ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Αναγνώριση λίστας αναφορών σε ακαδημαϊκά άρθρα με χρήση μεθόδων Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing)»

Του φοιτητή
Παπαπαύλου Χαρίλαου
Αρ. Μητρώου: 518109

Επιβλέπων
Σιδηρόπουλος Αντώνης
Αναπληρωτής Καθηγητής

Ημερομηνία 26/1/2025

Τίτλος Δ.Ε. Αναγνώριση λίστας αναφορών σε ακαδημαϊκά άρθρα με χρήση μεθόδων Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing)
Κωδικός Δ.Ε. 24232
Ονοματεπώνυμο φοιτητή Παπαπαύλου Χαρίλαος
Ονοματεπώνυμο εισηγητή Σιδηρόπουλος Αντώνης
Ημερομηνία ανάληψης Δ.Ε. 23-09-2024
Ημερομηνία περάτωσης Δ.Ε. 26-01-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιερώνω την παρούσα

Διπλωματική Εργασία

στην οικογένεια μου και στους φίλους μου.»

Πρόλογος

Η επιλογή του θέματος της διπλωματικής μου εργασίας προέκυψε από το ενδιαφέρον μου για την έρευνα μεθόδων Επεξεργασίας Φυσικής Γλώσσας. Πιο συγκεκριμένα, έχω σαν στόχο να αυτοματοποιήσω την διαδικασία της επεξεργασίας μεγάλων όγκων δεδομένων και να αξιοποιήσω την τεχνολογία για να βελτιωθεί η ερευνητική διαδικασία. Η παρούσα εργασία προσφέρει σημαντικό όφελος για την ανάπτυξη γνώσεων μου και για την προσωπική μου ανάπτυξη αλλά και για την επιστημονική κοινότητα. Απέκτησα πολύτιμη γνώση σε θέματα επεξεργασίας φυσικής γλώσσας και αξιολόγησης μοντέλων μέσα από την ανάλυση και την εφαρμογή μοντέλων μηχανικής μάθησης. Ταυτόχρονα, η πρακτική εφαρμογή αυτών των τεχνολογιών ενίσχυσε την ικανότητα μου να βρίσκω λύσεις σε προβλήματα που προέκυψαν. Τέλος, η εργασία συμβάλλει στην εύρεση εργαλείων που θα μπορούν ερευνητές να χρησιμοποιούν, βοηθώντας τους στο να ξοδέψουν στο ερευνητικό κομμάτι περισσότερους πόρους.

Περίληψη

Το βασικό αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη και η αξιολόγηση μοντέλων μηχανικής μάθησης με σκοπό να έχουν την δυνατότητα να διαχωρίσουν τις βιβλιογραφικές αναφορές από το υπόλοιπο κείμενο με όσο το δυνατόν μεγαλύτερη ακρίβεια.

Αρχικά, δημιουργήθηκε ένα dataset που περιλαμβάνει βιβλιογραφικές αναφορές σε στυλ MLA και APA και επιπλέον απλές προτάσεις σε μεγάλο εύρος ποικιλίας. Το dataset αυτό, δημιουργήθηκε με σκοπό να εκπαιδεύσουμε και να αξιολογήσουμε μοντέλα όπως το Support Vector Machine(SVM), K-Nearest Neighbors(KNN) και το BERT. Αφού χωρίσαμε το dataset σε 2 κατηγορίες, εκπαίδευσης και αξιολόγησης, αξιολογήσαμε τα μοντέλα μας.

Από τα αποτελέσματα συμπεράναμε ότι το KNN και το SVM παρουσίασαν ικανοποιητικά, και σε κάποιες περιπτώσεις υψηλά, ποσοστά επιτυχίας. Το BERT, όπως ήταν και αναμενόμενο διότι είναι σύγχρονο μοντέλο, έδειξε εξαιρετική απόδοση.

Η εργασία ολοκληρώθηκε δοκιμάζοντας τα μοντέλα σε ένα πραγματικό σύνολο δεδομένων, το οποίο περιλαμβάνει βιβλιογραφικές αναφορές οι οποίες δεν υπάρχουν στο αρχικό dataset ώστε να δοκιμαστούν τα μοντέλα στην ικανότητα να εντοπίζουν αναφορές με ακρίβεια σε ξένα δεδομένα.

Συνολικά, η εργασία συνεισφέρει στην αυτοματοποίηση της διαδικασίας διαχείρισης δεδομένων και παρέχει τεχνολογίες με σκοπό να βελτιωθεί η επιστημονική έρευνα.

«Recognition of reference lists in academic articles using Natural Language Processing methods»

«Papapavlou Charilaos»

Abstract

The main objective of this thesis is to develop and evaluate machine learning models with the aim of being able to separate bibliographic references from the rest of the text with the highest possible accuracy.

Initially, a dataset was created that includes bibliographic references on MLA and APA styles and simple sentences in a wide range of variety. This dataset was created with the purpose of training and evaluating models such as Support Vector Machines (SVM), K-Nearest Neighbors(KNN) and BERT. After we split the dataset into 2 categories, training and evaluation, we evaluated our models.

From the results, we concluded that KNN and SVM exhibited satisfactory, and in some cases high, success rates. BERT, as expected since it is a transformer model, showed excellent performance.

The work was completed by testing the models on a real dataset, which includes bibliographic references that are not present in the original dataset, in order to evaluate the models' ability to accurately identify references in foreign data.

Overall, the work contributes to the automation of the data management process and provides technologies aimed at improving scientific research.

Ευχαριστίες

Πριν την παρουσίαση των αποτελεσμάτων της παρούσας εργασίας, θέλω να ευχαριστήσω την οικογένεια μου για την υποστήριξή τους κατά την διάρκεια των ακαδημαϊκών μου σπουδών. Επίσης αισθάνομαι την υποχρέωση να ευχαριστήσω τον καθηγητή και επιβλέποντα της διπλωματικής μου, Σιδηρόπουλο Αντώνη, καθώς και τον υποψήφιο διδάκτορα Αριστοτέλη Καμπατζή, για την εξαιρετική τους καθοδήγηση και βοήθεια καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

1.1 Στόχος της διπλωματικής

Η παρούσα διπλωματική εργασία έχει σαν στόχο την ανάπτυξη ενός συστήματος το οποίο θα είναι σε θέση να αναγνωρίζει αυτόματα τις λίστες βιβλιογραφικών αναφορών μέσα σε ακαδημαϊκά άρθρα, με την χρήση μεθόδων Επεξεργασίας Φυσικής Γλώσσας (NLP).Αξιοποιώντας τεχνικές NLP το σύστημα θα μπορεί να εντοπίζει, εξάγει και να οργανώνει τις βιβλιογραφικές αναφορές, με σκοπό να αυτοματοποιηθεί η διαδικασία αναγνώρισης και διαχείρισης των αναφορών σε επιστημονικά κείμενα. Η χειροκίνητη διαχείριση των αναφορών, ειδικά σε μεγάλο όγκο δημοσιεύσεων είναι μια διαδικασία που απαιτεί χρόνο και υπόκειται σε ανθρώπινα σφάλματα, γι' αυτό άλλωστε υπάρχει και η ανάγκη για αυτοματοποίηση.

Αξιοποιώντας σύγχρονες τεχνικές NLP, όπου θα αναλύσουμε παρακάτω, για τον εντοπισμό συγγραφέων, τίτλων, εκδοτών και άλλων βασικών στοιχείων των αναφορών, το σύστημα θα είναι σε θέση να ανιχνεύει αναφορές ανεξαρτήτως μορφής (π.χ. APA, MLA, IEEE, Chicago).

Το σύστημα θα στοχεύει στην αυτοματοποίηση της διαδικασίας αναγνώρισης, εξαγωγής και οργάνωσης των βιβλιογραφικών αναφορών, με σκοπό την μεγάλη ακρίβεια και ταχύτητα για την διαχείριση τους. Μέσα από την ανάπτυξη και εφαρμογή των παραπάνω μεθόδων, το σύστημα θα προσφέρει σημαντική βοήθεια σε ερευνητές, επιστήμονες και φοιτητές, μειώνοντας τον χρόνο και το κόστος της διαχείρισης των βιβλιογραφικών δεδομένων.

1.2 Δομή εργασίας

- Η παρούσα διπλωματική εργασία είναι οργανωμένη σε επιμέρους κεφάλαια, καθένα από τα οποία εστιάζει σε διαφορετικές πτυχές του θέματος. Κάθε κεφάλαιο εμβαθύνει στην ανάλυση σχετικών τεχνικών και εννοιών, ενώ παράλληλα καθοδηγεί τον αναγνώστη από την θεωρία στην εφαρμογή της.
- Κεφάλαιο 2 «Βιβλιογραφική ανασκόπηση» Παρουσιάζει μία βιβλιογραφική ανασκόπηση και ανάλυση των ήδη υπαρχόντων συστημάτων που υπάρχουν και έχουν αναπτυχθεί για την αναγνώριση βιβλιογραφικών αναφορών. Επιπλέον, περιλαμβάνει μια ανασκόπηση της τωρινής κατάστασης στον τομέα και θίγει τα πλεονεκτήματα και τα μειονεκτήματα των υφιστάμενων προσεγγίσεων.
- Κεφάλαιο 3 «Ανάπτυξη και αξιολόγηση» Σε αυτό το κεφάλαιο παρουσιάζονται τα μοντέλα τα οποία εκπαιδεύσαμε και παράλληλα γίνεται ένας σχολιασμός των αποτελεσμάτων τους.
- Κεφάλαιο 4 «Συμπεράσματα και προτάσεις βελτίωσης» Θα συζητηθούν εκτενώς τα συμπεράσματα από την διπλωματική μου εργασία και επιπλέον θα αναφερθεί η μελλοντική έρευνα και η συνέχεια της εργασίας.

Κεφάλαιο 2ο: Βιβλιογραφική ανασκόπηση

2.1 Εισαγωγή

Αυτό το κεφάλαιο παρουσιάζει:

- Ορισμούς και μια ανασκόπηση της σχετικής βιβλιογραφίας για το NLP και τις σχετικές τεχνολογίες.
- Διατύπωση του προβλήματος της εργασίας και των ερευνητικών ερωτήσεων.

2.2 Τεχνητή νοημοσύνη

Η τεχνητή νοημοσύνη (Artificial intelligence - AI) είναι η νοημοσύνη των μηχανών ή του λογισμικού. Είναι ένα πεδίο μελέτης στην επιστήμη των υπολογιστών που αναπτύσσει και μελετά έξυπνες μηχανές. Η τεχνολογία της τεχνητής νοημοσύνης χρησιμοποιείται ευρέως σε όλη την βιομηχανία, την κυβέρνηση και την επιστήμη. Ο όρος AI καλύπτει ένα τεράστιο φάσμα εργαλείων που επιδιώκουν να δημιουργήσουν συστήματα με γνωστικές ικανότητες π.χ. η αντίληψη, η μάθηση και η κατανόηση γλώσσας. Σαν στόχο έχει να επεξεργάζεται σύνθετα δεδομένα [1].

Η AI καλύπτει ένα μεγάλο φάσμα τεχνολογιών, όπως η μηχανική μάθηση, η βαθιά μάθηση και τα νευρωνικά δίκτυα και έχουν σαν σκοπό την δημιουργία συστημάτων με βελτιωμένες γνωστικές ικανότητες. Ένας στόχος που έχει πετύχει η AI και συνεχίζει να τον βελτιώνει είναι η ανάπτυξη συστημάτων που μπορούν να προσαρμόζονται σε πληροφορίες και να αλληλοεπιδρούν με τους χρήστες. Για παράδειγμα οι μηχανές αναζήτησης όπως το Google Search χρησιμοποιούν προηγμένους αλγορίθμους AI για την εξατομίκευση των αποτελεσμάτων. Αντιστοίχως, στα συστήματα συστάσεων που χρησιμοποιούν εταιρίες όπως το Youtube και το Netflix, εφαρμόζουν τεχνικές AI για να προτείνουν περιεχόμενο στον χρήστη [2][3].

Ένα από τα πιο πρόσφατα παραδείγματα τεχνητής νοημοσύνης είναι τα αυτόνομα οχήματα όπως αυτά της Tesla τα οποία βασίζονται σε τεχνολογίες για την λήψη αποφάσεων και την πλοήγηση τους. Επιπλέον, έχει διευκολυνθεί πάρα πολύ πλέον η επικοινωνία μεταξύ ανθρώπου-μηχανής χάρη στους εικονικούς βοηθούς (π.χ. Google Assistant, Siri, Alexa) και έχουν βοηθήσει στην κατανόηση γλωσσικών εντολών [2][3].

Οι προοπτικές και το πόσο μακριά μπορεί να φτάσει η τεχνολογία με την χρήση της τεχνητής νοημοσύνης είναι τεράστιες, αλλά παράλληλα θίγονται ηθικά και κοινωνικά ζητήματα, όπως η προστασία των δεδομένων και η διαφάνεια των αλγορίθμων [4].

2.2.1 Μηχανική Μάθηση

Η μηχανική μάθηση (Machine Learning) είναι ένας κλάδος της τεχνητής νοημοσύνης που επικεντρώνεται στην ανάπτυξη και μελέτη αλγορίθμων που μπορούν να μάθουν από δεδομένα. Οι αλγόριθμοι αυτοί, επιτρέπουν στα συστήματα να αναγνωρίζουν σχέσεις, να εξάγουν μοτίβα και να μαθαίνουν αυτόνομα με αποτέλεσμα να εκτελούν καθήκοντα χωρίς ρητή ή προγραμματιστική καθοδήγηση. Οι μέθοδοι της μηχανικής μάθησης βασίζονται σε μαθηματικά μοντέλα που συνδέονται με την στατιστική που εξειδικεύονται στην πρόβλεψη και επιτρέπουν την συνεχή βελτίωση και προσαρμογή σε νέα δεδομένα, βελτιώνοντας την απόδοση τους με την πάροδο του χρόνου [5].

2.2.1.1 Βαθιά Μάθηση

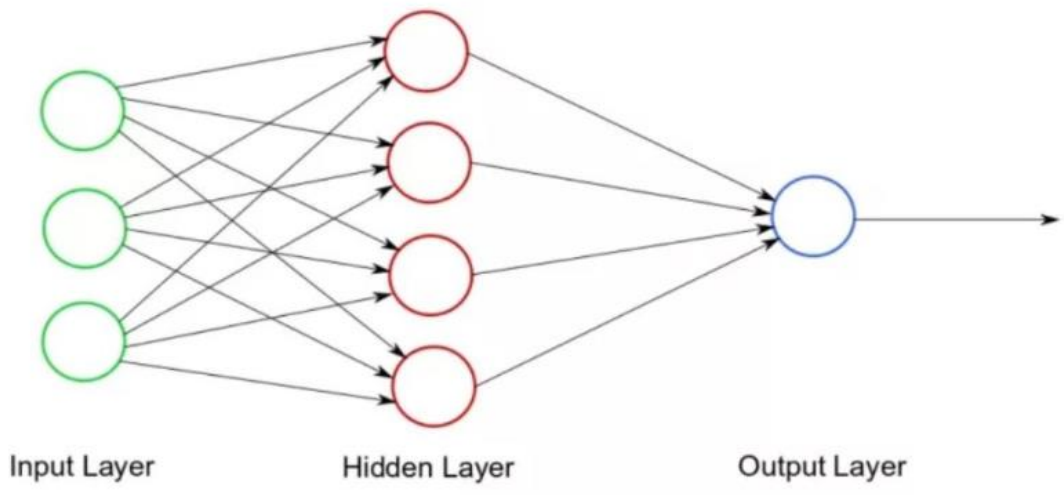
Η βαθιά μάθηση (Deep Lerning) είναι μια υποκατηγορία της μηχανικής μάθησης που διδάσκει στους υπολογιστές να επεξεργάζονται δεδομένα με τρόπο που μιμείται τον ανθρώπινο εγκέφαλο. Για να παράγουν ακριβείς πληροφορίες και προβλέψεις, τα μοντέλα βαθιάς μάθησης είναι ικανά να εντοπίζουν περίπλοκα μοτίβα σε εικόνες, κείμενα, ήχους και άλλα είδη δεδομένων. Οι τεχνικές βαθιάς μάθησης μπορούν να χρησιμοποιηθούν για την αυτοματοποίηση διαδικασιών όπως η περιγραφή εικόνας και η μεταγραφή ηχητικών αρχείων σε κείμενο που συνήθως απαιτούν ανθρώπινη νοημοσύνη. Το επίθετο "βαθιά" αναφέρεται στην χρήση πολλαπλών στρωμάτων (από τρεις έως αρκετές εκατοντάδες ή χιλιάδες) στο δίκτυο [5].

