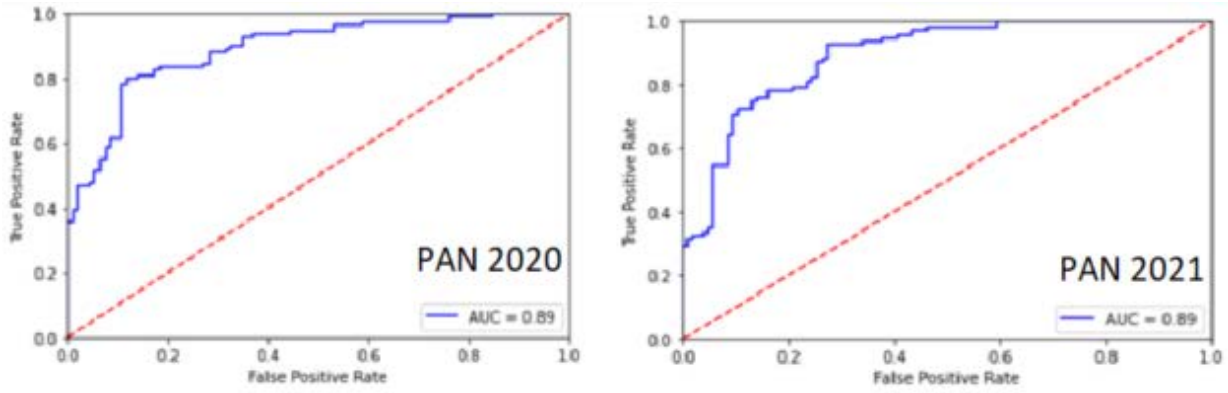
Εικόνα 27: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 6.

ID	Accuracy PAN 2020	Accuracy PAN 2021	AUC PAN 2020	AUC PAN 2021	F1-Score PAN 2020	F1-Score PAN 2021
1	81.2 (+/- 0.03)	81 (+/- 0.03)	88	87	80.6 (+/- 0.03)	79.8 (+/- 0.04)
2	81.7 (+/- 0.02)	81.2 (+/- 0.03)	89	87	81.2 (+/- 0.03)	80.4 (+/- 0.03)
3	85.7 (+/- 0.02)	84 (+/- 0.02)	90	90	83.8 (+/- 0.02)	82.7 (+/- 0.02)
4	86.1	85.1	91	90	84.2	82.8

	(+/- 0.02)	(+/- 0.02)			(+/- 0.02)	(+/- 0.03)
5	**92 (+/- 0.02)**	**91.4 (+/- 0.02)**	**96**	**94**	**92 (+/- 0.03)**	**89.8 (+/- 0.03)**
6	78.8 (+/- 0.04)	76.8 (+/- 0.04)	85	80	78.4 (+/- 0.04)	73.7 (+/- 0.04)
7	85.7 (+/- 0.02)	85.6 (+/- 0.03)	91	91	84.8 (+/- 0.03)	83.2 (+/- 0.03)

Πίνακας 18: Συγκεντρωτικά αποτελέσματα PAN 2020 & 2021 BERT Uncased.

Το πείραμα όπου με την χρήση του BERT Uncased επέφερε το καλύτερο αποτέλεσμα (ID 5 με 92% και 91.4% Accuracy στα PAN 2020 & 2021 αντίστοιχα) εκτελέστηκε με τον ίδιο τρόπο με την διαφορά ότι χρησιμοποιήθηκε το BERT Cased (Πίνακας 12).



Εικόνα 28: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 1.

ID	**Accuracy PAN 2020**	**Accuracy PAN 2021**	**AUC PAN 2020**	**AUC PAN 2021**	**F1-Score PAN 2020**	**F1-Score PAN 2021**
1	81.6 (+/- 0.03)	80.9 (+/- 0.04)	89	89	81.2 (+/- 0.04)	80.3 (+/- 0.04)

Πίνακας 19: Συγκεντρωτικά αποτελέσματα PAN 2020 & 2021 BERT Cased.

Με βάση τα παραπάνω αποτελέσματα μπορεί κανείς να διαπιστώσει ότι η μεθοδολογία που ακολουθήθηκε μπορεί να φέρει καλύτερα αποτελέσματα με την χρήση της Uncased έκδοσης του BERT.

Επίσης η προσέγγιση της παρούσας διπλωματικής εργασίας, κατάφερε να προσεγγίσει τις καλύτερες επιδόσεις για τα σώματα δεδομένων του 2020 και 2021 αντίστοιχα. Αναλυτικά στον παρακάτω πίνακα.

