Πρόγραμμα Μεταπτυχιακών Σπουδών

στην Αναλυτική των Επιχειρήσεων και Επιστήμη των Δεδομένων

Τμήμα Οργάνωσης και Διοίκησης Επιχειρήσεων

Διπλωματική Εργασία

Χρήση προ-εκπαιδευμένων μοντέλων επεξεργασίας φυσικής γλώσσας σε κλινικές σημειώσεις

Της

Αναστασίας Σισκοπούλου του Βασιλείου

Υποβλήθηκε ως απαιτούμενο για την απόκτηση του μεταπτυχιακού διπλώματος στην Αναλυτική των Επιχειρήσεων και Επιστήμη των Δεδομένων

Ημερομηνία

Μάρτιος 2023

Περίληψη

Η πρόοδος στον τομέα της τεχνητής νοημοσύνης έχει επηρεάσει βαθύτατα την ιατρική, καθώς οδηγεί σε ανανεωμένες δυνατότητες ανάλυσης των ηλεκτρονικών ιατρικών αρχείων. Ωστόσο, η παρουσία σημαντικών πληροφοριών σε μορφή ελεύθερου κειμένου δημιουργεί προκλήσεις στην αξιοποίηση αυτών των δεδομένων. Η ακριβής και έγκαιρη πρώιμη διάγνωση είναι κρίσιμη για την παροχή αποτελεσματικής ιατρικής περίθαλψης, και εδώ εισέρχεται η επεξεργασία φυσικής γλώσσας για την αυτοματοποίηση της διαδικασίας διάγνωσης. Μέσα από αυτό το πλαίσιο, η παρούσα έρευνα εστιάζεται στον αυτόματο προσδιορισμό ιατρικών εννοιών από τις σημειώσεις των εκπαιδευόμενων γιατρών κατά τις ιατρικές επισκέψεις. Στόχος είναι η επιτάχυνση της διαδικασίας μάθησης και αξιολόγησης των ασκούμενων ιατρών μέσω της μηχανικής μάθησης και επεξεργασίας φυσικής γλώσσας. Με επίκεντρο τα προ-εκπαιδευμένα μοντέλα Transformer και το μοντέλο BERT, η έρευνα επιδιώκει την καλύτερη κατανόηση και αξιοποίηση των ιατρικών δεδομένων για την ανάπτυξη ακριβέστερων διαγνώσεων και θεραπειών. Η μεθοδολογία που επιλέχθηκε ώστε να ελέγξουμε εάν το προ-εκπαιδευμένο μοντέλο BERT μπορεί να χρησιμοποιηθεί στην ιατρική κοινότητα και να βοηθήσει στην αυτοματοποίηση της διαδικασίας εκπαίδευσης των εκπαιδευόμενων ιατρών είναι η CRISP-DM. Η μελέτη επικεντρώθηκε μόνο στη χρήση του BERT, καθώς έχει επιδείξει εξαιρετικά αποτελέσματα σε προηγούμενες εργασίες της NLP, ενώ τα αποτελέσματα της έρευνας ήταν ικανοποιητικά, με το F1 score να φτάνει το 79,4%, ενώ το Recall και το Precision ήταν 82,6% και 76,4% αντίστοιχα.

Abstract

Progress in the field of artificial intelligence has deeply influenced medicine, as it leads to renewed capabilities in analyzing electronic medical records. However, the presence of significant information in free-text format poses challenges in utilizing this data. Accurate and timely early diagnosis is crucial for providing effective medical care, and this is where natural language processing comes in for automating the diagnostic process. Within this framework, the present research focuses on automatically identifying medical concepts from the notes of trainee doctors during medical visits. The goal is to accelerate the learning and evaluation process of trainee doctors through machine learning and natural language processing. Focused on pre-trained Transformer models and the BERT model, the research aims for a better understanding and utilization of medical data for developing more precise diagnoses and therapies. The methodology chosen to test whether the pre-trained BERT model can be used in the medical community and assist in automating the training process of trainee doctors is CRISP-DM. The study focused solely on the use of BERT, as it has demonstrated outstanding results in previous NLP tasks, while the research results were satisfactory, with an F1 score reaching 79.4%, while Recall and Precision were 82.6% and 76.4% respectively.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1- Εισαγωγή