2.2.2 Νευρωνικά δίκτυα

Τα Νευρωνικά δίκτυα (Neural Networks) αποτελούν μια κατηγορία αλγορίθμων μηχανικής μάθησης εμπνευσμένοι από τον τρόπο με τον οποίο είναι δομημένος ένας ανθρώπινος εγκέφαλος και από τον τρόπο με τον όποιο λειτουργεί. Ένα νευρωνικό δίκτυο συμπεριλαμβάνει στοιχεία τα οποία είναι διασυνδεδεμένα μεταξύ τους, γνωστά και ως «νευρώνες» τα οποία είναι χωρισμένα σε επίπεδα. Το επίπεδο εισόδου, τα κρυφά επίπεδα και το επίπεδο εξόδου. Εφαρμόζοντας μαθηματικούς υπολογισμούς οι νευρώνες επεξεργάζονται πληροφορίες. Επιπλέον, οι νευρώνες συμπεριλαμβάνουν και συνάψεις(weights) που προσαρμόζονται μέσω της διαδικασίας της εκπαίδευσης, με σκοπό να βελτιωθεί η ακρίβεια του δικτύου στις προβλέψεις του [6][7].

Κάθε νευρώνας λαμβάνει σήματα εισόδου και αφού τα επεξεργαστεί, μεταδίδει τα αποτελέσματα στους επόμενους νευρώνες του δικτύου. Η εκπαίδευση των νευρωνικών δικτύων γίνεται μέσω αλγορίθμων βελτιστοποίησης, οι οποίοι επιτρέπουν την επεξεργασία των «weights» για την μείωση των σφαλμάτων μεταξύ πλασματικών και πραγματικών αποτελεσμάτων [6][7].

Χάρη στα νευρωνικά δίκτυα και στην ικανότητα τους να αναγνωρίζουν πολύπλοκα μοτίβα και να μαθαίνουν από μη δομημένα δεδομένα έχουν σημειώσει πολλές σημαντικές επιτυχίες σε εφαρμογές όπως η αναγνώριση εικόνας και η επεξεργασία φυσικής γλώσσας. Αποτελούν τον πυρήνα του «Deep learning» και προσφέρουν υποστήριξη στην εξέλιξη της Τεχνητής νοημοσύνης [7].



Εικόνα 1. Επίπεδα νευρώνων [8].

2.3 Natural Language Processing

Η επεξεργασία Φυσικής Γλώσσας (NLP) είναι ένας υποτομέας της τεχνητής νοημοσύνης (AI) και της επιστήμης υπολογιστών, καθώς στοχεύει στην κατανόηση, ανάλυση και παραγωγή φυσικής γλώσσας από μηχανές. Πιο συγκεκριμένα, το NLP επιτρέπει στους υπολογιστές να «καταλαβαίνουν» και να επεξεργάζονται την ανθρώπινη γλώσσα, απλοποιώντας και διευκολύνοντας την αλληλεπίδραση μεταξύ ανθρώπων και υπολογιστών χρησιμοποιώντας μηχανική μάθηση. Η επεξεργασία φυσικής γλώσσας (NLP) επιτρέπει στους υπολογιστές και τις ψηφιακές συσκευές να αναγνωρίζουν, να κατανοούν και να παράγουν κείμενο και ομιλία συνδυάζοντας την υπολογιστική γλωσσολογία μαζί με τη μηχανική μάθηση, τη στατική μοντελοποίηση και την βαθιά μάθηση [6][9].

2.3.1 NLU

Το NLU σημαίνει κατανόηση φυσικής γλώσσας (Natural Language Understanding) και είναι ένας υποτομέας του NLP. Μιλάμε δηλαδή για το μέρος του συστήματος που δέχεται ένα ακατέργαστο κείμενο και εξέρχεται από αυτό, κείμενο που μπορεί να διαβαστεί από ένα μηχάνημα-υπολογιστή.

Συνολικά, η NLU είναι εξαιρετικά χρήσιμη στο θέμα της αυτόματης αναγνώρισης βιβλιογραφικών αναφορών, καθώς παρέχει τη δυνατότητα στους υπολογιστές να επεξεργάζονται σύνθετα κείμενα με μεγαλύτερη ακρίβεια και βάθος, μειώνοντας τα λάθη και αυξάνοντας την αποδοτικότητα στην οργάνωση της βιβλιογραφίας [6][9].

2.3.2 NLG

Η Παραγωγή Φυσικής Γλώσσας (Natural Language Generation) είναι κλάδος της Επεξεργασίας Φυσικής Γλώσσας (NLP) ο οποίος επικεντρώνεται στην ανάποδη διαδικασία από το NLU, δηλαδή στην δημιουργία φυσικής γλώσσας από υπολογιστικά συστήματα. Η NLG ασχολείται μόνο με την δημιουργία κειμένου που μπορεί να κατανοηθεί και διαβαστεί από ανθρώπους. Στην ουσία, η NLG επιτρέπει στους υπολογιστές να επικοινωνούν με φυσικό τρόπο, παράγοντας προτάσεις.

Επιπλέον, η NLG μπορεί να δημιουργήσει κείμενα, αυτόματα, όπως αναφορές, περιλήψεις, και άρθρα. Στον κλάδο της βιβλιογραφικής αναγνώρισης, η NLG μπορεί να αξιοποιηθεί για την παρουσίαση των βιβλιογραφικών αναφορών με οργανωμένο τρόπο που έχουν εξαχθεί από ακαδημαϊκά κείμενα. Έτσι, οι βιβλιογραφικές αναφορές μπορούν να παραχθούν με βάση τα καθιερωμένα πρότυπα αναφοράς, όπως APA, MLA ή IEE, καθιστώντας αυτή την τεχνολογία ιδιαίτερα χρήσιμη για την αυτόματη δημιουργία λιστών βιβλιογραφίας [9].

2.3.3 Τεχνικές Επεξεργασίας Φυσικής Γλώσσας (NLP)

Οι τεχνικές του NLP είναι η βάση για την ανάλυση, κατανόηση και εξαγωγή δεδομένων μέσα σε ακαδημαϊκά άρθρα και κείμενα. Στον τομέα της αυτόματης αναγνώρισης βιβλιογραφικών αναφορών, οι τεχνικές αυτές ενισχύουν την δυνατότητα των μηχανών να διαχειρίζονται βιβλιογραφικά δεδομένα με αποτελεσματικότητα και συνέπεια. Παρακάτω θα αναλυθούν τεχνικές που βοηθάνε στην διαδικασία της αναγνώρισης βιβλιογραφικών αναφορών.

2.3.3.1 Tokenization

Το Tokenization είναι η διαδικασία στην οποία ένα κείμενο διασπάτε σε μικρότερα κομμάτια όπως λέξεις, προτάσεις ή φράσεις, αυτά τα κομμάτια τα ονομάζουμε «tokens».Αυτό θα πρέπει να συμβαίνει πριν την μηχανική μάθηση, καθώς τα περισσότερα μοντέλα φυσικής γλώσσας απαιτούν συμβολική ή αριθμητική αναπαράσταση για να κατανοήσουν ένα κείμενο. Στο κομμάτι των βιβλιογραφικών

αναφορών, ο διαχωρισμός σε tokens καθιστά δυνατή την αναγνώριση και εξαγωγή των διαφόρων συστατικών μιας αναφοράς, όπως το όνομα του συγγραφέα, ο τίτλος του άρθρου, το έτος έκδοσης. Η διάσπαση διευκολύνει την ανάλυση της δομής της αναφοράς [9][10].

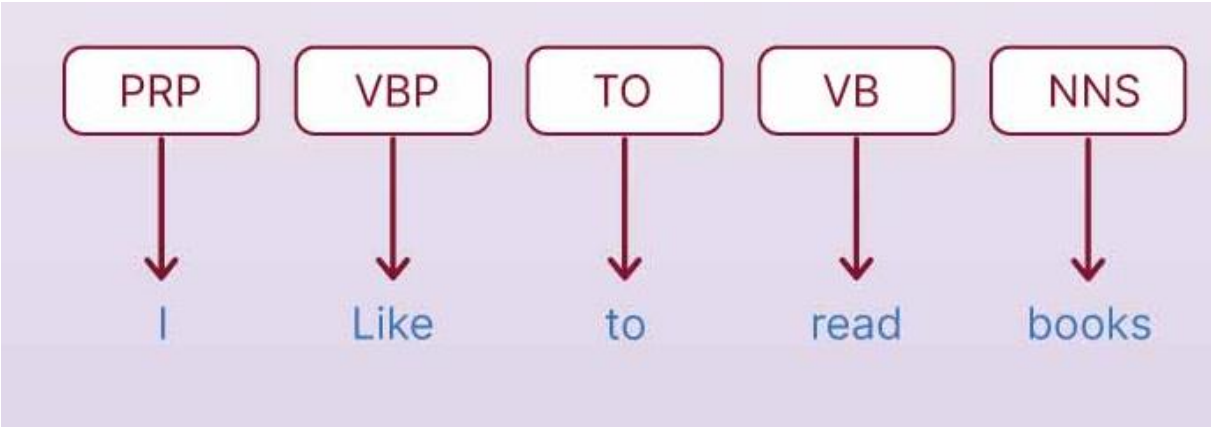
Εικόνα 2. Tokenizers

2.3.3.2 Named Entity Recognition (NER) – Αναγνώριση Οντοτήτων

Η Αναγνώριση Οντοτήτων (Named Entity Recognition – NER) ανήκει στην κατηγορία των βασικότερων τεχνικών του NLP και είναι πιθανώς το πρώτο βήμα προς την εξαγωγή δεδομένων. Έχει την δυνατότητα να εντοπίζει και να κατηγοριοποιεί συγκεκριμένες οντότητες μέσα σε ένα κείμενο όπως τοποθεσίες, ημερομηνίες και ονόματα προσώπων και οργανισμών. Το NER δεν λειτουργεί μόνο ως αυτόνομο εργαλείο, αλλά παίζει επίσης πολύ σημαντικό ρόλο σε διάφορες εφαρμογές NLP όπως η αναγνώριση ομιλίας, τα chatbots, η ανάλυση συναισθημάτων, η κατηγοριοποίηση κειμένων, η αυτόματη περίληψη κ.λπ. Το NER χρησιμοποιεί συγκεκριμένους αλγορίθμους και μοντέλα για την αναγνώριση των στοιχείων που προαναφέρθηκαν και να τα κατατάξει σε κατηγορίες. Θεωρείται μια ώριμη τεχνολογία με υψηλή ακρίβεια, βέβαια απαιτεί την χειροκίνητη κατασκευή μοντέλων και ακριβών συνόλων δεδομένων πράγμα που την καθιστά δαπανηρή τεχνολογία [11][12][13].

2.3.3.3 POS Tagging

Το POS Tagging (Part-of-Speech tagging) είναι μια διαδικασία στο NLP όπου όλες οι λέξεις σε ένα κείμενο επισημαίνονται στην σωστή γραμματική κατηγορία στην οποία ανήκουν, όπως ρήμα, ουσιαστικό, επίθετο και επίρρημα. Το POS Tagging συνδυάζει στατικές μεθόδους και μοντέλα μηχανικής μάθησης για να προβλέψει την κατάλληλη ετικέτα σε μία λέξη με σκοπό την εμφάνιση κατηγοριών στα συμφραζόμενα. Σύγχρονα συστήματα του POS Tagging χρησιμοποιούν επίσης νευρωνικά δίκτυα και προεκπαιδευμένα γλωσσικά μοντέλα για την βελτίωση της ακρίβειας [14][15].



Εικόνα 3. Παράδειγμα POS Tagging [16]

2.3.3.4 Regular Expressions (Regex) - Κανονικές Εκφράσεις

Οι κανονικές εκφράσεις(Regular Expressions – Regex) είναι ένας από τους αποτελεσματικότερους τρόπους για να εντοπίζονται μοτίβα λέξεων σε ένα κείμενο. Η regex, μπορεί να χρησιμοποιηθεί για να

εντοπίσει συγκεκριμένες μορφές αναφορών, συμπεριλαμβανόμενων των ονομάτων, των συγγραφέων, των τίτλων άρθρων και χρονολογιών. Όταν εντοπίζεται μια δομή αναφορών που ακολουθούν συγκεκριμένα στυλ βιβλιογραφικών αναφορών (π.χ. APA, MLA, IEEE), τότε το regex είναι ιδιαίτερα χρήσιμο [17][18].

2.3.3.5 Lemmatization και Stemming

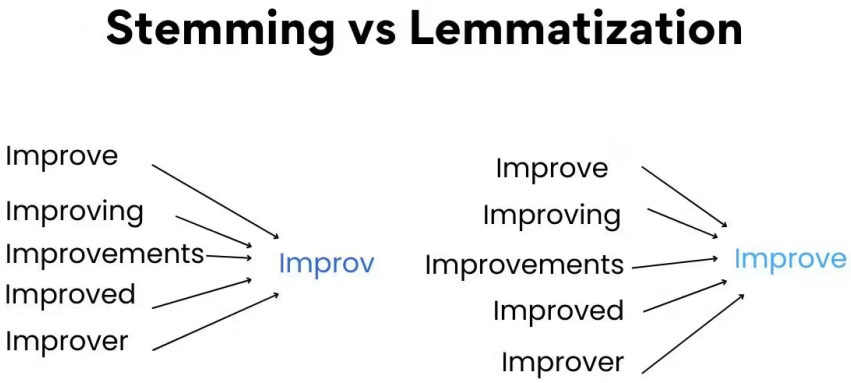
Το Lemmatization και το Stemming είναι τεχνικές που ανήκουν στην Φυσική Επεξεργασία Γλώσσας(NLP) και χρησιμεύουν στην κανονικοποίηση των λέξεων με σκοπό το σύστημα να μπορεί να αναγνωρίσει λέξεις με παρόμοια σημασία σε διαφορετικές μορφές. Πιο συγκεκριμένα:

Το **Lemmatization** έχει σαν στόχο να φέρει μια λέξη στην μορφή με την οποία θα εμφανιζόταν στο λεξικό, δηλαδή, είναι η διαδικασία καθορισμού του λήμματος μιας λέξης με βάση την προοριζόμενη σημασία της. Επιπλέον, το Lemmatization εξαρτάται από την σωστή αναγνώριση του προοριζόμενου μέρους του λόγου που περιβάλλει την πρόταση καθώς και το ευρύτερο πλαίσιο που την περιβάλλει. Σε πολλές γλώσσες, οι λέξεις εμφανίζονται σε πολλές κλιτικές μορφές. Για παράδειγμα, στα αγγλικά, το ρήμα 'to walk' μπορεί να εμφανιστεί ως 'walk', 'walked', 'walks' ή 'walking'. Η βασική μορφή, 'walk', που μπορεί κανείς να αναζητήσει σε ένα λεξικό, ονομάζεται λέμμα για τη λέξη [19][20].

Το **Stemming** είναι μια απλούστερη εκδοχή του Lemmatization που διαιρεί λέξεις από τις καταλήξεις τους με σκοπό να δημιουργήσει μια ρίζα τους, χωρίς να ενδιαφέρεται για την γραμματική σημασία της λέξης. Για παράδειγμα, Ένας αλγόριθμος Stemming μπορεί να μειώσει τις λέξεις fishing, fished και fisher στη ρίζα fish. Η ρίζα δεν χρειάζεται απαραίτητα να είναι λέξη, π.χ. argue, argued, argues, arguing και argus στη ρίζα argu [19][20].

Στην αναγνώριση βιβλιογραφικών αναφορών οι τεχνικές αυτές είναι ιδιαίτερα χρήσιμες καθώς:

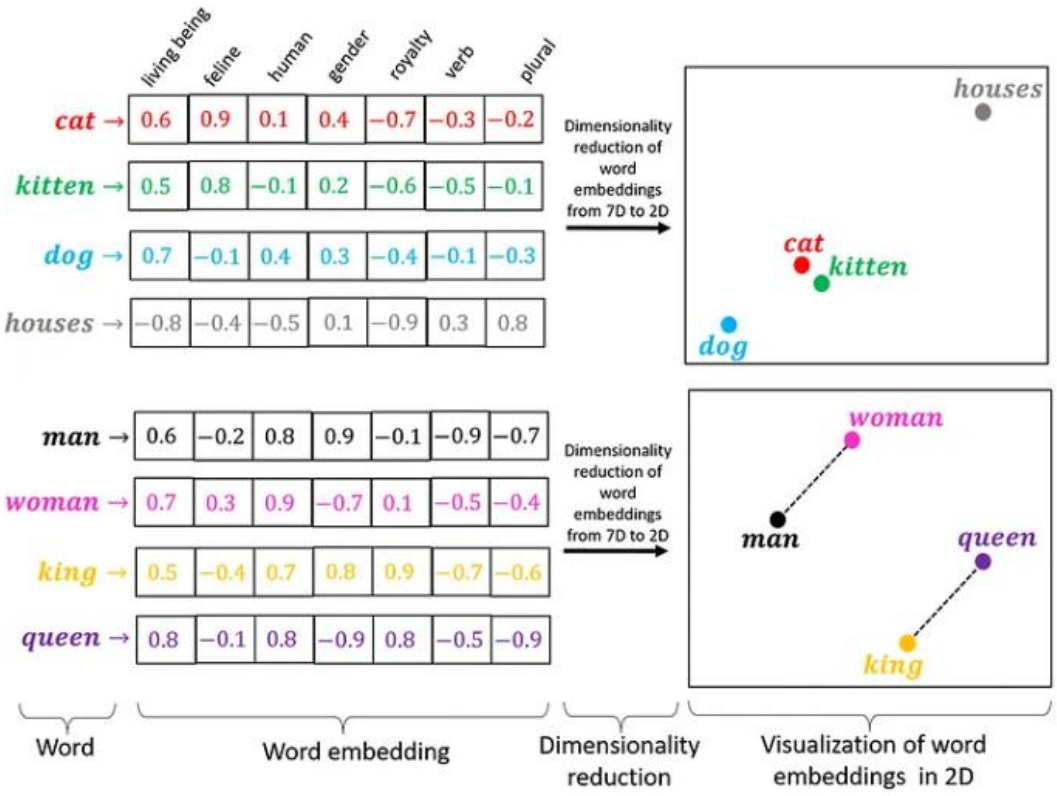
- Στην κανονικοποίηση, λέξεις που έχουν διαφορετικές μορφές αλλά ίδια σημασία, μπορούν να καταχωρίζονται ως ίδιες. (π.χ. "research" και "researching" καταχωρίζονται ως "research").
- Στην αναζήτηση πληροφοριών, όταν οι λέξεις κανονικοποιούνται, οι αναφορές μπορούν να εντοπιστούν πιο εύκολα από το σύστημα αναγνώρισης ακόμα και εάν αυτές οι αναφορές αναφέρονται με διαφορετικές καταλήξεις [19][20].