Approach	**AUC PAN 2020**	**F1-Score PAN 2020**
boenninghoff20-large[13]	96.9	93.6
Δική μου προσέγγιση	96	92

Πίνακας 20: Σύγκριση με την καλύτερη προσέγγιση του 2020.

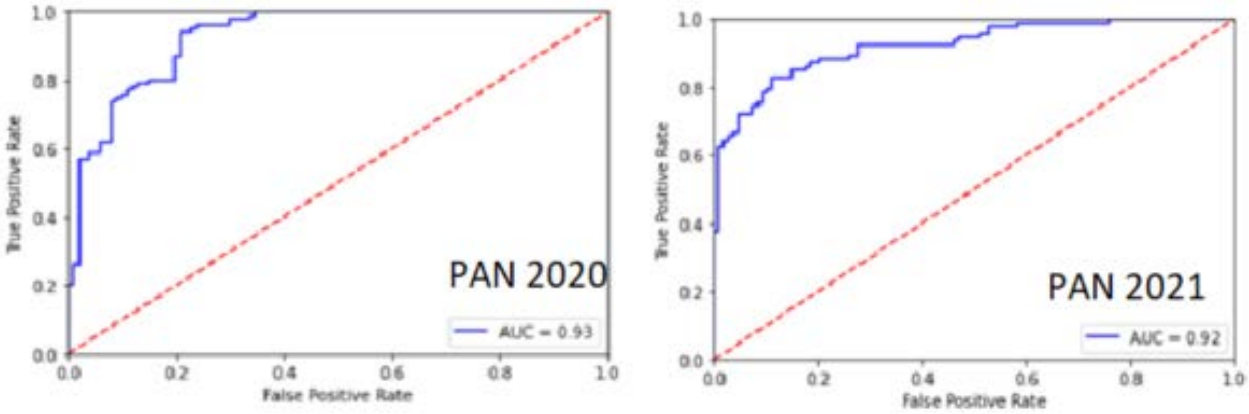
Με βάση τον παραπάνω πίνακα μπορεί κανείς να διαπιστώσει ότι προσεγγίστηκε η καλύτερη επίδοση του 2020. Η παρούσα όμως διπλωματική εργασία δεν έκανε χρήση του large σώματος δεδομένων, αλλά του small. Επίσης από το small που χρησιμοποιήθηκε δεν έγινε η χρήση όλου του συνόλου δεδομένων αλλά σχεδόν τα 2/3 αυτού, λόγω τεχνικών προβλημάτων και κόστους στο Google colab. Επομένως, δεδομένου αυτών των συνθηκών θα μπορούσε κανείς να συμπεράνει ότι η μεθοδολογία της παρούσας διπλωματικής εργασίας κατάφερε να ξεπεράσει την καλύτερη επίδοση στον διαγωνισμό του 2020.

Approach	AUC PAN 2021	F1-Score PAN 2021
boenninghoff21[15]	98.69	95.24
Δική μου προσέγγιση	94	89.8

Πίνακας 21: Σύγκριση με την καλύτερη προσέγγιση του 2021.

Ομοίως και για το 2021 προσεγγίστηκαν τα καλύτερα scores πάλι με την χρήση του small σώματος δεδομένων σε αντίθεση με την καλύτερη επίδοση του διαγωνισμού που χρησιμοποίησε το large. Αξίζει να σημειωθεί ότι το ίδιο μοντέλο της παρούσας διπλωματικής εργασίας που εκπαιδεύτηκε για το dataset του 2020, το ίδιο χρησιμοποιήθηκε και για το 2021. Δεν χρειάστηκε κάποια μετεκπαίδευση ή fine-tuning. Αντιθέτως η καλύτερη επίδοση του 2020 έλαβε μέρος στον διαγωνισμό του 2021 με άλλο μοντέλο και άλλη αρχιτεκτονική για να ξανά πάρει την πρώτη θέση εκπαιδεύοντας το μοντέλο στο large dataset. Επομένως, δεδομένου των παραπάνω μπορεί κανείς να διαπιστώσει ότι η μεθοδολογία της παρούσας διπλωματικής εργασίας σε αυτά τα δύο σώματα δεδομένων είναι ανταγωνιστική σε σχέση με την καλύτερη προσέγγιση του 2020 και 2021 αντίστοιχα.