Η ανάπτυξη της τεχνητής νοημοσύνης έχει επηρεάσει ευρέως διάφορους τομείς της πραγματικής ζωής, συμπεριλαμβανομένου και της ιατρικής. Οι ερευνητές έχουν πλέον στη διάθεση τους ηλεκτρονικά ιατρικά αρχεία τα οποία αναμένουν να βελτιώσουν τις ιατρικές υπηρεσίες. Ωστόσο, σημαντικές πληροφορίες που αναφέρονται στα ηλεκτρονικά ιατρικά αρχεία συχνά γράφονται σε ιατρικές σημειώσεις, για παράδειγμα το ιστορικό του ασθενούς, τα συμπτώματα, η νόσος που έχει διαγνωσθεί ο ασθενής, τα ιατρικά ευρήματα και άλλα, γεγονός που αποτελεί εμπόδιο στην αξιοποίηση προηγούμενων τεχνολογιών. Ένας ασθενής επηρεάζεται σημαντικά από την έγκαιρη και ακριβή διάγνωση σε πρώιμο στάδιο μιας νόσου. Οι εκπαιδευόμενοι γιατροί, καθώς δεν διαθέτουν εμπειρία, μπορεί να κάνουν λάθη όταν εντοπίζουν συμπτώματα και προσδιορίζουν τη διάγνωση. Εκτός αυτού, ο τεράστιος αριθμός συμπτωμάτων που μπορεί να εμφανίζονται σε διάφορες ασθένειες, προκαλεί σύγχυση στους επαγγελματίες της υγείας.

Η επεξεργασία φυσικής γλώσσας, η οποία αναλύει και οργανώνει αυτόματα πληροφορίες από ελεύθερο κείμενο, μπορεί να ξεπεράσει αυτό το εμπόδιο και να δώσει λύση στη διαδικασία της διάγνωσης με την αυτοματοποίηση διαφόρων διαδικασιών. Τέτοιες διαδικασίες είναι η αυτόματη εξαγωγή συμπτωμάτων, η αυτόματη ανίχνευση ασθενειών, η αυτόματη χαρτογράφηση των συμπτωμάτων και άλλες (Faris et al., 2022).

Τα προ-εκπαιδευμένα μοντέλα που βασίζονται στους Transformers έχουν ξεκινήσει μια νέα εποχή στη σύγχρονη επεξεργασία φυσικής γλώσσας. Έτσι, με την επιτυχία αυτών των μοντέλων στον γενικό τομέα, η ιατρική κοινότητα έχει αναπτύξει διάφορα μοντέλα τα οποία βασίζονται στο μοντέλο BERT (Bidirectional Encoder Representations from Transformers) και έχουν καταφέρει να αποτελούν την πρώτη επιλογή για κάθε εργασία στον ιατρικό κλάδο.

1.1 Περιγραφή προβλήματος

Η παρούσα έρευνα στοχεύει στον αυτόματο προσδιορισμό ιατρικών εννοιών που προέρχονται από τις σημειώσεις των εκπαιδευόμενων γιατρών για τους ασθενείς κατά τη διάρκεια επίσκεψης τους. Το σύνολο των δεδομένων έγινε διαθέσιμο από το Εθνικό Συμβούλιο Ιατρικών Εξεταστών (National Board of Medical Examiners-NBME), ενός οργανισμού που προσφέρει αξιολογήσεις και εκπαιδευτικές υπηρεσίες στους επαγγελματίες της υγείας, στους φοιτητές αλλά και σε άλλους φορείς. Σκοπός αυτού, η

συνεχής βελτίωση των γνώσεων των φορέων στις εξελισσόμενες ανάγκες της ιατρικής εκπαίδευσης και της υγειονομικής περίθαλψης.