Εικόνα 4. Stemming vs Lemmatization [21].

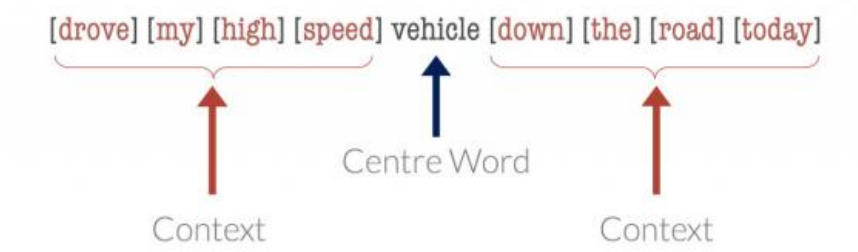
2.3.3.6 Word Embeddings

Τα Word Embeddings αποτελούν μια τεχνική αναπαράστασης λέξεων με βασικό στόχο να μετατρέψουν τις λέξεις σε αριθμητικά διανύσματα πολλαπλών διαστάσεων, όπου οι λέξεις που έχουν παρόμοια σημασία τοποθετούνται πιο κοντά η μία στην άλλη στον διανυσματικό χώρο. Αυτή η μέθοδος επιτρέπει στα συστήματα τεχνητής νοημοσύνης να κατανοούν και την έννοια αλλά και το συναίσθημα που μεταφέρει μια λέξη [22], [23], [24].



Εικόνα 5. Διάγραμμα ροής των Word Embeddings [25].

Η ανάγκη για τα Word embeddings βασίζεται στην ιδέα ότι οι λέξεις που εμφανίζονται σε μεγάλη συχνότητα σε παρόμοια συμφραζόμενα τείνουν να έχουν παρόμοιο νόημα. Κάθε λέξη αναπαρίσταται από έναν μοναδικό διανυσματικό πίνακα μέσω της εκπαίδευσης των μοντέλων σε μεγάλα σύνολα δεδομένων όπως μυθιστορήματα, βιβλία και ειδήσεις. Μοντέλα όπως το Word2Vec και το GloVe, είναι υπεύθυνα για να εκπαιδεύουν τα embeddings λαμβάνοντας υπόψιν τα συμφραζόμενα των λέξεων [22], [23], [24].



Εικόνα 6. Συμφραζόμενα [26]**.**

Με την χρήση των embeddings, οι λέξεις που σχετίζονται με ακαδημαϊκές αναφορές, όπως «συγγραφέας», «τίτλος» και «έτος» καταγράφονται ως άμεσα σχετικά στον διανυσματικό χώρο, με αποτέλεσμα να διευκολύνεται ο εντοπισμός των βιβλιογραφικών στοιχείων. Επιπροσθέτως, συμβάλουν στην εξακρίβωση των εννοιών και των σχέσεων μεταξύ λέξεων που εμφανίζονται συχνά [22], [23], [24].

2.4 Βιβλιοθήκες

2.4.1 Βιβλιοθήκες NLP

Για την ανάλυση, κατανόηση και επεξεργασία δεδομένων φυσικής γλώσσας, οι βιβλιοθήκες NLP αποτελούν κεντρικό εργαλείο και διαδραματίζουν σπουδαίο ρόλο στην αυτόματη αναγνώριση και διαχείριση βιβλιογραφικών αναφορών. Οι βιβλιοθήκες αυτές προσφέρουν ένα ευρύ φάσμα εργαλείων και τεχνικών που μπορούν να χρησιμοποιηθούν πάνω στο αντικείμενο της εργασίας. Στις ενότητες που ακολουθούν, παρουσιάζονται οι πιο σημαντικές βιβλιοθήκες NLP με έμφαση στην συμβολή τους στην αναγνώριση βιβλιογραφικών αναφορών.

2.4.1.1 SpaCy

Το spaCy είναι η πιο προηγμένη βιβλιοθήκη επεξεργασίας φυσικής γλώσσας (NLP), ανεπτυγμένη σε Python με κύριο σκοπό την γρήγορη και αποτελεσματική επεξεργασία δεδομένων. Λόγω της εξαιρετικής απόδοσης και των πολύ ακριβών προπαιδευμένων μοντέλων έχει γίνει ένα από τα πιο αξιόπιστα εργαλεία στον τομέα της επεξεργασίας φυσικής γλώσσας. Προσφέρει λειτουργίες όπως το tokenization, lemmatization, part-of-speech tagging και named entity recognition (NER) κάτι που το καθιστά ένα χρήσιμο εργαλείο για την εξαγωγή πληροφοριών από κείμενα [27], [28].

2.4.1.2 Hugging Face

Το Hugging Face δημιουργήθηκε το 2016 και είναι μια πλατφόρμα που έχει κάνει επανάσταση στον τρόπο με τον οποίο προσεγγίζουμε την τεχνητή νοημοσύνη και την μηχανική μάθηση. Λειτουργεί σαν ένα κοινωνικό δίκτυο που συμπεριλαμβάνει λάτρες της τεχνητής νοημοσύνης, ερευνητές και προγραμματιστές όπου μπορούν να συνεργάζονται μεταξύ τους και να μοιράζονται AI μοντέλα. Είναι φτιαγμένο να κάνει την τεχνητή νοημοσύνη πιο προσιτή, συνεργατική και ανοιχτή σε όλους. Για να εκπαιδευτεί ένα AI transformer μοντέλο απαιτείται να επενδυθεί ένα μεγάλο ποσό χρημάτων, κάτι το όποιο δεν είναι εφικτό για τις Start-ups και τις μικρές επιχειρήσεις. Το Hugging Face εκδημοκράτισε αυτή την διαδικασία και έδωσε πρόσβαση, δωρεάν, σε όλο τον κόσμο σε προεκπαιδευμένα μοντέλα [29], [30], [31].

2.4.2 Βιβλιοθήκες Μηχανικής Μάθησης

2.4.2.1 TensorFlow

Το TensorFlow δημιουργήθηκε από την Google Brain και είναι μια πλατφόρμα ανοιχτού κώδικα που αναπτύχθηκε για να εκπαιδεύει μοντέλα μηχανικής μάθησης. Επιπλέον, υποστηρίζει εφαρμογές και συστήματα μεγάλων δεδομένων(data centers) και επιτρέπει την δημιουργία αποδοτικών και ευέλικτων νευρωνικών δικτύων. Το TensorFlow στηρίζεται πάνω στην χρήση γραφημάτων ροής δεδομένων(data flow graphs) όπου οι ροές δεδομένων αναπαρίστανται ως ακμές και οι υπολογισμοί ως κόμβοι.

Το TensorFlow καθιστά δυνατή την εκτέλεση αλγορίθμων σε επεξεργαστές γραφικών (GPU), μέσω των γραφημάτων, βελτιώνοντας την ταχύτητα εκπαίδευσης του μοντέλου και την απόδοση του. Προσφέρει επίσης εργαλεία ανάλυσης δεδομένων, προσομοίωσης και οπτικοποίησης που διευκολύνουν την κατανόηση της λειτουργίας των δικτύων βαθιάς μάθησης. Επιπλέον, το TensorFlow συμπεριλαμβάνει προεκπαιδευμένα μοντέλα και API'S που καταφέρνουν και διευκολύνουν τον σχεδιασμό και την υλοποίηση εφαρμογών, όπως είναι η ανάλυση φυσικής γλώσσας, η επεξεργασία εικόνας και η αναγνώριση ομιλίας.

Στο τομέα της επεξεργασίας φυσικής γλώσσας (NLP) το TensorFlow επιτρέπει την δημιουργία chatbots και αυτόματων μεταφραστών και βελτιώνει την ακρίβεια και την απόδοση των μοντέλων σε εφαρμογές «σύστασης» όπως η διαφήμιση [32], [33], [34].

2.4.2.2 PyTorch

Το PyTorch αναπτύχθηκε από την Meta και όπως και το TensorFlow είναι υπεύθυνο για την ανάπτυξη και την εκπαίδευση μοντέλων βαθιάς μάθησης. Είναι ιδανικό για έρευνα και ανάπτυξη καθώς ξεχωρίζει για την αποδοτικότητα του και την υποστήριξη του να λειτουργεί και να συνεργάζεται με την GPU του υπολογιστή.

Το «dynamic computational graphs» είναι η πιο ισχυρή και σημαντική ιδιότητα του PyTorch. Επιτρέπει την δημιουργία και την επεξεργασία του γραφικού υπολογισμού κατά την εκτέλεση του προγράμματος(runtime) σε αντίθεση με το TensorFlow όπου το «computational graph» δημιουργείται πριν την εκτέλεση και δεν μεταβάλλεται. Αυτό, έχει σαν αποτέλεσμα κάθε φορά που εκτελείται ένας υπολογισμός, να πραγματοποιείται εκείνη την στιγμή(runtime) ξοδεύοντας όσους πόρους είναι απαραίτητο και προσαρμόζεται κάθε φορά στις απαιτήσεις των δεδομένων. Το PyTorch υποστηρίζει την χρήση GPU σε συνδυασμό με την τεχνολογία CUDA, με αποτέλεσμα να επιτυγχάνεται σημαντικά η απόδοση της εκπαίδευσης καθώς και η αξιολόγηση των μοντέλων [35], [36].

2.4.2.3 Scikit-learn

Η βιβλιοθήκη Scikit-learn είναι μια βιβλιοθήκη μηχανικής μάθησης, ανοιχτού κώδικα στην Python. Η βιβλιοθήκη Scikit-learn μπορεί να χρησιμοποιηθεί για επεξεργασία δεδομένων, μείωση διαστάσεων, ταξινόμηση, παλινδρόμηση, ομαδοποίηση και επιλογή μοντέλου, με τα αποτελέσματα αξιολόγησης να μπορούν να είναι σε μορφή χρόνου εκτέλεσης, ακρίβειας, πίνακα σύγχυσης, ποσοστού ψευδώς θετικών, ποσοστού ψευδώς αρνητικών, ακρίβειας, ανάκλησης και άλλων [37].

2.5 Προεκπαιδευμένα Μοντέλα

Τα προεκπαιδευμένα μοντέλα είναι υπεύθυνα να παρέχουν μια βάση για την ανάλυση και την κατανόηση μεγάλου όγκου γλωσσικών δεδομένων. Πιο συγκεκριμένα, έχουν εκπαιδευτεί σε άρθρα, κείμενα, συνομιλίες, προκειμένου να μάθουν τα μοτίβα και την δομή της κάθε γλώσσας. Αυτό έχει σαν αποτέλεσμα να μπορούν να εξάγουν και να αναγνωρίσουν πληροφορίες με υψηλή ακρίβεια και χωρίς να απαιτείται η εκπαίδευση από την αρχή.

Στο πλαίσιο της αναγνώρισης βιβλιογραφικών αναφορών, τα μοντέλα αυτά είναι ιδιαίτερα χρήσιμα, καθώς επιτρέπουν την αυτοματοποιημένη εξαγωγή στοιχείων. Επιπλέον, γλυτώνουμε χρόνο και πόρους διότι δεν τίθεται αναγκαίο να δημιουργηθούν νέα μοντέλα από το μηδέν και παρέχουν την δυνατότητα να προσαρμόζονται εύκολα σε πιο εξειδικευμένες εργασίες. Τα προεκπαιδευμένα μοντέλα χωρίζονται σε δύο κύριες κατηγορίες, τα παραδοσιακά και τα σύγχρονα μοντέλα [38], [39].

2.5.1 Παραδοσιακά μοντέλα

Τα παραδοσιακά μοντέλα είναι τα πρώτα προεκπαιδευμένα μοντέλα που χρησιμοποιήθηκαν για ανάλυση δεδομένων στον τομέα του NLP. Βασίζονται κυρίως σε στατιστικές τεχνικές και γραμμικούς ή μη γραμμικούς αλγόριθμους ταξινόμησης και παλινδρόμησης για την εξαγωγή των πληροφοριών από τα σύνολα δεδομένων. Τα παραδοσιακά μοντέλα δεν απαιτούν μεγάλη υπολογιστική ισχύ, γεγονός που τα καθιστά ιδανικά για εργασίες μεσαίας και μικρής κλίμακας. Στο πλαίσιο της αναγνώρισης βιβλιογραφικών αναφορών, τα μοντέλα αυτά, χρησιμοποιούνται κυρίως για ταξινόμηση και κατηγοριοποίηση των δεδομένων. Παρόλο που είναι αξιόπιστες λύσεις, η απόδοση τους περιορίζεται σε προβλήματα μεγάλης κλίμακας και σε προβλήματα που απαιτούν επεξεργασία μεγάλου όγκου δεδομένων και που απαιτούν κατανόηση περίπλοκων γλωσσικών σχέσεων καθώς η αρχιτεκτονική τους βασίζεται στην τεχνολογία feature engineering. Το feature engineering είναι μια διαδικασία και επικεντρώνεται στην επιλογή και δημιουργία χαρακτηριστικών (features) με σκοπό ένα μοντέλο να μάθει να κάνει προβλέψεις [38], [39].

2.5.1.1 K-Nearest Neighbors (KNN)

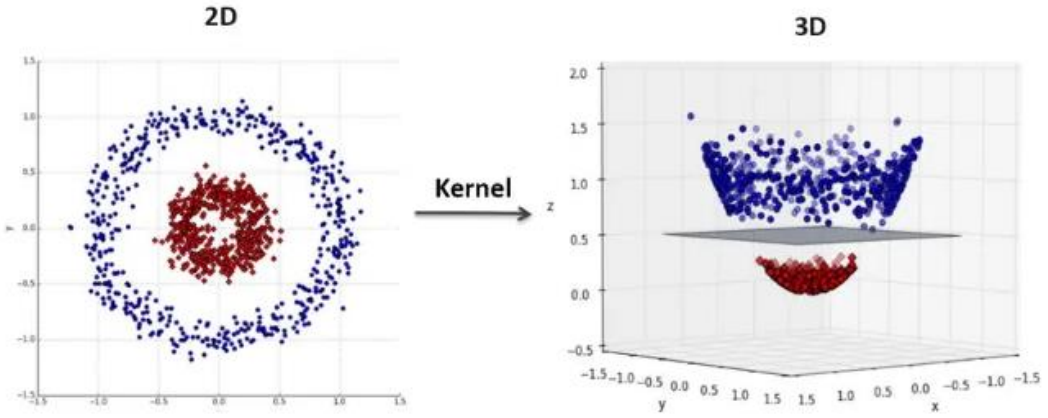
Ο αλγόριθμος KNN είναι ένας από τους πιο εύκολους αλγορίθμους μηχανικής μάθησης που χρησιμοποιούνται για διάφορα προβλήματα όπως η κατηγοριοποίηση και ταξινόμηση κειμένων, αναγνώριση προτύπων, ταξινόμηση εικόνων και η παρακολούθηση γεγονότων. Η ιδέα είναι να ταξινομήσουμε ένα άγνωστο αντικείμενο με βάση το επόμενο πιο κοντινό αντικείμενο στο σύνολο δεδομένων. Αυτό το κοντινότερο αντικείμενο σχηματίζει τη γειτονιά του άγνωστου αντικειμένου. Μια σημαντική παράμετρος του KNN είναι η επιλογή του K, το πλήθος των γειτόνων που λαμβάνονται υπόψη για την λειτουργία του μοντέλου. Ένα μικρό K μπορεί να προκαλέσει overfitting, ενώ ένα μεγάλο K μπορεί να μείωση την απόδοση του μοντέλου λόγω της ενσωμάτωσης μη σχετικών δεδομένων. Μια κατηγορία του αλγορίθμου είναι ο υπολογισμός του βάρους των γειτόνων. Για κάθε γείτονα στην γειτονιά, υπολογίζεται ένας συντελεστής, που ονομάζεται βάρος, για να ταξινομηθεί το

άγνωστο στοιχείο με μεγαλύτερη ακρίβεια. Το βάρος μπορεί να οριστεί ανάλογα με την απόσταση των γειτόνων από το άγνωστο αντικείμενο. Αυτή η λειτουργία έχει την δυνατότητα να βελτιώνει την ακρίβεια του μοντέλου ειδικά σε περιπτώσεις που τα δεδομένα μπορεί να είναι άνισα κατανεμημένα [40], [41].