Έπειτα από τα πειράματα με την χρήση των 2 εκδόσεων του BERT, πραγματοποιήθηκε και πείραμα με χρήση του μοντέλου RoBERTa για να δούμε κατά πόσο μπορεί να επιλύσει το πρόβλημα εξίσου καλά. Το πείραμα που εκτελέστηκε παρουσιάζεται στον πίνακα (Πίνακας 13) και έτρεξε με λιγότερα δεδομένα από ότι το καλύτερο πείραμα του BERT Uncased.



Εικόνα 29: AUC & ROC Curve PAN 2020 & 2021 για πείραμα με ID 1.

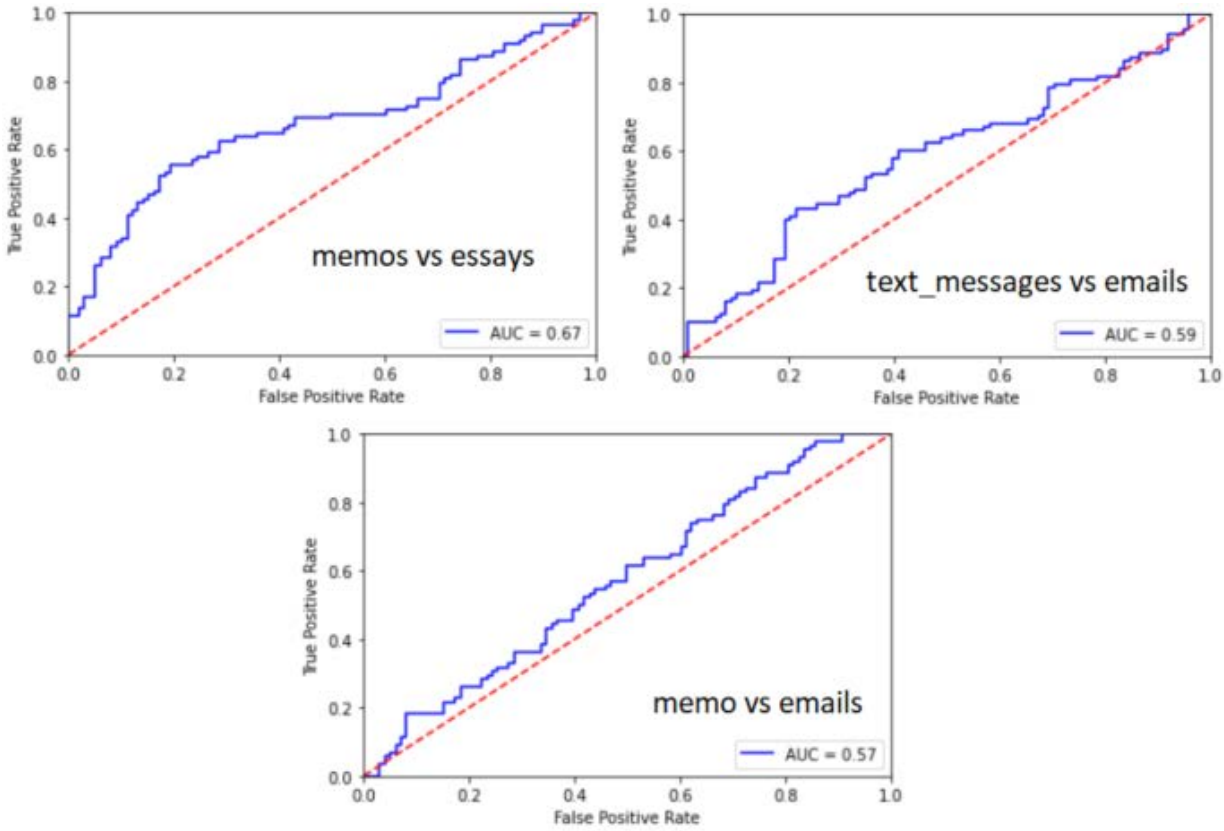
ID	Accuracy PAN 2020	Accuracy PAN 2021	AUC PAN 2020	AUC PAN 2021	F1-Score PAN 2020	F1-Score PAN 2021
1	**90 (+/- 0.02)**	**88.6 (+/- 0.02)**	**93**	**92**	**91 (+/- 0.02)**	**88.2 (+/- 0.03)**

Πίνακας 22: Συγκεντρωτικά αποτελέσματα PAN 2020 & 2021 RoBERTa.