Αναλυτικότερα, το πρόβλημα μας στοχεύει στην ανάγκη επιτάχυνσης της χρονοβόρας διαδικασίας μάθησης και αξιολόγησης των ασκούμενων ιατρών, η οποία προς το παρών εξαρτάται από τη συμβολή έμπειρων ιατρών. Κατά τη διάρκεια επίσκεψης ενός ασθενή σε έναν γιατρό, ο τρόπος με τον οποίο ερμηνεύει τα συμπτώματα του καθορίζει εάν η διάγνωση είναι ακριβής. Οι γιατροί μέχρι να αποκτήσουν άδεια άσκησης επαγγέλματος, έχουν κάνει αρκετή πρακτική να γράφουν σημειώσεις ασθενών που περιλαμβάνουν τα παράπονα τους, το ιατρικό ιστορικό, τα ευρήματα της φυσικής εξέτασης, τις πιθανές διαγνώσεις και τη φροντίδα παρακολούθησης. Η μηχανική μάθηση μαζί με την επεξεργασία φυσικής γλώσσας θα μπορούσαν να συμβάλουν στην επίτευξη της αυτοματοποίησης της παραπάνω διαδικασίας. Τα δεδομένα έγιναν διαθέσιμα με τη μορφή διαγωνισμού μέσα στη πλατφόρμα του Kaggle.

1.2 Σκοπός έρευνας

Ενώ οι εφαρμογές ή τεχνικές ποικίλλουν ανάλογα με το επιλεγμένο θέμα της επεξεργασίας φυσικής γλώσσας, σε αυτήν την έρευνα εξετάζουμε μια πιο ολοκληρωμένη κάλυψη εφαρμογών της και συμπεριλαμβάνουμε εργασίες που βασίζονται στα προ-εκπαιδευμένα μοντέλα Transformer και πραγματοποιούμε εκτενέστερη αναφορά στο μοντέλο BERT. Η βιβλιογραφία των παραπάνω καλύπτει τις μεθόδους βαθιάς μάθησης που υπάρχουν σε δεδομένα ιατρικού κειμένου, οι οποίες επιτυγχάνουν απόδοση στα επίπεδα ή κοντά στην τελευταία λέξη της τεχνολογίας. Στοχεύουμε να μελετήσουμε τον διαγωνισμό και τις διαφορετικές λύσεις που έχουν προκύψει αλλά και στην ανάπτυξη μιας δικιάς μας. Ειδικότερα, η ερευνητική διαδικασία την οποία θα ακολουθήσουμε, έχει σκοπό την εξαγωγή πληροφοριών με την αυτοματοποίηση του προσδιορισμού ιατρικών εννοιών μέσα από τις σημειώσεις που κρατάνε οι εκπαιδευόμενοι ιατροί κατά τη διάρκεια αλληλεπίδρασής τους με τους ασθενείς.

1.3 Τομέας προβλημάτων

Η έρευνα αυτή βασίζεται σε διάφορους κλάδους της επιστήμης των υπολογιστών, συμπεριλαμβανομένης της επεξεργασίας φυσικής γλώσσας και των προ-εκπαιδευμένων μοντέλων. Τα επιμέρους θέματα της μελέτης, όπως τα προ-εκπαιδευμένα μοντέλα Transformer και το μοντέλο BERT, αναλύονται περαιτέρω στους προαναφερθέντες

κύριους τομείς. Η παρακάτω ενότητα θα καλύψει αυτά τα θέματα έχοντας ως στόχο την κατανόηση των βασικών ιδεών.

1.3.1 Επεξεργασία Φυσικής Γλώσσας- Natural Language Processing (NLP)

Η επεξεργασία φυσικής γλώσσας είναι υποσύνολο της τεχνητής νοημοσύνης και αξιοποιεί τη μηχανική μάθηση για την εξέλιξη της. Πιο συγκεκριμένα, βοηθάει τους υπολογιστές στην κατανόηση των λέξεων ή προτάσεων που είναι γραμμένες σε ανθρώπινες γλώσσες. Παρόλη την εξέλιξη που έχει υπάρξει, η NLP αντιμετωπίζει προκλήσεις στην κατανόηση του σαρκασμού, σε σχήματα λόγου αλλά και σε τοπικές διαλέκτους (Rosett & Hagerty, 2021). Η μηχανική μετάφραση, η αναγνώριση ομιλίας, οι μηχανές αναζήτησης είναι μερικά παραδείγματα της NLP στον πραγματικό κόσμο. Αυτά προκύπτουν μέσα από τα πιο συνηθισμένα προβλήματα που έχει να αντιμετωπίσει η NLP: Ανάλυση συναισθήματος (Sentiment analysis), ταξινόμηση εγγράφων (Document classification), αυτόματη συμπλήρωση (Autocomplete), μετάφραση γλώσσας (Language translation), ταξινόμηση προθέσεων (Intent classification) (Raina & Krishnamurthy, 2022).