Εικόνα 7. Διάγραμμα KNN [42]**.**

2.5.1.2 Support Vector Machines (SVM)

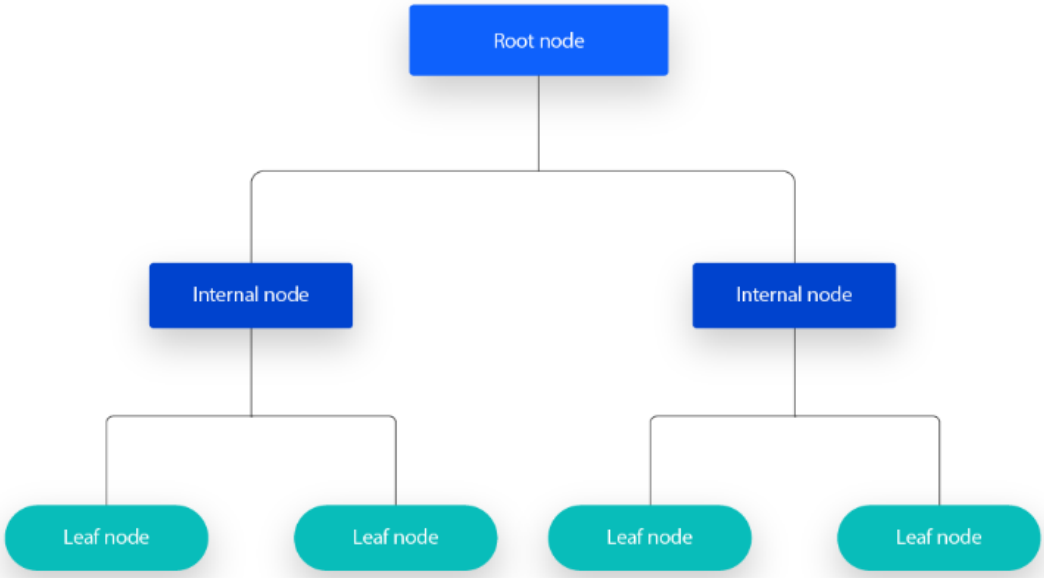
Το SVM είναι μια μέθοδος μηχανικής μάθησης που βασίζεται στην θεωρία Στατιστικής Μάθησης. Ξεπερνά την μεγάλη πολυπλοκότητα της δομής ενός νευρωνικού δικτύου και τα προβλήματα υπερπροσαρμογής(overfitting). Το μεγαλύτερο πλεονέκτημα του SVM είναι ότι μπορεί να διαχειρίζεται προβλήματα με μικρά δείγματα δεδομένων πολύ αποτελεσματικά και αυτό το καθιστά ιδιαίτερα χρήσιμο σε περιπτώσεις που τα δεδομένα είναι περιορισμένα. Εκτός από την εξαιρετική ικανότητα μάθησης σε μικρά δείγματα, έχει εξαιρετική αποτελεσματικότητα και έχει εφαρμοστεί αποτελεσματικά στην αναγνώριση προτύπων και στην εκτίμηση συναρτήσεων. Το SVM έχει καλύτερη γενίκευση από άλλα μοντέλα εφόσον βασίζεται στην αρχή της ελαχιστοποίησης του δοκίμου κινδύνου(structural risk minimization). Πιο συγκεκριμένα υπάρχουν δύο πλεονεκτήματα. Ελαχιστοποιεί το σφάλμα γενίκευσης και μεγιστοποιεί το περιθώριο για την ικανότητα γενίκευσης. Επιπλέον, το SVM είναι από τα πρώτα μοντέλα που έβαλαν σε λειτουργία τον πυρήνα “Kernel”, δηλαδή μια μέθοδο που πρώτα χαρτογραφεί τα σημεία εισόδου σε έναν υψηλής διάστασης χώρο χαρακτηριστικών μέσω μιας μη γραμμικής χαρτογράφησης και στη συνέχεια βρίσκει ένα βέλτιστο διαχωριστικό υπερεπίπεδο. Παρόλο που το SVM είναι αποτελεσματικό και ευέλικτο, σε μεγάλα σύνολα δεδομένων θα παρουσιάσει προβλήματα και προκλήσεις λόγο του υψηλού υπολογιστικού κόστους [43], [44], [45].



Εικόνα 8. Kernel in SVM [46].

2.5.1.3 Δέντρα Αποφάσεων (Decision Trees)

Τα δέντρα αποφάσεων είναι ένας από τους πιο δημοφιλείς αλγορίθμους μηχανικής μάθησης για ταξινόμηση και παλινδρόμηση. Πρόκειται για έναν μη παραμετρικό αλγόριθμο, δηλαδή δεν βασίζεται σε παραδοχές που είναι προκαθορισμένες για την κατανομή των δεδομένων. Έχει μια ιεραρχική δομή δέντρου η οποία αποτελείται από έναν κόμβο ρίζας, από κλάδους, εσωτερικούς κόμβους και φύλλα. Ένα δέντρο απόφασης ξεκινάει με έναν ριζικό κόμβο ο οποίος δεν έχει κανέναν εισερχόμενο κλάδο. Οι εξερχόμενοι κλάδοι του ριζικού κόμβου καταλήγουν σε εσωτερικούς κόμβους, γνωστοί και ως «κομβόι απόφασης». Μετέπειτα, βασιζόμενοι στα διαθέσιμα χαρακτηρίστηκα, διεξάγεται μια διαδικασία με σκοπό να σχηματίσουν ομοιογενή υποσύνολα τα οποία υποδεικνύονται από τα φύλλα κόμβους ή τους τερματικούς κόμβους. Οι κόμβοι αυτοί αντιπροσωπεύουν όλα τα πιθανά αποτελέσματα του συνόλου δεδομένων [47], [48], [49].



Εικόνα 9. Decision Tree [50]**.**

2.5.2 Σύγχρονα μοντέλα

Η πρόοδος στον τομέα της Βαθιάς Μάθησης οδήγησε στη δημιουργία των γλωσσικών σύγχρονων μοντέλων (Transformer). Αυτά τα μοντέλα είναι μια ισχυρή κατηγορία και έχουν αποδείξει την αποτελεσματικότητά τους σε πολλές εφαρμογές τεχνητής νοημοσύνης. Αυτά τα μοντέλα εισήχθησαν αρχικά για να λύσουν προβλήματα στον τομέα της Επεξεργασίας Φυσικής Γλώσσας (NLP), όπως η δημιουργία κειμένου και η αναγνώριση οντοτήτων. Το κύριο χαρακτηριστικό των Transformers είναι η ικανότητά τους να λαμβάνουν υπόψη τις σημασιολογικές εξαρτήσεις μεταξύ των λέξεων σε ένα κείμενο χωρίς τη χρήση παραδοσιακών αναδρομικών αρχιτεκτονικών. Αυτό επιτυγχάνεται μέσω μηχανισμών που επικεντρώνονται στην παράλληλη επεξεργασία πληροφοριών σε μεγάλες ακολουθίες δεδομένων, όπως ο Μηχανισμός Προσοχής(Attention Mechanism). Ο μηχανισμός αυτός, επιτρέπει στα μοντέλα να επικεντρώνονται στις σημαντικότερες λέξεις ενός κειμένου, ανεξάρτητα από την θέση στην οποία βρίσκονται στην ακολουθία, καθιστώντας τα μοντέλα αυτά ιδιαίτερα αποδοτικά και ευέλικτα. Επιπλέον, χάρη στην δομή του Transformer, επιτυγχάνεται η παράλληλη επεξεργασία των δεδομένων. Αυτή η δυνατότητα μειώνει αισθητά τον χρόνο εκπαίδευσης συγκριτικά με τα Long Short-Term Memory Networks(LTSMs) και τα RNNs και παράλληλα βελτιώνει και την ακρίβεια [51], [52].

2.5.2.1 BERT

Το Bidirectional Encoder Representations from Transformers(BERT) αναπτύχθηκε από την Google το 2018 και είναι γνωστό για την αμφίδρομη εκπαίδευση του. Πιο συγκεκριμένα, σε αντίθεση με τα παραδοσιακά μοντέλα, το BERT όταν αναλύει μια λέξη λαμβάνει υπόψη του και τα συμφραζόμενα της λέξης, τόσο τα προηγούμενα όσο και τα επόμενα. Με αυτόν τον τρόπο, ενισχύει την κατανόηση της πραγματικής σημασίας των λέξεων, γεγονός που βελτιώνει σημαντικά την ακρίβεια στις αναλύσεις. Είναι αξιοσημείωτο για τη δραματική του βελτίωση σε σύγκριση με τα προηγούμενα κορυφαία μοντέλα, και ως ένα πρώιμο παράδειγμα ενός μεγάλου γλωσσικού μοντέλου. Από το 2020, το BERT είναι μια πανταχού παρούσα βάση αναφοράς στα πειράματα επεξεργασίας φυσικής γλώσσας

(NLP). Το BERT εκπαιδεύεται μέσω της πρόβλεψης μάσκας κωδικών και της πρόβλεψης επόμενης πρότασης. Ως αποτέλεσμα αυτής της διαδικασίας εκπαίδευσης, το BERT μαθαίνει συμφραστικές, λανθάνουσες αναπαραστάσεις των tokens στο συμφραζόμενό τους. Η εκπαίδευση του μοντέλου BERT επηρεάζεται και αλλάζει η απόδοση και τα αποτελέσματα που θα προκύψουν από διάφορες παραμέτρους. Κάποιες από τις βασικότερες παραμέτρους είναι οι παρακάτω: [53].

- **Epochs**: Τα epochs ή αλλιώς οι εποχές, είναι ίσως η βασικότερη παράμετρος και είναι υπεύθυνα για να καθορίζουν τον αριθμό των διελεύσεων που θα πραγματοποιήσει το μοντέλο. Όσο μεγαλύτερος είναι ο αριθμός των Epochs τόσο πιο πολύ επιτρέπεται στο μοντέλο να μάθει καλύτερα τα δεδομένα. Ωστόσο, ένας μεγάλος αριθμός Epochs μπορεί να έχει και αρνητικά αποτελέσματα όπως «overfitting».
- **Batch Size**: Το Batch Size είναι υπεύθυνο για τον αριθμό των δειγμάτων που χρησιμοποιεί το μοντέλο σε κάθε του βήμα κατά την διάρκεια της εκπαίδευσης. Όσο μικρότερη είναι η τιμή του τόσο μπορεί να οδηγήσει σε μεγαλύτερη ακρίβεια άλλα απαιτούν περισσότερη δύναμη και χρόνο. Ωστόσο, ιδανικά πρέπει να γίνουν δοκιμές ώστε να βρεθεί ο καλύτερος αριθμός του Batch Size.
- **Class Weights**: Σε μια περίπτωση όπου τα δεδομένα είναι άνισα κατανεμημένα, οι κλάσεις μπορούν εξισορροπήσουν την κάθε κατηγορία.

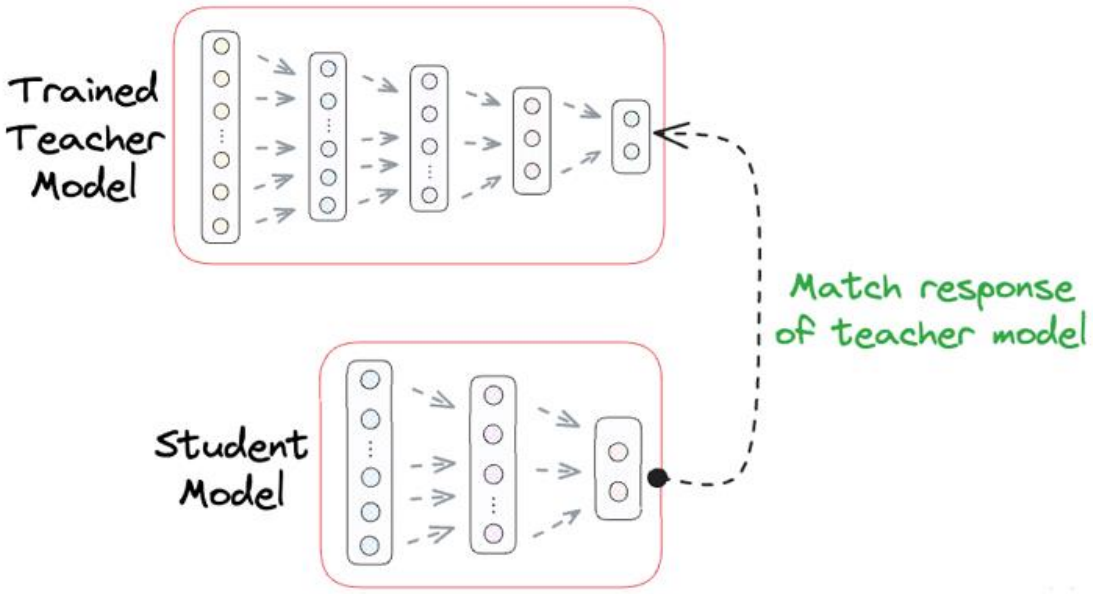
2.5.2.2 RoBERTa (Robustly Optimized BERT Pre-training Approach)

Το RoBERTa είναι μια βελτιωμένη έκδοση του BERT και δημιουργήθηκε για να ξεπεράσει μερικά από τα μειονεκτήματα της αρχικής τεχνικής εκπαίδευσης του BERT. Οι ερευνητές που το δημιούργησαν διαπίστωσαν πως το RoBERTa αποδίδει καλύτερα όταν αυξάνουν το μέγεθος των δεδομένων εκπαίδευσης και όταν μετατρέπουν παραμέτρους, όπως τα epochs και τα batches, επιτυγχάνουν καλύτερες αποδόσεις στο μοντέλο.

Η βασικότερη και πιο σημαντικής διαφορά ανάμεσα στο BERT και στο RoBERTa είναι η αφαίρεση του NSP(Next Sentence Prediction), το οποίο είναι για την εκπαίδευση του μοντέλου BERT ώστε να μια δεύτερη πρόταση ακολουθεί μιας πρώτης. Η αφαίρεση του NSP βελτιώνει την απόδοση του μοντέλου σε πιο ανόμοια σύνολα δεδομένων [54].

2.5.2.3 DistilBERT (Distilled BERT)

Το DistilBERT είναι μια πιο ελαφριά εκδοχή του μοντέλου BERT. Είναι ανεπτυγμένη χρησιμοποιώντας μια διαδικασία που ονομάζεται «knowledge distillation» η οποία διαδικασία μειώνει τις παραμέτρους του μοντέλου καθιστώντας το ταχύτερο χωρίς να μειώνει την ακρίβεια του. Στο «knowledge distillation», ένα μικρότερο μοντέλο(student model) εκπαιδεύετε από ένα μεγαλύτερο μοντέλο(teacher model), διατηρώντας τις πληροφορίες του αρχικού μοντέλου. Το DistilBERT παρέχει σχεδόν τις ίδιες επιδόσεις με το BERT εξοικονομώντας χρόνο και κόστος στις εφαρμογές του NLP και παρέχει λύσεις σε περιπτώσεις που ο υπολογιστικός πόρος είναι περιορισμένος [55], [56].



Εικόνα 10. Teacher-Student Models [57].

2.5.2.4 ALBERT (A Lite BERT)

Το ALBERT έχει σαν στόχο να μειώσει το μέγεθος των μοντέλων BERT αλλά και την υπολογιστική πολυπλοκότητα τους, διατηρώντας όσο το περισσότερο δυνατόν την ακρίβεια. Ένας τρόπος για να μειωθεί η πολυπλοκότητα είναι η παραγοντοποίηση των παραμέτρων στα embeddings. Στο κλασσικό μοντέλο BERT τα embeddings χρειάζονται μεγάλη υπολογιστική ισχύ, ενώ στο ALBERT, τα embeddings χωρίζονται μικρότερες και ανεξάρτητες παραμέτρους με αποτέλεσμα την καλύτερη και πιο αποδοτική διαχείριση των πόρων.

Ακόμη μία τεχνική βελτιστοποίησης στο ALBERT είναι το parameter-sharing των «layers». Το βασικό μοντέλο BERT σε κάθε «layer» χρησιμοποιεί ξεχωριστές παραμέτρους, σε αντίθεση με το ALBERT που επαναχρησιμοποιεί τις ίδιες στα διαδοχικά επίπεδα του νευρωνικού δικτύου. Αυτό έχει σαν αποτέλεσμα να μειώσει σημαντικά το μέγεθος των παραμέτρων χωρίς να επηρεαστεί η απόδοση [55].