Αξιοσημείωτο είναι ότι με την χρήση του RoBERTa μπορεί να προσεγγιστεί η καλύτερη επίδοση του 2020 και 2021 με ακόμα πιο λίγα δεδομένα από το small dataset. Αυτό μας οδηγεί στο συμπέρασμα ότι το RoBERTa ως μοντέλο μπορεί να επιφέρει καλύτερα αποτελέσματα σε σχέση με το BERT.

5.5.3 Αποτελέσματα PAN 2022

Έπειτα από το Contrastive Learning δεν εφαρμόστηκε κάποιο training για Classification στο συγκεκριμένο σώμα δεδομένων, καθώς πειραματικά δεν επέφερε καλά αποτελέσματα. Αντιθέτως εφαρμόστηκε η τεχνική της εύρεσης Threshold ως προς την ομοιότητα. Με βάση τον πίνακα (Πίνακας 10) προκύπτουν τα παρακάτω αποτελέσματα.



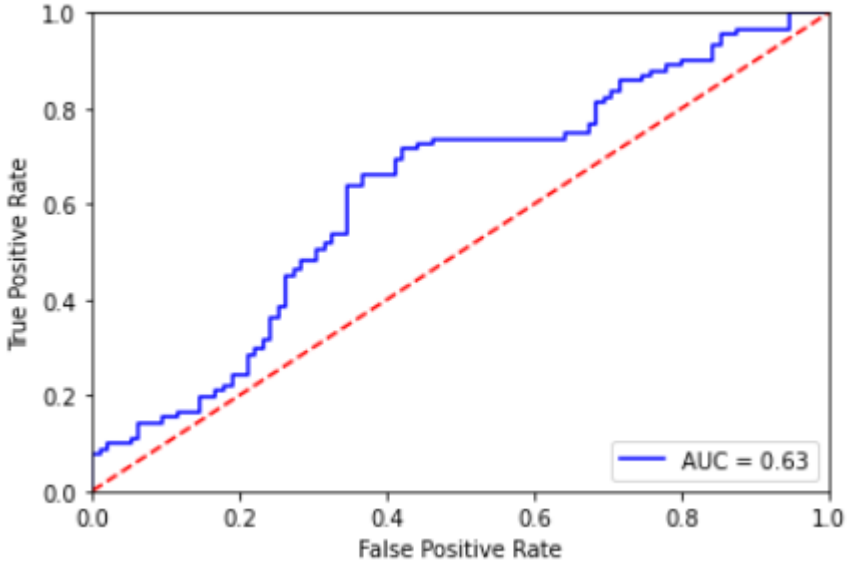
Εικόνα 30: AUC & ROC Curve PAN 2022 για πείραμα με ID 1.

ID	Accuracy	AUC	F1-Score
memo vs essays	65.01 (+/- 0.04)	67	64.2 (+/- 0.05)
Text_message vs email	61.8 (+/- 0.04)	59	58.2 (+/- 0.04)
memo vs email	53.4 (+/- 0.05)	57	51.7 (+/- 0.06)

Πίνακας 23: Συγκεντρωτικά αποτελέσματα για διαφορετικών τύπων κειμένων PAN 2022.

Δεν προέκυψε μοντέλο για άλλους συνδυασμούς καθώς δεν προέκυψαν καλά αποτελέσματα.

Το γενικό αποτελέσματα σε όλο το Test Dataset είναι το παρακάτω:



Εικόνα 31: Γενικό AUC & ROC Curve PAN 2022 για πείραμα με ID 1.

ID	Accuracy	AUC	F1-Score
1	61.3 (+/- 0.05)	63	59.7 (+/- 0.05)

Πίνακας 24: Συγκεντρωτικά αποτελέσματα για PAN 2022.

Παρακάτω παρουσιάζεται η σύγκριση με την καλύτερη προσέγγιση του 2022.