1.3.2 Transformer

Τον Δεκέμβριο του 2017, η Google AI δημοσίευσε μια έρευνα σχετικά με την μηχανική μετάφραση (machine translation), μέσα από την οποία έκανε την εμφάνιση της η αρχιτεκτονική δικτύου των προ-εκπαιδευμένων μοντέλων Transformer. Στην έρευνα αυτή, προσπάθησαν να βρουν μοντέλα που θα μπορούσαν να μεταφράσουν αυτόματα πολύγλωσσο κείμενο. Το Transformer βασίζεται στην προσοχή (attention), αντικαθιστώντας τα επαναλαμβανόμενα στρώματα που χρησιμοποιούνται πιο συχνά στις αρχιτεκτονικές κωδικοποιητή-αποκωδικοποιητή (encoder-decoder) με αυτό-προσοχή (self-attention). Αποδείχθηκε ότι για εργασίες μετάφρασης, μπορεί να εκπαιδευτεί πολύ πιο γρήγορα από τις προ υπάρχουσες αρχιτεκτονικές (Vaswani et al., 2017).

Στην συνέχεια όμως, ανακάλυψαν ότι μπορούν να χρησιμοποιήσουν την αρχιτεκτονική του Transformer σε ένα ευρύ φάσμα άλλων εργασιών, όπως επεξεργασία φυσικής γλώσσας, αναγνώριση ομιλίας κ.α.

Τα μοντέλα που βασίζονται στην αρχιτεκτονική του Transformer δημιουργούν διαφορικά βάρη που υποδεικνύουν ποιες λέξεις σε μια πρόταση είναι οι πιο κρίσιμες για περαιτέρω επεξεργασία. Για την πραγματοποίηση αυτού, ο μετασχηματιστής

(transformer) χρησιμοποιεί στοιβαγμένα επίπεδα τόσο για τον κωδικοποιητή όσο και για τον αποκωδικοποιητή:

- Κωδικοποιητής-Encoder: το μέρος αυτό επεξεργάζεται το δεδομένο εισόδου, που είναι κείμενο, αναζητά σημαντικά σημεία και δημιουργεί μια ενσωμάτωση (embedding) για κάθε λέξη με βάση τη συνάφεια με άλλες λέξεις της πρότασης.
- Αποκωδικοποιητής-Decoder: παίρνει το σημείο εξόδου του κωδικοποιητή, που είναι μια ενσωμάτωση, και την μετατρέπει ξανά σε έξοδο κειμένου,

Όπως περιγράφονται στο «Attention is all you need» (Vaswani et al., 2017).

1.3.3 Μοντέλο BERT

Το μοντέλο BERT προτάθηκε τον Οκτώβριο του 2018 από ερευνητές της Google AI. Αποτελεί ένα μοντέλο μηχανικής μάθησης για την επεξεργασία φυσικής γλώσσας το οποίο έχει εκπαιδευτεί χρησιμοποιώντας δεδομένα από το «Wikipedia» και το «Book Corpus», ένα σύνολο δεδομένων που περιέχει πάνω από 10 χιλιάδες βιβλία διαφορετικών ειδών. Είναι ένας αμφίδρομος μετασχηματιστής (transformer), που είναι προ-εκπαιδευμένος (pretrained), χρησιμοποιώντας έναν συνδυασμό πρόβλεψης επόμενης πρότασης (next sentence prediction) και αντικειμένων μοντελοποίησης γλώσσας με μάσκα (masked language modeling objective) (Devlin et al., 2019).

Το μοντέλο BERT έφερε επανάσταση στον χώρο της επεξεργασίας φυσικής γλώσσας δίνοντας λύση στα πιο κοινά προβλήματα της, όπως ανάλυση συναισθήματος, εξαγωγή συμπερασμάτων κ.α., και μάλιστα έχοντας καλύτερο αποτέλεσμα συγκριτικά με τα προηγούμενα μοντέλα.

Πιο λεπτομερείς πληροφορίες αναφορικά με την αρχιτεκτονική του μοντέλου BERT και τον τρόπο λειτουργίας θα γίνουν στο Κεφάλαιο 4.