2.5.2.5 BioBERT (Biomedical BERT)

Το BioBERT είναι μια εξειδικευμένη εκδοχή του BERT και ασχολείται αποκλειστικά με βιοϊατρικά κείμενα όπου επιστημονικοί όροι και τεχνικές απαιτούν ειδική κατανόηση και ακρίβεια. Βασίζεται στο βασικό μοντέλο BERT και αναπτύχθηκε περεταίρω μέσω της εκπαίδευσης του πάνω σε μεγάλα σύνολα δεδομένων βιοϊατρικού περιεχομένου. Η εξειδικευμένη εκπαίδευση, του επιτρέπει να αναγνωρίζει και να διαχειρίζεται σύνθετους βιοϊατρικούς όρους και να εκτελεί εργασίες όπως η αναγνώριση ορολογιών και η εξαγωγή οντοτήτων. Το BioBERT είναι ένα ζωτικό εργαλείο για την

επεξεργασία και αξιολόγηση εξειδικευμένων δεδομένων υγείας λόγω της εκτεταμένης χρήσης του στη βιοϊατρική έρευνα, των εξαιρετικών αποδόσεών του σε κλινικές και φαρμακευτικές μελέτες [58].

2.5.2.6 GPT

Το Generative Pre-trained Transformer(GPT) αναπτύχτηκε από την OpenAI και είναι μια σειρά προεκπαιδευμένων μοντέλων που ξεχωρίζουν για τις ικανότητες τους στην δημιουργία φυσικής γλώσσας. Βασίζεται στην αρχιτεκτονική των μετασχηματιστών (transformers), όπου και χρησιμοποιεί μηχανισμούς προσοχής (self-attention). Το GPT λαμβάνει υπόψη μόνο το προηγούμενο συμφραζόμενο της λέξης για να κατανοήσει το κείμενο και παρόλα αυτά, το GPT έχει παρουσιάσει εξαιρετικά αποτελέσματα και καθιστάτε ιδανικό για εφαρμογές παραγωγής κειμένου.

Η αρχιτεκτονική του GPT είναι ρυθμισμένη έτσι ώστε να μην αναλύει το μελλοντικό κείμενο, παρόλα αυτά, επιτυγχάνει υψηλή ακρίβεια στην παραγωγή κειμένου. Αυτή η αρχιτεκτονική το καθιστά ιδανικό για εφαρμογές που δημιουργούν περιεχόμενο όπως η αυτόματη συνομιλία και η δημιουργία ερωτήσεων (Brown et al., 2020). Μετά την εισαγωγή νεότερων μοντέλων, όπως είναι το GPT-3, έχει την δυνατότητα να μεταφράσει και να αναλύσει κείμενα καθιστώντας το πιο ευέλικτο [59].

2.6 Επίλογος

Το συγκεκριμένο κεφάλαιο παρείχε μια ολοκληρωμένη εικόνα από τις τεχνολογίες που ήδη υπάρχουν στον χώρο της Επεξεργασίας Φυσικής Γλώσσας (NLP) σχετικά με την αναγνώριση βιβλιογραφικών αναφορών. Αναφέρθηκαν και αναλύθηκαν τεχνικές του NLP όπως το Named Entity Recognition (NER), το tokenization, το stemming και το lemmatization, όπου είναι σημαντικές για την κατανόηση κειμένων. Έγινε και μια αναφορά στην βιβλιοθήκη spaCy και στο hugging Face, 2 τεχνολογίες που παρέχουν καταπληκτικά εργαλεία για την εφαρμογή των προαναφερόμενων τεχνικών.

Τέλος, στο κεφάλαιο αναφέρθηκαν κάποια από τα πιο σημαντικά και γνωστά μοντέλα μηχανικής μάθησης για την αναγνώριση κειμένων, όπως το K-Nearest Neighbors(KNN) και το Support Vector Machine (SVM), καθώς και μερικά από τα transformer μοντέλα όπως το BERT και το RoBERTa. Για την αναγνώριση πολύπλοκων γλωσσικών μοντέλων και για τον διαχωρισμό αναφορών από το υπόλοιπο κείμενο, χρησιμοποιήσαμε τα παραπάνω μοντέλα αφού έχουν αποδειχθεί εξαιρετικά.

Κεφάλαιο 3ο: Ανάπτυξη και αξιολόγηση

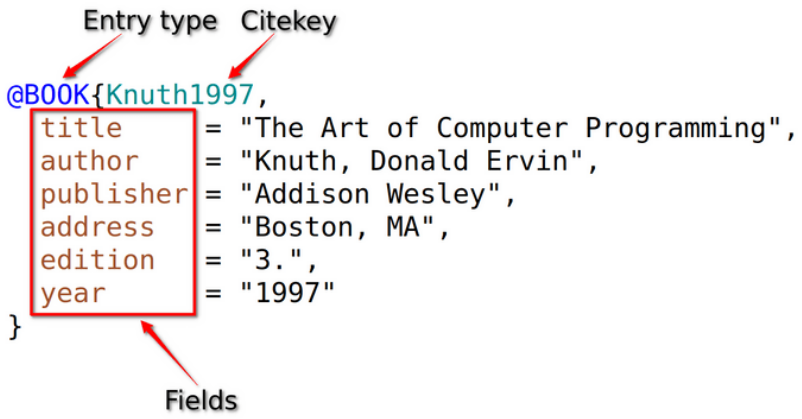
3.1 Εισαγωγή

Σε αυτό το κεφάλαιο, θα αναπτύξουμε ένα αυτοματοποιημένο σύστημα από προεκπαιδευμένα μοντέλα, με σκοπό να τα εκπαιδεύσουμε ώστε να είναι σε θέση να μπορούν να ξεχωρίζουν τις βιβλιογραφικές αναφορές από το υπόλοιπο περιεχόμενο ενός ακαδημαϊκού άρθρου. Αρχικά, θα παρουσιαστεί η διαδικασία δημιουργίας του dataset. Έπειτα, θα αναλυθούν τεχνικές και scripts με την γλώσσα python που χρησιμοποιήθηκαν για την επεξεργασία τους. Στην συνέχεια, θα συγκριθούν και εξεταστούν τα μοντέλα που επιλέχθηκαν για την αναγνώριση των αναφορών και θα παρουσιαστούν τα αποτελέσματα τους. Το κεφάλαιο τελειώνει με την αξιολόγηση και την σύγκριση των αποτελεσμάτων των μοντέλων και την συζήτηση για μελλοντικές επεκτάσεις της παρούσας εργασίας.

3.2 Δημιουργία και επεξεργασία Dataset

3.2.1 Συλλογή δεδομένων

Αρχικά, συλλέχθηκαν περίπου 10.000 βιβλιογραφικές αναφορές από την βάση δεδομένων Scopus, σε μορφή BibTex. Η εξαγωγή των αναφορών έγινε σε αυτή την μορφή διότι τα αρχεία BibTex παρέχουν οργανωμένες όλες τις πληροφορίες για τις αναφορές, όπως το όνομα του συγγραφέα, τον τίτλο του άρθρου, το έτος δημοσίευσής και το doi. Αυτές οι πληροφορίες είναι απαραίτητες ώστε να μετατρέψουμε τα δεδομένα στην μορφή βιβλιογραφικών αναφορών με την οποία δουλεύουμε, δηλαδή APA και MLA.



Εικόνα 11.Παράδειγμα BibTex [60]**.**

3.2.2 Μετατροπή Αναφορών

Για την διαδικασία της μετατροπής των αναφορών σε διαφορετικά στυλ (APA,MLA) αναπτύχθηκε ένα script γραμμένο στην python, από το μηδέν. Ο αρχικός σκοπός αυτού του script είναι να διαβάσει τα αρχεία BibTex που του δίνουμε και να εξάγει μόνο τα απαραίτητα στοιχεία και να τα μορφοποιήσει σύμφωνα με το στυλ της αναφοράς που χρειαζόμαστε.

Μετά την ολοκλήρωση της μετατροπής, σημαδέψαμε κάθε αναφορά με την ετικέτα 1. Αυτό γίνεται για να υποδείξουμε ότι πρόκειται για βιβλιογραφική αναφορά. Είναι απαραίτητη αυτή η προσέγγιση για να διαφοροποιηθούν οι βιβλιογραφικές αναφορές από το απλό κείμενό.

```
Salles, Arleen, Farisco, Michele. Neuroethics and AI ethics: a proposal for collaboration. *BMC Neuroscience*, vol. 25, no. 1, 2024.;1
Pira, Lirandë, Ferrie, Chris. On the interpretability of quantum neural networks. *Quantum Machine Intelligence*, vol. 6, no. 2, 2024.;1
```

Εικόνα 12. Παράδειγμα ετικέτας

Το «;» που παρατηρείται πριν από τον αριθμό 1, είναι ο διαχωριστής του csv αρχείου τον όποιο δηλώσαμε εμείς, με σκοπό να μπορεί να καταλάβει μελλοντικά το μοντέλο μας ποιο είναι η αναφορά και ποιο η ετικέτα.

3.2.3 Συλλογή απλών προτάσεων

Για να δημιουργηθεί ένα ισορροπημένο dataset, πέρα από τις αναφορές, χρειαζόμαστε και προτάσεις γενικού περιεχομένου. Για να εκπαιδευτεί σωστά το μοντέλο μας, χρησιμοποιήσαμε προτάσεις οι οποίες έχουν μεγάλη ποικιλία στο ύφος, στην σύνταξη και στην θεματολογία. Περιλάμβαναν προτάσεις από επιστημονικά άρθρα, καθημερινά κείμενα, βιβλία, ηλεκτρονικές πηγές και μη ακαδημαϊκά περιεχόμενα.

Μετέπειτα, αφού έγινε η συλλογή των ποικιλόμορφων προτάσεων, φτιάχτηκε ένα script στην python, το οποίο είχε σαν στόχο να προσθέσει την ετικέτα «0» στο τέλος κάθε πρότασης, μαζί με τον διαχωριστή «;», υποδεικνύοντας ότι πρόκειται για πρόταση και όχι για βιβλιογραφική αναφορά.

```
Artificial Intelligence (AI) has become an integral part of modern society.;0
From self-driving cars to virtual assistants, AI continues to reshape industries.;0
Have you ever wondered how these systems learn to perform their tasks?;0
At the core of AI lies machine learning, a process where algorithms improve through experience.;0
Imagine this: a robotic arm in a factory precisely assembles a product without human intervention.;0
```

Εικόνα 13.Παράδειγμα προτάσεων

3.2.4 Δημιουργία τελικού dataset

Το τελικό dataset με το οποίο θα εκπαιδεύσουμε το μοντέλο μας, διαμορφώθηκε με σκοπό να εξασφαλίζεται η ισορροπία και η ποικιλία των δεδομένων. Κάθε γραμμή αποτελείται από δύο πεδία:

- Την αναφορά ή την πρόταση
- Την ετικέτα (0 ή 1), που υποδεικνύει αν ανήκει σε βιβλιογραφική αναφορά ή όχι.

Επόμενο βήμα είναι να το χωρίσουμε σε δύο κατηγορίες. Το training set και το testing set. Ουσιαστικά, θα δημιουργηθούν 2 ξεχωριστά dataset. Με το training set, το όποιο θα αποτελείται από το 80% του συνολικού dataset, θα το χρησιμοποιήσουμε για την εκπαίδευση του μοντέλου. Το testing set, το οποίο αποτελείται από το υπόλοιπο 20% του συνολικού dataset, θα το χρησιμοποιήσουμε για να πραγματοποιήσουμε δοκιμές και να εξαγάγουμε αποτελέσματα. Ο διαχωρισμός γίνεται με τυχαίο τρόπο και με μερικούς περιορισμούς ώστε η κατανομή να είναι ομοιόμορφη.

```
# Ποσοστά (π.χ. 80%-20%)
sunolo = data.shape[0]
print(f"Δείγματα για εκπαίδευση: {len(train_texts)} ({len(train_texts) / sunolo:.0%})")
print(f"Δείγματα για έλεγχο: {len(test_texts)} ({len(test_texts) / sunolo:.0%})")
```

```
Δείγματα για εκπαίδευση: 8312 (80%)
Δείγματα για έλεγχο: 2079 (20%)
```

Εικόνα 14. Διαχωρισμός του dataset

3.2.5 Προεπεξεργασία δεδομένων

Για να διασφαλιστεί ένα ποιοτικό και καθαρό dataset, πρέπει να μεταβούμε σε μια επεξεργασία των δεδομένων, εφαρμόζοντας τα παρακάτω βήματα.

- Αφαίρεση ειδικών χαρακτήρων: Αφαιρέθηκαν μη απαραίτητοι χαρακτήρες, σύμβολα και αριθμοί που ενδέχεται να προκαλέσουν κάποιο σφάλμα στην ποιότητα και στην εκπαίδευση του μοντέλου
- Tokenization: Χωρίσαμε το κείμενο σε tokens.

3.2.6 Παρουσίαση dataset

Στον παρακάτω πίνακα θα δούμε μια μικρογραφία του dataset.

Κείμενο	**Ετικέτα**
Artificial Intelligence (AI) has become an integral part of modern society.	0
From self-driving cars to virtual assistants, AI continues to reshape industries.	0
Martinez-Martin, Nicole. A broader approach to ethical challenges in digital mental health. *World Psychiatry*, vol. 23, no. 3, 2024, pp. 394 – 395.	1
I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.	1

Το τελικό dataset θα αποτελέσει την βάση για την εκπαίδευση και την δοκιμή των μοντέλων NLP, όπως το BERT, SVM και το KNN των οποίων τα αποτελέσματα των πειραμάτων θα παρουσιαστούν στις παρακάτω ενότητες.

3.3 Αποτελέσματα μοντέλων

Σε αυτή την ενότητα θα δούμε τα αποτελέσματα των μοντέλων σε συνδυασμό με τις σημαντικές παραμέτρους που μπορεί να επηρεάσουν την απόδοσή τους. Με την χρήση διαφορετικών συνδυασμών των παραμέτρων μπορούμε να πετύχουμε και διαφορετικά αποτελέσματα. Σκοπός μας

είναι να εντοπίσουμε τον καλύτερο συνδυασμό παραμέτρων ώστε να έχουμε και το καλύτερο δυνατό αποτέλεσμα.

3.3.1 KNN

Στο μοντέλο KNN, πειραματιστήκαμε με 3 παραμέτρους. Το πλήθος των γειτόνων (n_neighbors), στον οποίο χρησιμοποιήσαμε τις επιλογές 3,5,7,10,13. Την απόσταση των γειτόνων (weights) στην οποία χρησιμοποιήσαμε το uniform και το distance και τέλος, τον τύπο της μέτρησης απόστασης (metric) όπου χρησιμοποιήσαμε τα euclidean, manhattan, minkowski.

Πίνακας 1. Απόδοση για n_neighbors=3, weights=uniform, metric=euclidean.

	0	**1**	**Macro avg**	**Weighted avg**
Precision	0.98	0.98	0.98	0.98
Recall	0.79	1.00	0.90	0.98
F1-score	0.87	0.99	0.93	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9779653070792			

Πίνακας 2. Απόδοση για n_neighbors=3, weights=uniform, metric=manhattan.

	0	**1**	**Macro avg**	**Weighted avg**
Precision	0.49	1.00	0.74	0.95
Recall	0.97	0.89	0.93	0.90
F1-score	0.65	0.94	0.79	0.91
Support	207	1926	2133	2133
KNN Accuracy	0.8982653539615			

Πίνακας 3. Απόδοση για n_neighbors=3, weights=uniform, metric=minkowski.