Approach	AUC PAN 2022	F1-Score PAN 2022
najafi22[17]	59.8	57.6
Δική μου προσέγγιση	**63**	**59.7**

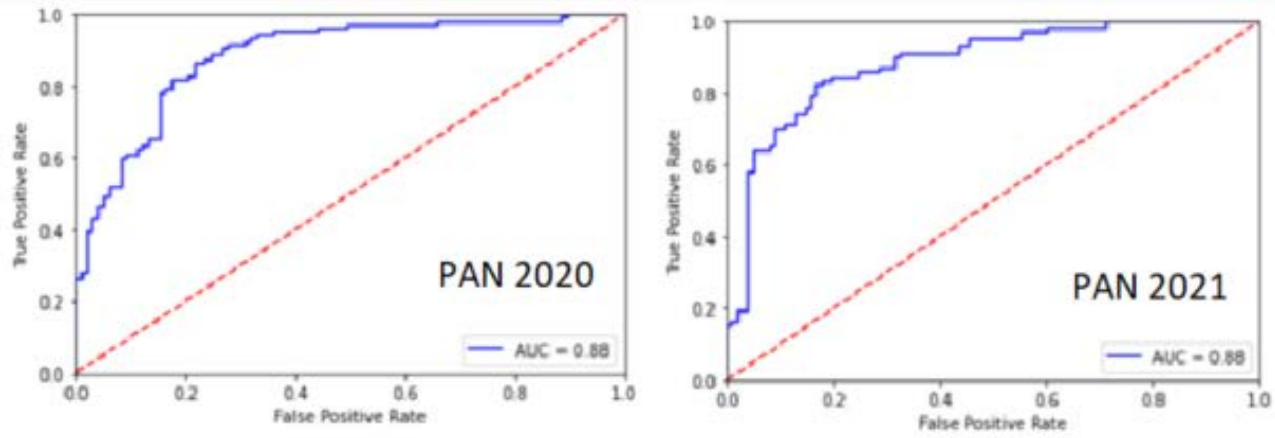
Πίνακας 25: Σύγκριση με την καλύτερη προσέγγιση του 2022.

Με βάση τον παραπάνω πίνακα φαίνεται ότι η μεθοδολογία του Contrastive Learning με χρήση BERT και Bi-LSTM αλλά και ο διαχωρισμός των δεδομένων που αναφέρθηκε σε μικρότερα σώματα δεδομένων με βάση την μορφολογία των

κειμένων ξεπέρασε την καλύτερη προσέγγιση του 2022. Όμως αξίζει να σημειωθεί ότι ακόμα τα scores κυμαίνονται σε χαμηλά επίπεδα και με τυπική απόκλιση 0.05, όπου αν την λάβουμε υπόψιν στα scores τότε η μεθοδολογία μπορεί να βελτιστοποιηθεί ή να επεκταθεί. Αυτό διότι αν αφαιρέσουμε το 0.05 από το 63% του AUC Score ή από το 59.7 του F1 τότε πάλι τα αποτελέσματα είναι αρκετά χαμηλά, αλλά ελπιδοφόρα.

5.6 Αποτελέσματα Classification Task

Το Classification task εκτελέστηκε μόνο στα σώματα δεδομένων το 2020 και 2021 και 2015. Τα πείραμα που εκτελέστηκε για τα 2020 και 2021 παρουσιάζεται στον πίνακα (Πίνακας 14) .

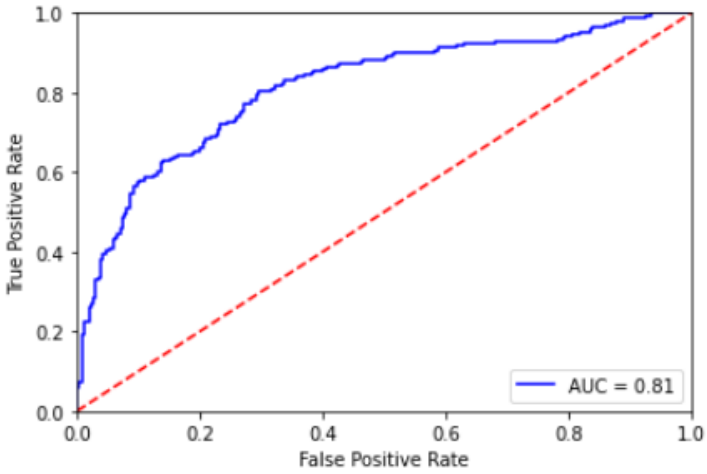


Εικόνα 32: AUC & ROC Curve PAN 2020 & 2021 για πείραμα Classification με ID 1.

ID	Accuracy PAN 2020	Accuracy PAN 2021	AUC PAN 2020	AUC PAN 2021	F1-Score PAN 2020	F1-Score PAN 2021
1	82.5 (+/- 0.03)	82.08 (+/- 0.03)	88	88	80.4 (+/- 0.03)	81.3 (+/- 0.03)

Πίνακας 26: Συγκεντρωτικά αποτελέσματα PAN 2020 & 2021 BERT Uncase Classification.