1.4 Δομή της εργασίας

Το δεύτερο κεφάλαιο της διπλωματικής παρουσιάζει τις υπάρχουσες έρευνες που σχετίζονται με την επεξεργασία φυσικής γλώσσας, τα προ-εκπαιδευμένα μοντέλα Transformers και το μοντέλο BERT σε εφαρμογές ιατρικού περιεχομένου. Το κεφάλαιο 3 περιγράφει τη μεθοδολογία που χρησιμοποιήθηκε για την πρόβλεψη των σημαντικών χαρακτηριστικών σε κάθε σημείωμα ασθενούς. Στο κεφάλαιο 4 έγινε λεπτομερής αναφορά στον τρόπο ανάπτυξης του θέματος και στην ερμηνεία των αποτελεσμάτων. Συμπεράσματα και μελλοντικές προτάσεις που σχετίζονται με την διπλωματική θα περιγραφούν στο κεφάλαιο 5.

Κεφάλαιο 2- Ανασκόπηση της βιβλιογραφίας

2.1 Εισαγωγή

Όπως έχει ήδη αναφερθεί, η έρευνα αυτή βασίζεται σε διαφορετικούς τομείς της επιστήμης των δεδομένων. Οι αναγνώστες στο παρόν κεφάλαιο, θα έχουν τη δυνατότητα να λάβουν εμπεριστατωμένη γνώση σχετικά με την επεξεργασία φυσικής γλώσσας πάνω σε εφαρμογές της ιατρικής επιστήμης και για την ακρίβεια πάνω σε δεδομένα κειμένου και να προσδιορίσουν τα ερευνητικά κενά που υπάρχουν. Εν συνεχεία, θα πραγματοποιηθεί βιβλιογραφική επισκόπηση στα προ-εκπαιδευμένα μοντέλα Transformer και πιο λεπτομερείς αναφορά στο μοντέλο BERT και τις εφαρμογές του πάνω σε ιατρικά δεδομένα. Τα παραπάνω θα ωφελήσουν τον αναγνώστη στην εξοικείωση με τις έννοιες και τις πρακτικές που χρησιμοποιούνται με στόχο την καλύτερη κατανόηση των επόμενων κεφαλαίων.

2.2 Natural Language Processing σε ιατρικά δεδομένα

2.2.1 Εισαγωγή

Η επεξεργασία φυσικής γλώσσας εφαρμόζει τεχνικές που βασίζονται στη χρήση υπολογιστών για την ανάλυση και κατανόηση του λόγου ή του γραπτού κειμένου. Η NLP μπορεί να εφαρμοστεί σε διάφορους τομείς όπως αυτόματη μετάφραση (machine translation), κατηγοριοποίηση κειμένου (text categorization), φιλτράρισμα ανεπιθύμητων μηνυμάτων (spam filtering), εξαγωγή πληροφοριών (information extraction), συνόψιση κειμένου (summarization) και στην ιατρική (medicine) (Khurana et al., 2022). Περαιτέρω ανάλυση της χρήσης της NLP θα πραγματοποιηθεί πάνω στα ιατρικά δεδομένα (clinical data).

Η εφαρμογή ηλεκτρονικών αρχείων υγείας (Electronic Health Records- EHR) έχει καταστήσει στους ερευνητές άμεσα διαθέσιμα ψηφιακά δεδομένα για τεχνικές επεξεργασίας γλώσσας. Τα EHRs έχουν κατορθώσει να προωθήσουν την κλινική έρευνα και την παροχή υγειονομικής περίθαλψης. Αρκετές αναλύσεις έχουν εντοπίσει τον αυξανόμενο αριθμό ερευνητικών μελετών που έχουν πραγματοποιηθεί σχετικά με τις εφαρμογές της NLP στην ιατρική (Li et al., 2022; J. Liu et al., 2022; Locke et al., 2021). Οι πιο διαδεδομένες εφαρμογές περιλαμβάνουν ταξινόμηση (classification) και πρόβλεψη (prediction), μάθηση αναπαράστασης (representation learning), απάντηση ερωτήσεων (question answering), ιατρικοί διάλογοι (medical dialogues) και γενικότερα εφαρμογές στη δημόσια υγεία.