	0	**1**	**Macro avg**	**Weighted avg**
Precision	0.98	0.98	0.98	0.98
Recall	0.79	1.00	0.90	0.98
F1-score	0.87	0.99	0.93	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9779653070792311			

Πίνακας 4. Απόδοση για n_neighbors=3, weights=distance, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.98	0.98	0.98	0.98
Recall	0.79	1.00	0.90	0.98
F1-score	0.87	0.99	0.93	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9779653070792311			

Πίνακας 5. Απόδοση για n_neighbors=3, weights=distance, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.49	1.00	0.74	0.95
Recall	0.97	0.89	0.93	0.90
F1-score	0.65	0.94	0.79	0.91
Support	207	1926	2133	2133
KNN Accuracy	0.8982653539615565			

Πίνακας 6. Απόδοση για n_neighbors=3, weights=distance, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.98	0.98	0.98	0.98
Recall	0.79	1.00	0.90	0.98
F1-score	0.87	0.99	0.93	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9779653070792311			

Πίνακας 7. Απόδοση για n_neighbors=5, weights=uniform, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.91	0.98	0.94	0.97
Recall	0.79	0.99	0.89	0.97
F1-score	0.85	0.98	0.91	0.97
Support	207	1926	2133	2133
KNN Accuracy	0.9718706047819972			

Πίνακας 8. Απόδοση για n_neighbors=5, weights=uniform, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.37	1.00	0.69	0.94
Recall	0.98	0.82	0.90	0.84
F1-score	0.54	0.90	0.72	0.87
Support	207	1926	2133	2133
KNN Accuracy	0.8391936240037506			

Πίνακας 9. Απόδοση για n_neighbors=5, weights=uniform, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.91	0.98	0.94	0.97
Recall	0.79	0.99	0.89	0.97
F1-score	0.85	0.98	0.91	0.97
Support	207	1926	2133	2133
KNN Accuracy	0.9718706047819972			

Πίνακας 10. Απόδοση για n_neighbors=5, weights=distance, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.97	0.98	0.97	0.98
Recall	0.79	1.00	0.89	0.98
F1-score	0.87	0.99	0.93	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9774964838255977			

Πίνακας 11. Απόδοση για n_neighbors=5, weights=distance, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.51	1.00	0.75	0.95
Recall	0.98	0.90	0.94	0.91
F1-score	0.67	0.95	0.81	0.92
Support	207	1926	2133	2133
KNN Accuracy	0.9057665260196905			

Πίνακας 12. Απόδοση για n_neighbors=5, weights=distance, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.97	0.98	0.97	0.98
Recall	0.79	1.00	0.89	0.98
F1-score	0.87	0.99	0.93	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9774964838255977			

Πίνακας 13. Απόδοση για n_neighbors=7, weights=uniform, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.91	0.99	0.95	0.98
Recall	0.87	0.99	0.93	0.98
F1-score	0.89	0.99	0.94	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9789029535864979			

Πίνακας 14. Απόδοση για n_neighbors=7, weights=uniform, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.18	1.00	0.59	0.92
Recall	1.00	0.52	0.76	0.56
F1-score	0.31	0.68	0.49	0.64
Support	207	1926	2133	2133
KNN Accuracy	0.5625879043600562			

Πίνακας 15. Απόδοση για n_neighbors=7, weights=uniform, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.91	0.99	0.95	0.98
Recall	0.87	0.99	0.93	0.98
F1-score	0.89	0.99	0.94	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9789029535864979			

Πίνακας 16. Απόδοση για n_neighbors=7, weights=distance, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.96	0.99	0.97	0.98
Recall	0.87	1.00	0.93	0.98
F1-score	0.911	0.99	0.95	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9840600093764651			

Πίνακας 17. Απόδοση για n_neighbors=7, weights=distance, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.48	1.00	0.74	0.95
Recall	1.00	0.88	0.94	0.89
F1-score	0.64	0.94	0.79	0.91
Support	207	1926	2133	2133
KNN Accuracy	0.8931082981715893			

Πίνακας 18. Απόδοση για n_neighbors=7, weights=distance, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.96	0.99	0.97	0.98
Recall	0.87	1.00	0.93	0.98
F1-score	0.91	0.99	0.95	0.9
Support	207	1926	2133	2133
KNN Accuracy	0.9840600093764651			

Πίνακας 19. Απόδοση για n_neighbors=10, weights=uniform, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.91	0.99	0.95	0.98
Recall	0.91	0.99	0.95	0.98
F1-score	0.91	0.99	0.95	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9821847163619315			

Πίνακας 20. Απόδοση για n_neighbors=10, weights=uniform, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.13	1.00	0.57	0.92
Recall	1.00	0.29	0.65	0.36
F1-score	0.23	0.45	0.34	0.43
Support	207	1926	2133	2133
KNN Accuracy	0.36099390529770276			

Πίνακας 21. Απόδοση για n_neighbors=10, weights=uniform, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.91	0.99	0.95	0.98
Recall	0.91	0.99	0.95	0.98
F1-score	0.91	0.99	0.95	0.9
Support	207	1926	2133	2133
KNN Accuracy	0.9821847163619315			

Πίνακας 22. Απόδοση για n_neighbors=10, weights=distance, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.98	0.99	0.98	0.99
Recall	0.86	1.00	0.93	0.99
F1-score	0.92	0.99	0.96	0.99
Support	207	1926	2133	2133
KNN Accuracy	0.9854664791373652			

Πίνακας 23. Απόδοση για n_neighbors=10, weights=distance, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.35	1.00	0.68	0.94
Recall	1.00	0.80	0.90	0.82
F1-score	0.52	0.89	0.71	0.86
Support	207	1926	2133	2133
KNN Accuracy	0.8223159868729489			

Πίνακας 24. Απόδοση για n_neighbors=10, weights=distance, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.98	0.99	0.98	0.99
Recall	0.86	1.00	0.93	0.99
F1-score	0.92	0.99	0.96	0.99
Support	207	1926	2133	2133
KNN Accuracy	0.9854664791373652			

Πίνακας 25. Απόδοση για n_neighbors=13, weights=uniform, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.93	0.99	0.96	0.98
Recall	0.87	0.99	0.93	0.98
F1-score	0.90	0.99	0.95	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9812470698546648			

Πίνακας 26. Απόδοση για n_neighbors=13, weights=uniform, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.13	1.00	0.56	0.92
Recall	1.00	0.26	0.63	0.33
F1-score	0.23	0.41	0.32	0.40
Support	207	1926	2133	2133
KNN Accuracy	0.3333333333333333			

Πίνακας 27. Απόδοση για n_neighbors=13, weights=uniform, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.93	0.99	0.96	0.98
Recall	0.87	0.99	0.93	0.98
F1-score	0.90	0.99	0.95	0.98
Support	207	1926	2133	2133
KNN Accuracy	0.9812470698546648			

Πίνακας 28. Απόδοση για n_neighbors=13, weights=distance, metric=euclidean.

	0	1	Macro avg	Weighted avg
Precision	0.98	0.99	0.99	0.99
Recall	0.87	1.00	0.94	0.99
F1-score	0.93	0.99	0.96	0.99
Support	207	1926	2133	2133
KNN Accuracy	0.986404125644632			

Πίνακας 29. Απόδοση για n_neighbors=13, weights=distance, metric=manhattan.

	0	1	Macro avg	Weighted avg
Precision	0.25	1.00	0.63	0.93
Recall	1.00	0.68	0.84	0.71
F1-score	0.40	0.81	0.61	0.77
Support	207	1926	2133	2133
KNN Accuracy	0.7135489920300047			

Πίνακας 30. Απόδοση για n_neighbors=13, weights=distance, metric=minkowski.

	0	1	Macro avg	Weighted avg
Precision	0.98	0.99	0.99	0.99
Recall	0.87	1.00	0.94	0.99
F1-score	0.93	0.99	0.96	0.99
Support	207	1926	2133	2133
KNN Accuracy	0.986404125644632			

3.3.1.1 Συμπεράσματα για KNN

1. Επιπτώσεις της απόστασης και του βάρους:
   - Με τα metrics euclidean και minkowski για n_neighbors=3 ή n_neighbors=5, το μοντέλο KNN πετυχαίνει αρκετά υψηλή ακρίβεια (~98%). Στην συγκεκριμένη περίπτωση με τις επιλεγμένες παραμέτρους, η επιλογή του βάρους δεν επηρεάζει σημαντικά την απόδοση του μοντέλου μας.
   - Αντιθέτως, με το metric manhattan έχουμε μια εμφανής πτώση της ακρίβειας σε όλα τα βάρη αλλά πιο έντονα στο uniform, μειώνοντας την ακρίβεια έως και 56% σε ορισμένες περιπτώσεις. Η επιλογή της απόστασης είναι καίρια για την σωστή απόδοση του μοντέλου.

2. Επίδραση της επιλογής «n_neighbors»:
    - Με τα euclidean και minkowski metrics, ο αριθμός των γειτόνων 5 και 7 βελτιώνει την ακρίβεια. Ο αριθμός 10 των γειτόνων παραμένει επίσης αποτελεσματικός με την ακρίβεια να φτάνει σχεδόν το 98%.
    - Ωστόσο, παρατηρείτε πτώση στην ακρίβεια όταν χρησιμοποιούνται περισσότεροι γείτονες με το manhattan metric. Αυτό σημαίνει ότι ο συγκεκριμένος συνδυασμός των παραμέτρων δεν ενδείκνυται για το συγκεκριμένο σύνολο δεδομένων.
3. Προσαρμογή του Μοντέλου στις Κατηγορίες:
    - Στην πολυπληθέστερη κατηγορία (ετικέτα 1) το μοντέλο αποδεικνύεται ότι αποδίδει καλύτερα στην ταξινόμηση της, εφόσον η τιμή της ανάκλασης και της ακρίβειας είναι πολύ υψηλή.
    - Έχουμε σχετικά χαμηλότερη απόδοση στην κατηγορία με ετικέτα 0, κάτι που υποδηλώνει ότι το μοντέλο δυσκολεύεται να λειτουργήσει με την μικρότερη κατηγορία. Αυτό μπορεί να οφείλεται σε ανισορροπία των δεδομένων.
4. Γενικά συμπεράσματα:
    - Η απόδοση του μοντέλου KNN επηρεάζεται σημαντικά από την επιλογή της μετρικής απόστασης και πιο συγκεκριμένα η manhattan φαίνεται να μην είναι η κατάλληλη για το συγκεκριμένο σύνολο δεδομένων.
    - Η καλύτερη επιλογή φαίνεται να είναι η euclidean ή minkowski και οι 3 με 7 γείτονες για το συγκεκριμένο μοντέλο και το συγκεκριμένο σύνολο δεδομένων.

3.3.2 SVM

Στο μοντέλο Support Vector Machine, πειραματιστήκαμε με 2 παραμέτρους. Τον πυρήνα kernel, στον οποίο χρησιμοποιήσαμε τις επιλογές: linear, rbf, poly και τον Regularization parameter C στον οποίο χρησιμοποιήσαμε τις επιλογές 0.1, 1, 10. Παρακάτω θα παρουσιαστούν οι όλοι συνδυασμοί με τις παραμέτρους και θα σχολιαστούν τα αποτελέσματα τους.

Πίνακας 31. Απόδοση για Kernel: linear, C: 0.1.

	0	**1**	**Macro avg**	**Weighted avg**
Precision	0.94	1.00	0.97	0.99
Recall	0.98	0.99	0.99	0.99
F1-score	0.96	1.00	0.98	0.99
Support	207	1926	2133	2133
SVM Accuracy	0.9920300046882325			

Πίνακας 32. Απόδοση για Kernel: linear, C:1.

	0	1	Macro avg	Weighted avg
Precision	0.99	1.00	0.99	1.00
Recall	0.97	1.00	0.98	1.00
F1-score	0.98	1.00	0.99	1.00
Support	207	1926	2133	2133
SVM Accuracy	0.9957805907172996			

Πίνακας 33. Απόδοση για Kernel: linear, C: 10.

	0	1	Macro avg	Weighted avg
Precision	0.99	1.00	0.99	0.99
Recall	0.96	1.00	0.98	0.99
F1-score	0.97	1.00	0.99	0.99
Support	207	1926	2133	2133
SVM Accuracy	0.9948429442100328			

Πίνακας 34. Απόδοση για Kernel: rbf, C: 0.1

	0	1	Macro avg	Weighted avg
Precision	0.95	1.00	0.97	0.99
Recall	0.96	0.99	0.98	0.99
F1-score	0.95	1.00	0.97	0.99
Support	207	1926	2133	2133
SVM Accuracy	0.9910923581809657			

Πίνακας 35. Απόδοση για Kernel: rbf, C: 1.

	0	1	Macro avg	Weighted avg
Precision	0.99	0.99	0.99	0.99
Recall	0.95	1.00	0.97	0.99
F1-score	0.97	1.00	0.98	0.99
Support	207	1926	2133	2133
SVM Accuracy	0.9943741209563994			

Πίνακας 36. Απόδοση για Kernel: rbf, C: 10.

	0	1	Macro avg	Weighted avg
Precision	0.99	0.99	0.99	0.99
Recall	0.95	1.00	0.97	0.99
F1-score	0.97	1.00	0.98	0.99
Support	207	1926	2133	2133
SVM Accuracy	0.9943741209563994			

Πίνακας 37. Απόδοση για Kernel: poly, C: 0.1.

	0	1	Macro avg	Weighted avg
Precision	1.00	0.91	0.96	0.92
Recall	0.11	1.00	0.55	0.91
F1-score	0.19	0.95	0.57	0.
Support	207	1926	2133	2133
SVM Accuracy	0.9132676980778247			

Πίνακας 38. Απόδοση για Kernel: poly, C: 1.

	0	1	Macro avg	Weighted avg
Precision	1.00	0.94	0.97	0.94
Recall	0.37	1.00	0.68	0.94
F1-score	0.54	0.97	0.75	0.93
Support	207	1926	2133	2133
SVM Accuracy	0.9385841537740272			

Πίνακας 39. Απόδοση για Kernel: poly, C: 10.

	0	1	Macro avg	Weighted avg
Precision	1.00	0.94	0.97	0.94
Recall	0.37	1.00	0.68	0.94
F1-score	0.54	0.97	0.75	0.93
Support	207	1926	2133	2133
SVM Accuracy	0.9385841537740272			

3.3.2.1 Συμπεράσματα για SVM

1. Kernel: Linear
   - Η ακρίβεια κυμαίνεται μεταξύ 99.2% και 99.6% πράγμα που καθιστά τον γραμμικό πυρήνα εξαιρετικό.
   - Για C=1 επιτυγχάνεται η υψηλότερη ακρίβεια (99.6%). Φαίνεται πως η τιμή του C επηρεάζει το αποτέλεσμα.
   - Η κλάση 0 έχει χαμηλότερα αποτελέσματα στα precision και recall αλλά παραμένουν ικανοποιητικά. Αυτό ευθύνεται στο γεγονός ότι είναι μικρότερη σε δεδομένα στο dataset
   - Στην κλάση 1, που είναι και η κυρίαρχη στο dataset, παρατηρούμε σχεδόν τέλειες τιμές, δηλαδή το μοντέλο μας με αυτές τις παραμέτρους ταξινομεί τέλεια την κυρίαρχη κατηγορία.
2. Kernel: RBF
   - Η ακρίβεια κυμαίνεται μεταξύ 99.1% έως και 99.4%, η οποία είναι εξίσου πολύ υψηλή
   - Για C=1 και C=10 το μοντέλο μας δίνει τα καλύτερα αποτελέσματα.
   - Η κλάση 0 είναι χαμηλότερη σε επιτυχία από την κλάση 1, η οποία ταξινομείται με πολύ μεγάλη ακρίβεια.
3. Kernel: Polynomial
   - Ο πολυωνυμικός πυρήνας δίνει με διαφορά τα χειρότερα αποτελέσματα σε σχέση με τους άλλους δυο.
   - Η ακρίβεια κυμαίνεται μεταξύ 91.3% έως 93.8%, με την χαμηλότερη να είναι για C=0.1.
   - Το μοντέλο δυσκολεύεται να εντοπίσει την κλάση 0, και αυτό φαίνεται από το χαμηλό recall με 11% έως 37% επιτυχία.
   - Αντιθέτως η κλάση 1 έχει πολύ καλή απόδοση.

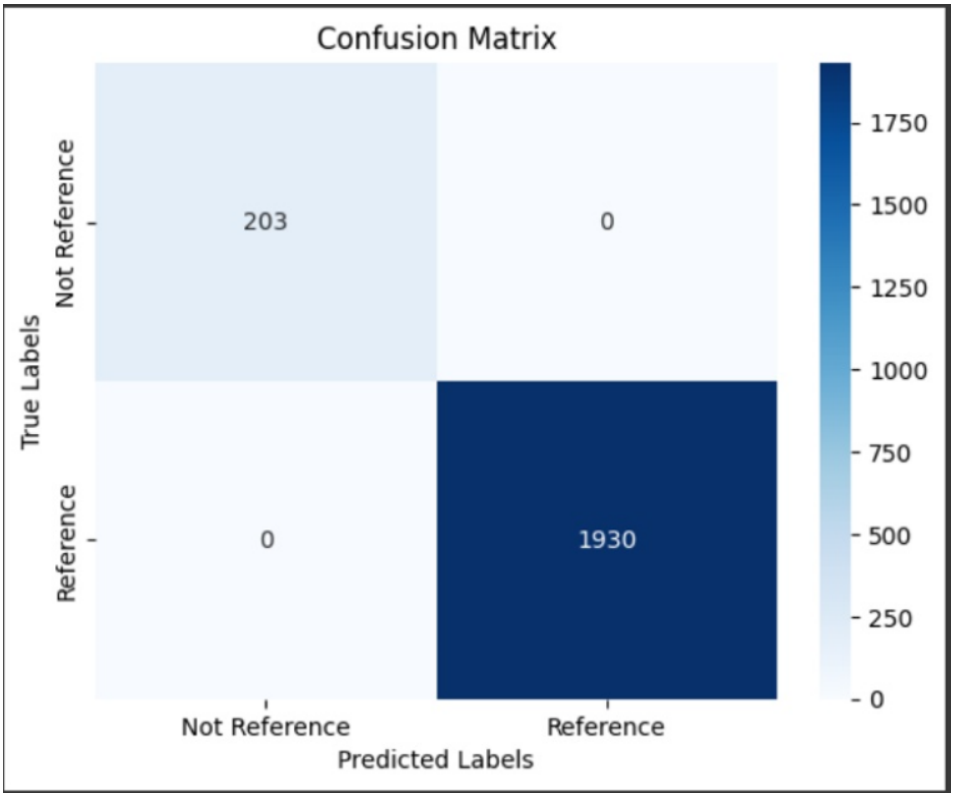
3.3.3 BERT

Στο μοντέλο BERT, για να εξετάσουμε την απόδοση του στις προβλέψεις, πειραματιστήκαμε με διάφορες παραμέτρους. Τον αριθμό των εποχών (Epochs) όπου και χρησιμοποιήσαμε τις επιλογές: 1,3,5,7. Η επόμενη παράμετρος που αλλάξαμε είναι το «Batch Size» με τις επιλογές: 8,16,32 και παράλληλα το «Class Weighting» το αφήσαμε σταθερά στο 2. Τέλος, σε μερικές περιπτώσεις αλλάξαμε και το μέγεθος με το οποίο διαχωρίζαμε το Dataset (Train-Test) από 70%-30% σε 80%-20%. Παρακάτω θα δούμε και θα αναλύσουμε έναν συνδυασμό από τον προαναφερόμενων παραμέτρων και θα σχολιαστούν τα αποτελέσματα με σκοπό να βρεθεί η καταλληλότερος συνδυασμός.