Για το σώμα δεδομένων του 2015 ακολουθούν τα παρακάτω αποτελέσματα του πειράματος από τον πίνακα (Πίνακας 15).



Εικόνα 33: AUC & ROC Curve PAN 2015 για πείραμα Classification με ID 1.

ID	Accuracy (%)	AUC score (%)	F1-score (%)
1	76.4 (+/- 0.02)	81	73.1 (+/- 0.03)

Πίνακας 27: Συγκεντρωτικά αποτελέσματα PAN 2015 BERT Uncased Classification.

Με βάση τα αποτελέσματα προκύπτει ότι η διαδικασία του Contrastive learning μπορεί να επιφέρει καλύτερα αποτελέσματα, με λιγότερα δεδομένα και λιγότερες εποχές στην διαδικασία του training.

6 Συμπεράσματα

Σύμφωνα με τα παραπάνω αποτελέσματα η μεθοδολογία που ακολουθήθηκε στην παρούσα διπλωματική με Contrastive Learning, μπορεί να επιλύσει το πρόβλημα του Authorship Verification. Φαίνεται ότι η χρήση του BERT Uncased και του RoBERTa μπορεί να επιφέρει πάρα πολύ καλά αποτελέσματα κυρίως σε 3 από τα συνολικά 4 σώματα δεδομένων. Πειραματικά προκύπτει ότι με την χρήση του Contrastive Learning και το masking των αριθμητικών ψηφίων με ένα

συγκεκριμένο ψηφίο, διατηρώντας την μορφολογία, μπορούν να επιτευχθούν καλύτερα αποτελέσματα, σε σχέση με κάποια διαδικασία Classification, με λιγότερα δεδομένα και λιγότερες εποχές.

Επίσης αξίζει να σημειωθεί ότι με την χρήση του BERT Uncased και ενός Bi-LSTM με την διαδικασία του Contrastive Learning με περίπου τα 2/3 του μικρού σώματος δεδομένων του 2020 και 2021 επιτεύχθηκε και προσεγγίστηκε το καλύτερο αποτέλεσμα [13] και [15] στους διαγωνισμούς του 2020 και 2021 αντίστοιχα. Κατά μια έννοια θα μπορούσε να πει κανείς ότι ξεπεράστηκε, καθώς η καλύτερη προσέγγιση σε αυτούς τους διαγωνισμούς χρησιμοποίησε το μεγάλο σώμα δεδομένων και με διαφορετικά μοντέλα σε αντίθεση με την παρούσα διπλωματική όπου εκπαιδεύτηκε ένα μόνο μοντέλο.

Μπορεί κανείς να διαπιστώσει επίσης ότι παίζει αρκετά μεγάλο ρόλο το πλήθος των δεδομένων και ο τύπος των κειμένων. Για παράδειγμα η ίδια μεθοδολογία δεν επέφερε καλά αποτελέσματα στο σώμα δεδομένων του 2022. Επειδή όμως σε επίπεδο Accuracy, AUC score και F1-Score η μεθοδολογία της παρούσας διπλωματικής ξεπέρασε έστω και λίγο την καλύτερη προσέγγιση του 2022 φαίνεται ότι μια επέκταση της μεθοδολογίας που παρουσιάζει η παρούσα διπλωματική να μπορεί να επιλύσει το πρόβλημα και για το σώμα δεδομένων του 2022.

Για παράδειγμα η χρήση triplet loss και ενδεχομένως και κάποιο άλλο προ-εκπαιδευμένο γλωσσικό μοντέλο να είναι μια αρκετά ελπιδοφόρα προσέγγιση, ιδίως εάν συνδυαστούν τα Embeddings του προ-εκπαιδευμένου μοντέλου με κάποια παραδοσιακά Features όπως POS-tags και n-grams.

Τέλος αξίζει να σημειωθεί ότι η χρήση τέτοιων μοντέλων απαιτεί πολλά δεδομένα και υπολογιστική ισχύ. Πολλές φορές μια απλή GPU δεν επαρκεί λόγω μνήμης της vRAM. Αυτό διότι όλες οι εργασίες ενός Transformer αλλά και το ίδιο το σώμα δεδομένων αποθηκεύονται στην μνήμη της GPU. Στην παρούσα διπλωματική χρειάστηκε η Pro έκδοση του Colab με μια Premium GPU με μνήμη 40GB η οποία για κάποια επιπλέον πειράματα δεν επαρκούσε, οπότε δεν εκτελέστηκαν.

Βιβλιογραφία

[1]. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[2]. Stamatatos, Efstathios. "Authorship verification: a review of recent advances." *Research in Computing Science* 123 (2016): 9-25.

[3]. Potha, Nektaria, and Efstathios Stamatatos. "An improved impostors method for authorship verification." *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8*. Springer International Publishing, 2017.

[4]. Ding, Steven HH, et al. "Learning stylometric representations for authorship analysis." *IEEE transactions on cybernetics* 49.1 (2017): 107-121.

[5]. Tyo, Jacob, Bhuwan Dhingra, and Zachary C. Lipton. "Siamese Bert for Authorship Verification." *CLEF (Working Notes)*. 2021.

[6]. Potha, Nektaria, and Efstathios Stamatatos. "A profile-based method for authorship verification." *Artificial Intelligence: Methods and Applications: 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings 8*. Springer International Publishing, 2014.

[7]. Boenninghoff, Benedikt, et al. "Explainable authorship verification in social media via attention-based similarity learning." *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019.

[8]. Stamatatos, Efstathios, et al. "Overview of the author identification task at PAN 2014." *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014*. 2014.

[9]. Bagnall, Douglas. "Author identification using multi-headed recurrent neural networks." *arXiv preprint arXiv:1506.04891* (2015).

[10]. Moreau, Erwan, et al. "Author verification: Basic stacked generalization applied to predictions from a set of heterogeneous learners-notebook for pan at clef 2015." *CLEF 2015-Conference and Labs of the Evaluation forum*. CEUR, 2015.

[11]. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20 (1995): 273-297.

[12]. Kestemont, Mike, et al. "Overview of the Cross-Domain Authorship Verification Task at PAN 2020." *CLEF (Working Notes)*. 2020.

[13]. Boenninghoff, Benedikt, et al. "Deep bayes factor scoring for authorship verification." *arXiv preprint arXiv:2008.10105* (2020).

[14]. Bevendorff, Janek, et al. "Overview of pan 2021: Authorship verification, profiling hate speech spreaders on twitter, and style change detection." *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings 12*. Springer International Publishing, 2021.

[15]. Boenninghoff, Benedikt, Robert M. Nickel, and Dorothea Kolossa. "O2D2: Out-of-distribution detector to capture undecidable trials in authorship verification." *arXiv preprint arXiv:2106.15825* (2021).

[16]. Stamatatos, Efstathios, et al. "Overview of the authorship verification task at PAN 2022." CEUR workshop proceedings. Vol. 3180. 2022.

[17]. Najafi, Maryam, and Ehsan Tavan. "Text-to-Text Transformer in Authorship Verification Via Stylistic and Semantical Analysis." Proceedings of the CLEF. 2022.

[18]. Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *The Journal of Machine Learning Research* 21.1 (2020): 5485-5551.

[19]. Ihianle, Isibor Kennedy, et al. "A deep learning approach for human activities recognition from multimodal sensing devices." *IEEE Access* 8 (2020): 179028-179038.

[20]. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[21]. Jawahar, Ganesh, Benoît Sagot, and Djamé Seddah. "What does BERT learn about the structure of language?." *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*. 2019.

[22]. Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units." *arXiv preprint arXiv:1508.07909* (2015).

[23]. Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).

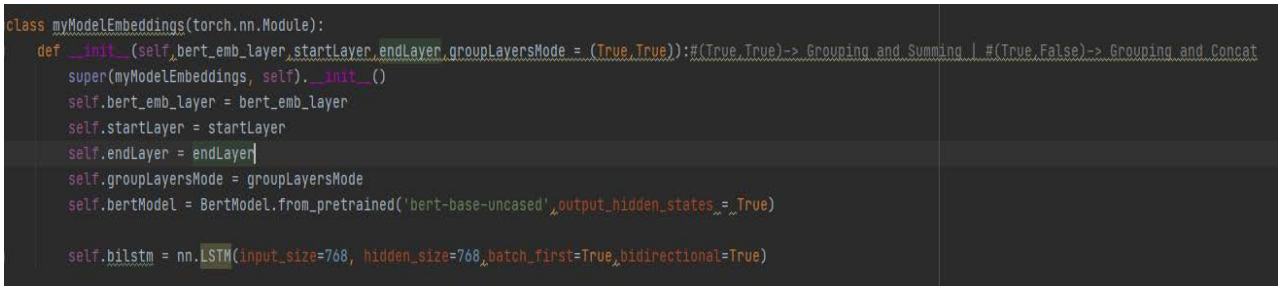
[24]. Gage, Philip. "A new algorithm for data compression." *C Users Journal* 12.2 (1994): 23-38.