Πίνακας 40. Απόδοση για Epochs = 3, Batch Size = 16, Class = 2, Split = 80% Train + 20% Test

	0	1	Macro avg	Weighted avg
Precision	1:00	1.00	1:00	1:00
Recall	1:00	1:00	1:00	1:00
F1-score	1:00	1.00	1:00	1:00
Support	207	1930	2133	2133

Epoch	Training Loss	Validation Loss
1	0.045900	0.001050
2	0.000000	0.000016
3	0.000000	0.000013

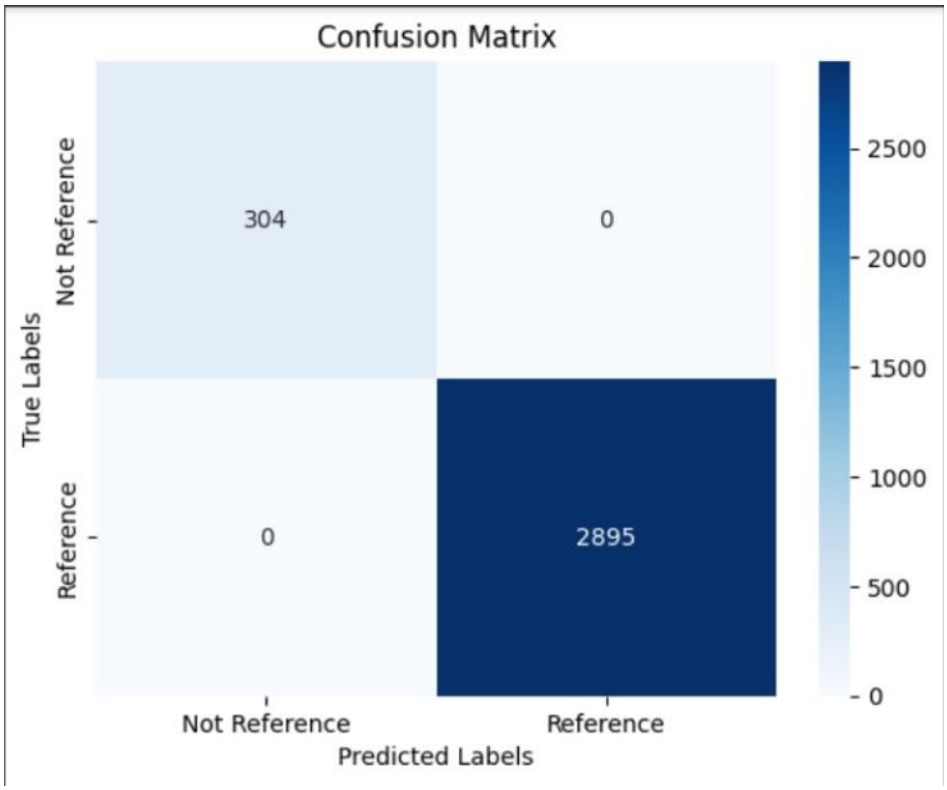


Εικόνα 15. Confusion Matrix για Epochs = 3, Batch Size = 16, Class = 2.

Πίνακας 41. Απόδοση για Epochs = 5, Batch Size = 32, Class = 2, Split = 70% Train + 30% Test

	0	1	Macro avg	Weighted avg
Precision	1:00	1.00	1:00	1:00
Recall	1:00	1:00	1:00	1:00
F1-score	1:00	1.00	1:00	1:00
Support	304	2895	3199	3199

Epoch	Training Loss	Validation Loss
1	0.000700	0.000382
2	0.000100	0.042938
3	0.000000	0.000025
4	0.000000	0.000017
5	0.000000	0.000015

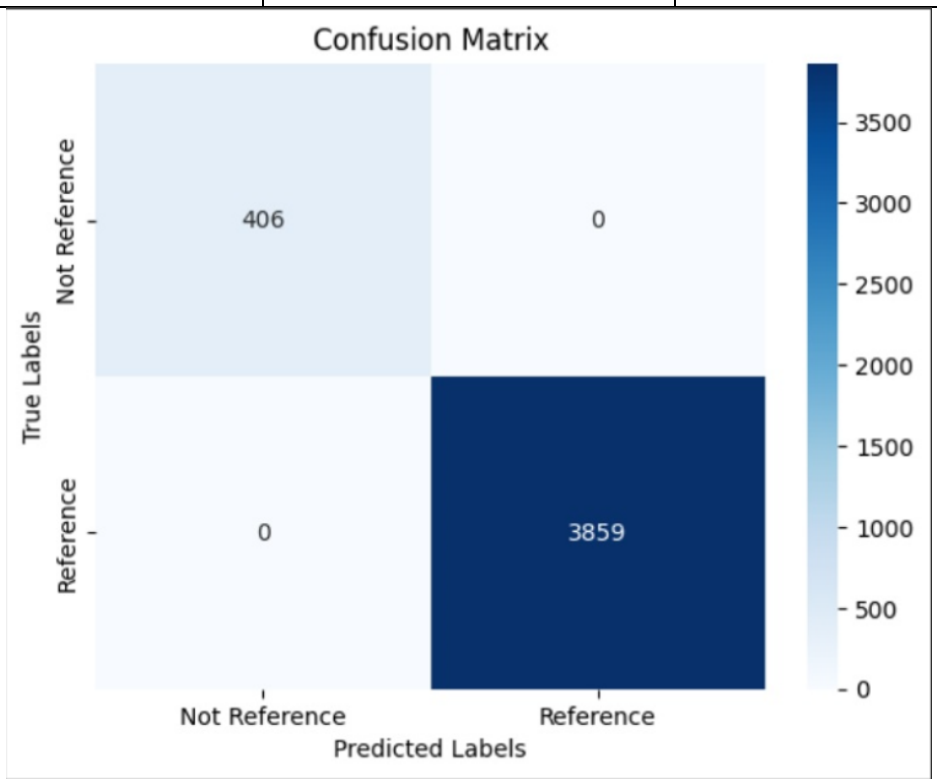


Εικόνα 16. Confusion Matrix για Epochs = 5, Batch Size = 32, Class = 2.

Πίνακας 42. Απόδοση για Epochs = 7, Batch Size = 8, Class = 2, Split = 60% Train + 40% Test

	0	1	Macro avg	Weighted avg
Precision	1:00	1.00	1:00	1:00
Recall	1:00	1:00	1:00	1:00
F1-score	1:00	1.00	1:00	1:00
Support	406	3859	4265	4265

Epoch	Training Loss	Validation Loss
1	0.000000	0.000029
2	0.003300	0.005100
3	0.000000	0.000006
4	0.000000	0.000003
5	0.000000	0.000002
6	0.000000	0.000002
7	0.000000	0.000002

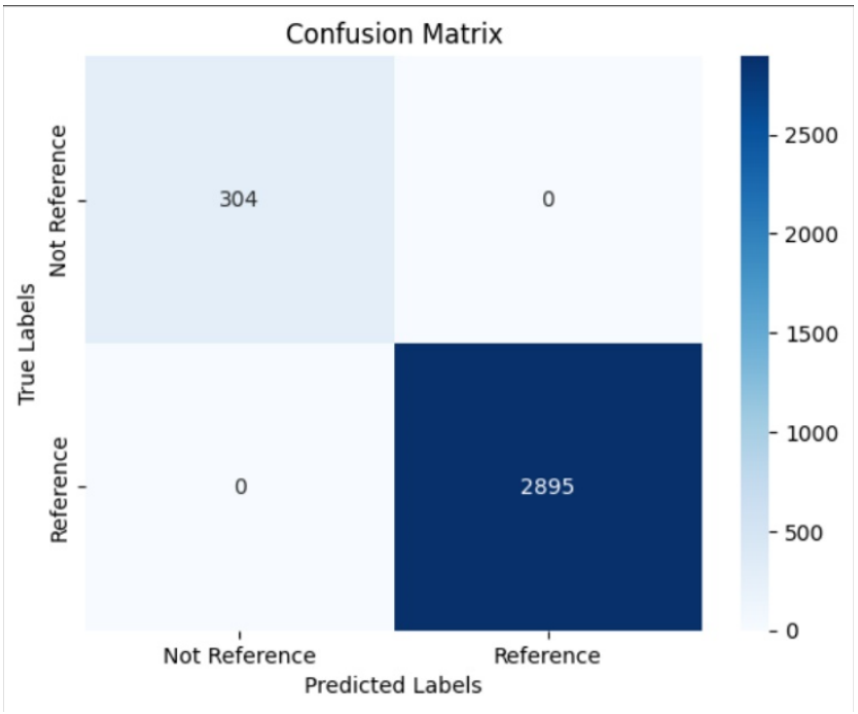


Εικόνα 17. Confusion Matrix για Epochs = 7, Batch Size = 8, Class = 2.

Πίνακας 43. Απόδοση για Epochs = 1, Batch Size = 16, Class = 2, Split = 70% Train + 30% Test

	0	1	Macro avg	Weighted avg
Precision	1:00	1.00	1:00	1:00
Recall	1:00	1:00	1:00	1:00
F1-score	1:00	1.00	1:00	1:00
Support	304	2895	3199	3199

Epoch	Training Loss	Validation Loss
1	0.044600	0.000395



Εικόνα 18. Confusion Matrix για Epochs = 1, Batch Size = 16, Class = 2.

3.3.3.1 Συμπεράσματα για BERT

- Πίνακας αξιολόγησης: Σέ όλα τα πειράματα με τις διαφορετικές παραμέτρους παρατηρείται ένα μοτίβο. Τα αποτελέσματα για την κατηγορία με ετικέτα 0 (Not Reference) και για την κατηγορία με ετικέτα 1(Reference) δείχνουν να είναι τέλεια με τις τιμές να είναι στο 1.00 για precision, recall και F1-score. Η ακρίβεια του μοντέλου είναι και αυτή στο 1.00 πράμα που σημαίνει ότι το μοντέλο ήταν ικανό να κατηγοριοποιήσει ορθά όλα τα παραδείγματα. Επιπλέον, το macro avg και το weighted avg είναι επίσης 1.00 υποδεικνύοντας ισορροπία μεταξύ των κατηγοριών. Τέλος, τα αποτελέσματα που έχουμε υποδηλώνουν ότι το μοντέλο BERT έχει άριστη ακρίβεια στην ταξινόμηση.
- Πίνακας «Training Loss» και «Validation Loss»: Η μείωση της απώλειας δηλώνει ότι το μοντέλο BERT κατάφερε και προσαρμόστηκε πολύ γρήγορα στο Dataset μας, γεγονός που σημαίνει ότι

έχει καταφέρει και έμαθε το σύνολο δεδομένων άριστα. Επιπλέον, το γεγονός ότι το «Validation Loss» είναι εξίσου χαμηλό δηλώνει ότι το μοντέλο λειτουργεί σωστά στο παρών σύνολο δεδομένων.

- Πίνακας «Confusion Matrix»: Από τον συγκεκριμένο πίνακα παρατηρούμε ότι το μοντέλο κατέγραψε και προέβλεψε σωστά όλα τα δείγματα που ανήκουν στην κατηγορία «Reference» και «Not Reference». Επιπλέον, σε κανένα από τα πειράματα μας δεν έχουμε False Positives(FP) ή False Negatives (FN) αφού το μοντέλο δεν έκανε λάθος σε κάποια πρόβλεψη.

3.4 Δοκιμή Μοντέλων σε Δεδομένα Εκτός του Dataset

Ο επόμενος στόχος μας, αφού εκπαιδεύσαμε και αξιολογήσαμε τα μοντέλα στα σύνολα δεδομένων «training» και «testing», είναι να επεκτείνουμε τις δοκιμές με σκοπό να αξιολογήσουμε την ικανότητα των μοντέλων να αναγνωρίζουν βιβλιογραφικές αναφορές σε πραγματικά άρθρα που δεν περιλαμβάνονται στο αρχικό dataset. Αυτή η προσέγγιση, έχει σαν στόχο να ελέγξει την απόδοση των μοντέλων σε άγνωστα δεδομένα. Για την αξιολόγηση, χρησιμοποιήσαμε αποσπάσματα κειμένου και βιβλιογραφικές αναφορές σε μορφή MLA και APA, από επιστημονικά άρθρα και βιβλία.

3.5 Επίλογος

Στο 3ο κεφάλαιο παρουσιάστηκαν και σχολιάστηκαν τα αποτελέσματα των μοντέλων που χρησιμοποιήσαμε, δηλαδή τα SVM, KNN και BERT. Πέρα από τις δοκιμές που έγιναν σε σύνολα δεδομένων εκπαίδευσης και δοκιμής έγιναν δοκιμές και σε δεδομένα εκτός του αρχικού dataset. Κάθε μοντέλο είχε διαφορετικά μειονεκτήματα και πλεονεκτήματα κάτι που κάνει την επιλογή του μοντέλου δύσκολο και υποκειμενικό ζήτημα. Συνολικά, είδαμε ξεκάθαρα ότι ενώ τα παραδοσιακά μοντέλα πρόσφεραν ικανοποιητικά αποτελέσματα και μια απλότητα, τα σύγχρονα μοντέλα όπως το BERT πρόσφεραν ανώτερα αποτελέσματα και είχαν την ικανότητα να διαχειριστούν σωστά μεγάλο όγκο δεδομένων, κάτι που είναι κρίσιμο για την βελτίωση της αξιοπιστίας των μοντέλων.

Κεφάλαιο 4ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

4.1 Συμπεράσματα

Συμπερασματικά, μπορούμε να πούμε με σιγουριά ότι το μοντέλο BERT, συγκριτικά με τα μοντέλα KNN και SVM, τα ξεπέρασε σημαντικά στην απόδοση να ξεχωρίσει τις βιβλιογραφικές αναφορές από το απλό κείμενο. Ωστόσο, πρέπει να σημειωθεί ότι αυτή η εξαιρετική απόδοση του BERT μπορεί να αποδίδεται στο περιορισμένο μέγεθος του dataset που χρησιμοποιήσαμε στην παρούσα εργασία. Το μοντέλο αυτό, είναι ικανό να διαχειρίζεται και να επεξεργάζεται έναν τεράστιο όγκο δεδομένων. Η απόδοση του σε ένα μικρό σύνολο δεδομένων ίσως δεν αντιπροσωπεύει στο έπακρο την πραγματικότητα της ικανότητας του μοντέλου σε σύνθετα σύνολα δεδομένων.

4.2 Προτάσεις βελτίωσης & μελλοντική έρευνα

4.2.1 Εμπλουτισμός του Dataset

Ο εμπλουτισμός του dataset είναι σημαντικός για την καλύτερη απόδοση του μοντέλου στην διαχείριση της αναγνώρισης βιβλιογραφικών αναφορών. Ο εμπλουτισμός μπορεί να γίνει με πολλές διαφορετικές προσεγγίσεις ώστε να ενισχυθεί και να ολοκληρωθεί το dataset.

- Προσθήκη γλωσσών: Το dataset μας είναι γραμμένο και λειτουργεί μόνο για την αγγλική γλώσσα. Μια αναβάθμιση των μοντέλων, θα ήταν η ένταξη αναφορών και κειμένων σε άλλες γλώσσες, ώστε να τα επιτρέψει να διαχειρίζονται παραπάνω δεδομένα. Για διεθνή ακαδημαϊκά άρθρα αυτό είναι σημαντικό και θα ενισχύσει την ικανότητα των μοντέλων ως προς την αναγνώριση αναφορών ανεξάρτητα από την γλώσσα στην οποία είναι γραμμένη μια αναφορά.
- Αύξηση του dataset: Ειδικά σε μοντέλα όπως το Bert, που είναι transformer μοντέλα, όσο μεγαλύτερος είναι ο όγκος του dataset τόσο καλύτερα αποδίδουν. Η προσθήκη περισσότερων προτάσεων και αναφορών θα ενισχύσει την ικανότητα των μοντέλων να αναγνωρίζουν τις αναφορές από το απλό κείμενο με μεγαλύτερη ακρίβεια. Η προσθήκη αναφορών και κειμένου από άρθρα, επιστημονικές εκθέσεις αλλά και από διαφορετικούς τομείς όπως η ιατρική, η μηχανική και διάφορες επιστήμες θα προσφέρει μεγαλύτερη ποικιλία.
- Προσθήκη Στυλ Αναφορών: Το τωρινό dataset μας αποτελείται από το APA και MLA στυλ βιβλιογραφικών αναφορών. Η προσθήκη περισσότερων στυλ αναφορών, όπως IEEE, Harvard, Chicago, θα βοηθήσει τα μοντέλα να αποκτήσουν την ικανότητα να αναγνωρίζουν περισσότερες αναφορές, καθώς κάθε στυλ έχει μοναδικές και διαφορετικές ιδιαιτερότητες.