[25]. Shah, Anshul, et al. "Max-Margin Contrastive Learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. No. 8. 2022.

[26]. Konstantinou, Stefanos, A. Zinonos, and J. Li. "Different Encoding Approaches for Authorship Verification." CLEF, 2022.

Παράρτημα Α

Στην παρακάτω εικόνα ακολουθεί η κλάση σε γλώσσα Python και Pytorch που αντιστοιχεί στην αρχιτεκτονική του μοντέλου.



```
class myModelEmbeddings(torch.nn.Module):
    def __init__(self,bert_emb_layer,startLayer,endLayer,groupLayersMode = (True,True)):#(True,True)-> Grouping and Summing | #(True,False)-> Grouping and Concat
        super(myModelEmbeddings, self).__init__()
        self.bert_emb_layer = bert_emb_layer
        self.startLayer = startLayer
        self.endLayer = endLayer
        self.groupLayersMode = groupLayersMode
        self.bertModel = BertModel.from_pretrained('bert-base-uncased',output_hidden_states = True)

        self.bilstm = nn.LSTM(input_size=768, hidden_size=768,batch_first=True,bidirectional=True)
```

Επειδή έχουμε Contrastive Learning για να είναι πιο εύκολη η υλοποίηση έχουμε την forward() function στην pytorch και μια βοηθητική forward_once() function που κατά την εκτέλεση προηγείται της forward και στην ουσία είναι υπεύθυνη να κρατήσει τα outputs του BERT πριν γίνει οποιοδήποτε polling ή επεξεργασία.

```
def forwardOnce(self, sent_id, mask):
    outputs = self.bertModel(sent_id, attention_mask=mask)

    if self.groupLayersMode == (True,False):
        embeddings = self.concatSpecificLayersOfBERT(outputs)
        return embeddings
    elif self.groupLayersMode == (True,True):
        embeddings = self.sumSpecificLayersOfBERT(outputs)
        # embeddings = self.getCLSEmbeddingsFromLayers(outputs)
        return embeddings
    else:
        # embeddings = self.getSpecificLayerOfBERT(outputs)
        embeddings = self.getCLSEmbeddings(outputs)
        return embeddings
```

```python
def forward(self, sent_id1, mask1,hidden):

    # forward pass of input 1
    output1 = self.forwardOnce(sent_id1, mask1)

    out1, (hidden1,cell1) = self.bilstm(output1,hidden)
    out_split1 = out1.view(sent_id1.shape[0], 512, 2, 768)
    batch_indices = torch.arange(0, sent_id1.shape[0])
    seq_indices = 512 - 1
    direction_full1 = torch.cat([out_split1[batch_indices, seq_indices, 0], out_split1[batch_indices, 0, 1]], dim=-1)
    return direction_full1
```

Για την διαδικασία του pooling πρέπει να λάβουμε υπόψιν μας τον attention μηχανισμό (attention mask). Εάν δεν τον λάβουμε τότε οποιοδήποτε pooling (average ή max ή sum) θα συμπεριλάβει και τα pad tokens που έχουν Embeddings (βάρη) μηδενικά. Αυτό υλοποιήθηκε με την παρακάτω συνάρτηση.

```python
def pooling(self,token_embeddings, mask, strategy='avg'):
    if strategy == 'max':
        input_mask_expanded = mask.unsqueeze(-1).expand(token_embeddings.size()).float()
        token_embeddings[input_mask_expanded == 0] = -1e9  # Set padding tokens to large negative value
        max_setence_embeddings = torch.max(token_embeddings, 1)[0]
        return max_setence_embeddings
    elif strategy == 'avg':
        in_mask = mask.unsqueeze(-1).expand(token_embeddings.size()).float()
        # perform mean-pooling but exclude padding tokens (specified by in_mask)
        avg_setence_embeddings = torch.sum(token_embeddings * in_mask, 1) / torch.clamp(in_mask.sum(1), min=1e-9)
        return avg_setence_embeddings
    elif strategy == 'sum':
        sum_setence_embeddings = torch.sum(token_embeddings[0:len(token_embeddings)],1)
        return sum_setence_embeddings
```

Ο κώδικας υπάρχει διαθέσιμος και στο github.

https://github.com/icsd13152/Thesis_MSC_in_AI_AuthorVerif