4.2.2 Προσθήκη μοντέλων

Η προσθήκη και η δοκιμή καινούργιων μοντέλων μηχανικής μάθησης, θα μας παρέχει πληροφορίες για την αποτελεσματικότητα διαφορετικών μεθόδων στην αναγνώριση βιβλιογραφικών αναφορών. Κάποια μοντέλα μπορεί να είναι ιδιαίτερα αποτελεσματικά στην ανάλυση μεγάλου όγκου δεδομένων και να έχουν την δυνατότητα να διαχειρίζονται σωστά το πρόβλημα του overfitting. Στόχος μας είναι να βρούμε το ιδανικότερο μοντέλο για να κάνει την εργασία την οποία θέλουμε, δηλαδή να ξεχωρίζει πιο αποδοτικά τις βιβλιογραφικές αναφορές σε ένα άρθρο.

Η δοκιμή πολλαπλών μοντέλων είναι απαραίτητη διαδικασία διότι κάποια μοντέλα μπορεί να προσφέρουν βελτιωμένα αποτελέσματα σε πιο σύνθετα σύνολα δεδομένων, ενώ άλλα μοντέλα είναι

καταλληλότερα για περιπτώσεις όπου το σύνολο δεδομένων είναι πιο αυστηρό και περιορισμένο. Ωστόσο, αξίζει να σημειωθεί ότι η απόδοση κάθε μοντέλου μεταβάλλεται ανάλογα με τα δεδομένα και τις συγκεκριμένες απαιτήσεις της εφαρμογής. Κάποια από τα μοντέλα μπορεί να προσφέρουν βέλτιστα αποτελέσματα σε συγκεκριμένες περιπτώσεις γι' αυτό είναι απαραίτητη η προσεκτική αξιολόγηση και σύγκριση των μοντέλων σε πολλά σύνολα δεδομένων.

4.2.3 Αξιολόγηση σε πραγματικά δεδομένα

Κατά την διαδικασία της αξιολόγησης των μοντέλων σε πραγματικά δεδομένα, εμφανίστηκαν πολλές προκλήσεις που πρόβαλαν ευκαιρίες για περαιτέρω βελτίωση. Μια από τις προκλήσεις που συναντήσαμε είναι η δυνατότητα των μοντέλων να κρίνουν με ακρίβεια τα όρια μεταξύ πότε αρχίζει και πότε τελειώνει μια βιβλιογραφική αναφορά σε ένα πραγματικό άρθρο. Τα μοντέλα αδυνατούν να αναγνωρίσουν πότε τελειώνει η μία αναφορά και πότε ξεκινάει η επόμενη καθώς οι αναφορές είναι συνεχόμενες και δεν συνοδεύονται από κάποιο διαχωριστικό χαρακτήρα.

Αυτή η αδυναμία των μοντέλων ίσως μπορεί να λυθεί με την προσθήκη εξελιγμένων τεχνικών του NER(Named Entity Recognition), όπου θα βοηθήσουν τα μοντέλα να εντοπίσουν με μεγαλύτερη ακρίβεια το όριο μιας αναφοράς. Η προσθήκη των τεχνικών αυτών θα μπορούσαν να ενισχύσουν τα μοντέλα στο να εντοπίζουν και να ξεχωρίζουν τα συστατικά μιας αναφοράς, όπως το όνομα του συγγραφέα, το έτος έκδοσης και τον τίτλο του άρθρου. Έτσι, εφόσον γινόταν η σωστή διάκριση των συστατικών, θα μπορούσε το μοντέλο να εκπαιδευτεί ώστε όταν μια αναφορά τελειώνει στο έτος έκδοσης και η επόμενη λέξη που θα εντοπίζει θα είναι συγγραφέας να αντιλαμβάνεται ότι πρόκειται για δύο ξεχωριστές αναφορές. Μην ξεχνάμε βέβαια πως χρησιμοποιούμε πολλά στυλ βιβλιογραφικών αναφορών και δεν έχουν όλα το ίδιο μοτίβο. Γι' αυτό τον λόγο η εφαρμογή προχωρημένων τεχνικών του NER απαιτεί έναν τεράστιο όγκο δεδομένων εκπαίδευσης. Πιο συγκεκριμένα, χρειαζόμαστε εκατομμύρια πραγματικά δεδομένα ώστε το μοντέλο να αντιληφθεί τα διάφορα μοτίβα που υπάρχουν και χρησιμοποιούνται στις βιβλιογραφικές αναφορές. Η διαδικασία της συλλογής εκατομμυρίων πραγματικών δεδομένων είναι μια χρονοβόρα διαδικασία που απαιτεί πολλούς πόρους και την συνεργασία ειδικών στον τομέα της επεξεργασίας φυσικής γλώσσας και πολλών βιβλιογραφικών βάσεων δεδομένων. Επιπλέον, οι αναφορές από πραγματικά δεδομένα περιέχουν μεγάλη ποικιλία σε στυλ, καθώς περιλαμβάνουν μη τυποποιημένες βιβλιογραφικές αναφορές αλλά και λάθη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature 2015 521:7553*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[2] M. Gardner *et al.*, "AllenNLP: A Deep Semantic Natural Language Processing Platform," pp. 1–6, Jun. 2019, doi: 10.18653/v1/w18-2501.

[3] "(PDF) Artificial Intelligence Methods in Natural Language Processing: A Comprehensive Review." Accessed: Jan. 10, 2025. [Online]. Available: https://www.researchgate.net/publication/379523905_Artificial_Intelligence_Methods_in_Natural_Language_Processing_A_Comprehensive_Review

[4] X. Ma, "Application of artificial intelligence in computer network technology," in *2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*, IEEE, Jul. 2023, pp. 182–186. doi: 10.1109/AIARS59518.2023.00043.

[5] V. Vyas, K. Ravi, V. Ravi, V. Uma, S. Setlur, and V. Govindaraju, "Article citation study: Context enhanced citation sentiment detection." [Online]. Available: http://www.sciencedirect.com

[6] G. Alexandridis, I. Varlamis, K. Korovesis, G. Caridakis, and P. Tsantilas, "A survey on sentiment analysis and opinion mining in greek social media," *Information (Switzerland)*, vol. 12, no. 8, Aug. 2021, doi: 10.3390/info12080331.

[7] "Neural network (machine learning) - Wikipedia." Accessed: Jan. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Neural_network_(machine_learning)

[8] "Νευρωνικά Δίκτυα (Neural Networks): Ορισμός & Εφαρμογές." Accessed: Jan. 20, 2025. [Online]. Available: https://bigblue.academy/gr/neuronika-diktua

[9] F. N. A. Al Omran and C. Treude, "Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments," in *IEEE International Working Conference on Mining Software Repositories*, IEEE Computer Society, Jun. 2017, pp. 187–197. doi: 10.1109/MSR.2017.42.

[10] "Tokenizers - Hugging Face NLP Course." Accessed: Jan. 10, 2025. [Online]. Available: https://huggingface.co/learn/nlp-course/chapter2/4

[11] J. Lee *et al.*, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020, doi: 10.1093/bioinformatics/btz682.

[12] "Named-entity recognition - Wikipedia." Accessed: Jan. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Named-entity_recognition

[13] "What Is Named Entity Recognition? | IBM." Accessed: Jan. 10, 2025. [Online]. Available: https://www.ibm.com/think/topics/named-entity-recognition

[14] R. V. Siva Balan, K. Walia, and K. Gupta, “A Systematic Review on POS Tagging,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, IEEE, Apr. 2022, pp. 1531–1536. doi: 10.1109/ICACITE53722.2022.9823658.

[15] “Part-of-speech tagging - Wikipedia.” Accessed: Jan. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Part-of-speech_tagging

[16] “Part Of Speech Tagging – POS Tagging in NLP | byteiota.” Accessed: Jan. 20, 2025. [Online]. Available: https://byteiota.com/pos-tagging/

[17] “Regular expression - Wikipedia.” Accessed: Jan. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Regular_expression

[18] “Regular expressions - IBM Documentation.” Accessed: Jan. 10, 2025. [Online]. Available: https://www.ibm.com/docs/en/db2/11.5?topic=reference-regular-expressions

[19] R. Pramana, Debora, J. J. Subroto, A. A. S. Gunawan, and Anderies, “Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity,” in *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)*, IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/ICITDA55840.2022.9971451.

[20] “What Are Stemming and Lemmatization? | IBM.” Accessed: Jan. 10, 2025. [Online]. Available: https://www.ibm.com/think/topics/stemming-lemmatization

[21] “A Detailed Study on Stemming vs Lemmatization In Python.” Accessed: Jan. 20, 2025. [Online]. Available: https://www.turing.com/kb/stemming-vs-lemmatization-in-python

[22] “C8-2.” [Online]. Available: www.kaggle.com

[23] S. Chen, G. Chen, and W. Wang, “The joint effect of semantic and syntactic word embeddings on sentiment analysis,” in *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, IEEE, Sep. 2016, pp. 366–370. doi: 10.1109/ICNIDC.2016.7974598.

[24] “What Are Word Embeddings? | IBM.” Accessed: Jan. 10, 2025. [Online]. Available: https://www.ibm.com/think/topics/word-embeddings

[25] “Word Embedding: Basics. Create a vector from a word | by Hariom Gautam | Medium.” Accessed: Jan. 20, 2025. [Online]. Available: https://medium.com/@hari4om/word-embedding-d816f643140

[26] “Intro to Word Embeddings and Vectors for Text Analysis.” Accessed: Jan. 20, 2025. [Online]. Available: https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/

[27] A. K. Singh and A. Verma, “An Efficient method for Aspect Based Sentiment Analysis Using SpaCy and Vader,” in *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, IEEE, Jun. 2021, pp. 130–135. doi: 10.1109/CSNT51715.2021.9509650.

[28] “IEEE Xplore Full-Text PDF:” Accessed: Jan. 10, 2025. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7962368

[29] S. Misra, P. Tandon, and P. C. Panda, “Optimization of Hugging Face Transformers for Fake News Detection,” in *2023 International Conference on Data Science, Agents & Artificial*

Intelligence (ICDSAAI), IEEE, Dec. 2023, pp. 1–5. doi: 10.1109/ICDSAAI59313.2023.10452606.

[30] "Hugging Face – The AI community building the future." Accessed: Jan. 10, 2025. [Online]. Available: https://huggingface.co/

[31] "Hugging Face - Wikipedia." Accessed: Jan. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Hugging_Face

[32] S. S. Nandan Challapalli, G. Mishra, Y. Pachauri, A. Mishra, S. R. Kumar, and L. Kumar, "Comparing TensorFlow.js and TensorFlow in Python: An Accessibility and Usage Analysis," in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, Sep. 2023, pp. 250–254. doi: 10.1109/IC3I59117.2023.10397702.

[33] H. T. R. Adie, I. A. Pradana, and Pranowo, "Parallel Computing Accelerated Image Inpainting using GPU CUDA, Theano, and Tensorflow," in *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, Jul. 2018, pp. 621–625. doi: 10.1109/ICITEED.2018.8534858.

[34] L. Barba-Guaman, J. E. Naranjo, and A. Ortiz, "Object detection in rural roads using Tensorflow API," in *2020 International Conference of Digital Transformation and Innovation Technology (Incodtrin)*, IEEE, Oct. 2020, pp. 84–88. doi: 10.1109/Incodtrin51881.2020.00028.

[35] K. G. Kanagachidambaresann, "EAI/Springer Innovations in Communication and Computing Programming withhTensorFlow Solution forrEdge Computing Applications." [Online]. Available: http://www.springer.com/series/15427

[36] N. Ketkar and J. Moolayil, "Introduction to PyTorch," in *Deep Learning with Python*, Berkeley, CA: Apress, 2021, pp. 27–91. doi: 10.1007/978-1-4842-5364-9_2.

[37] "Scikit-Learn | SpringerLink." Accessed: Jan. 10, 2025. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-33383-0_5

[38] "Pre-trained models: Past, present and future - ScienceDirect." Accessed: Jan. 10, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000231

[39] "Pre-trained models for natural language processing: A survey | Science China Technological Sciences." Accessed: Jan. 10, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11431-020-1647-3

[40] T. Mladenova and I. Valova, "Comparative analysis between the traditional K-Nearest Neighbor and Modifications with Weight-Calculation," in *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, Oct. 2022, pp. 961–965. doi: 10.1109/ISMSIT56059.2022.9932693.

[41] "KNN Model-Based Approach in Classification | SpringerLink." Accessed: Jan. 10, 2025. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-39964-3_62

[42] "What is the k-nearest neighbors algorithm? | IBM." Accessed: Jan. 20, 2025. [Online]. Available: https://www.ibm.com/think/topics/knn

[43] V. Jakkula, "Tutorial on Support Vector Machine (SVM)".

[44] "Support vector machine - ScienceDirect." Accessed: Jan. 10, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/B9780128157398000067

[45] B. Zhang, J. Su, and X. Xu, “A Class-Incremental Learning Method for Multi-Class Support Vector Machines in Text Classification,” in *2006 International Conference on Machine Learning and Cybernetics*, IEEE, 2006, pp. 2581–2585. doi: 10.1109/ICMLC.2006.258853.

[46] “SVM kernels and its type. Support Vector Machines (SVMs) are a… | by Abhishek Jain | Medium.” Accessed: Jan. 20, 2025. [Online]. Available: https://medium.com/@abhishekjainindore24/svm-kernels-and-its-type-dfc3d5f2dcd8

[47] D. V. Patil and R. S. Bichkar, “A Hybrid Evolutionary Approach To Construct Optimal Decision Trees With Large Data Sets,” in *2006 IEEE International Conference on Industrial Technology*, IEEE, 2006, pp. 429–433. doi: 10.1109/ICIT.2006.372250.

[48] “Decision trees - de Ville - 2013 - WIREs Computational Statistics - Wiley Online Library.” Accessed: Jan. 10, 2025. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wics.1278

[49] “What are decision trees? | Nature Biotechnology.” Accessed: Jan. 10, 2025. [Online]. Available: https://www.nature.com/articles/nbt0908-1011

[50] “What is a Decision Tree? | IBM.” Accessed: Jan. 20, 2025. [Online]. Available: https://www.ibm.com/think/topics/decision-trees

[51] A. Gillioz, J. Casas, E. Mugellini, and O. A. Khaled, “Overview of the Transformer-based Models for NLP Tasks,” Sep. 2020, pp. 179–183. doi: 10.15439/2020F20.

[52] “How do Transformers work? - Hugging Face NLP Course.” Accessed: Jan. 10, 2025. [Online]. Available: https://huggingface.co/learn/nlp-course/chapter1/4

[53] A. Kampatzis, A. Sidiropoulos, K. Diamantaras, and S. Ougiaroglou, “Sentiment Dimensions and Intentions in Scientific Analysis: Multilevel Classification in Text and Citations,” May 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics13091753.

[54] “RoBERTa.” Accessed: Jan. 10, 2025. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta

[55] “Exploring transformer models for sentiment classification: A comparison of BERT, RoBERTa, ALBERT, DistilBERT, and XLNet - Areshey - 2024 - Expert Systems - Wiley Online Library.” Accessed: Jan. 10, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.13701

[56] “DistilBERT.” Accessed: Jan. 10, 2025. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/distilbert

[57] “Knowledge Distillation for Model Compression.” Accessed: Jan. 20, 2025. [Online]. Available: https://blog.dailydoseofds.com/p/knowledge-distillation-for-model

[58] “BioBERT: a pre-trained biomedical language representation model for biomedical text mining | Bioinformatics | Oxford Academic.” Accessed: Jan. 10, 2025. [Online]. Available: https://academic.oup.com/bioinformatics/article/36/4/1234/5566506

[59] “Generative pre-trained transformer - Wikipedia.” Accessed: Jan. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

[60] E. Gibney, “How ‘magic angle’ graphene is stirring up physics,” *Nature*, vol. 565, no. 7737, pp. 15–18, Jan. 2019, doi: 10.1038/D41586-018-07848-2).

ΠΑΡΑΡΤΗΜΑ Α : Κώδικας

Ο κώδικας της εφαρμογής βρίσκεται στο αποθετήριο: https://github.com/patakisGR/citation