2.2.2 Προκλήσεις και δυσκολίες

Τα μεγάλα σώματα ιατρικών κειμένων είναι ταυτόχρονα ένας από τους πιο απαιτούμενους και ένας από τους λιγότερο προσβάσιμους πόρους στην επεξεργασία φυσικής γλώσσας στον ιατρικό κλάδο. Αυτό συμβαίνει κυρίως επειδή πρέπει να ληφθούν υπόψη τα ζητήματα εμπιστευτικότητας των ασθενών και του μεγάλου κόστους της δημιουργίας annotated δεδομένων (annotations: δεν υπάρχει επίσημη λέξη στα ελληνικά, στην ουσία είναι η κατηγοριοποίηση (categorization) και η επισήμανση (labeling) των δεδομένων για την εφαρμογή τους στην τεχνητή νοημοσύνη). Απόρροια αυτού, η καθυστερημένη πρόοδος της NLP στον ιατρικό τομέα σε σύγκριση με τον ευρύτερο τομέα της NLP.

Παρόλη τη δυνατότητα της NLP να μετριάσει την οικονομική επιβάρυνση στον ιατρικό κλάδο και να βελτιώσει την αξιοπιστία, το μόνο ελεύθερα διαθέσιμο μεγάλο σύνολο ιατρικών σημειώσεων είναι το MIMIC-III. Υπάρχουν και άλλα σώματα δεδομένων, όπως το CLEF, που περιέχει 565 χιλιάδες σημειώσεις, αλλά είναι προς το παρόν περιορισμένο και αναμένεται απόφαση από την κυβέρνηση για την διαθεσιμότητα του. Το Πανεπιστήμιο του Πίτσμπουργκ διαθέτει ένα σώμα που σχετίζεται με την κοινή εργασία TREC, αλλά διανέμεται μόνο στους συμμετέχοντες. Τέλος, η βάση δεδομένων eICU εξαιρεί το κείμενο ιατρικών σημειώσεων, με αιτία την προστασία των ιατρικών πληροφοριών (Yaneva et al., 2022). Οι εν λόγω προκλήσεις και οι δυσκολίες που εντοπίζονται στην NLP για την αξιοποίηση των ιατρικών κειμένων θα αναλυθούν παρακάτω.

2.2.2.1 Ιδιωτικότητα (privacy) των δεδομένων

Εξαιτίας της φύσης των πληροφοριών που περιέχονται στα EHRs κρίθηκε αναγκαία η ύπαρξη ρυθμιστικών νόμων όπως ο νόμος περί Ασφάλισης Υγείας και Φορητότητας και Λογοδοσίας (Health Insurance and Portability and Accountability Act- HIPAA). Αυτό είχε ως συνέπεια, προτού εκτελεστούν οποιεσδήποτε εργασίες ή κοινοποιηθούν δεδομένα, πρέπει να ληφθούν μέτρα για την διατήρηση του απορρήτου (Fernández-Alemán et al., 2013).

2.2.2.2 Προκαταλήψεις και υπερ-προσαρμογή κατά τη διάρκεια της εκπαίδευσης

Οι προβλέψεις των μοντέλων της NLP επηρεάζονται από τις προκαταλήψεις στα δεδομένα εκπαίδευσης. Τέτοιες προκαταλήψεις περιλαμβάνουν σύνολα δεδομένων εκπαίδευσης που δεν είναι πλήρως αντιπροσωπευτικά του πληθυσμού, δεδομένα που λείπουν ή είναι εσφαλμένα ταξινομημένα.

Η υπερ-προσαρμογή ενός μοντέλου μηχανικής μάθησης συμβαίνει όταν το μοντέλο μαθαίνει τις λεπτομέρειες και τον θόρυβο στα δεδομένα εκπαίδευσης στο βαθμό που επηρεάζει αρνητικά την απόδοση του σε νέα δεδομένα. Ωστόσο έχουν αναπτυχθεί αρκετές μέθοδοι για την μείωση της υπερ-προσαρμογής (Salman & Liu, 2019). Αυτό είναι συνέπεια τον περιορισμένων δεδομένων εκπαίδευσης, κάτι που είναι πιθανό στην ιατρική έρευνα λόγω και της ιδιωτικότητας.

2.2.2.3 Ασυνέπειες στη συγγραφή ιατρικών σημειώσεων

Οι ιατρικές σημειώσεις είναι μια μορφή ελεύθερου κειμένου διαμορφώνονται από ιατρικό προσωπικό, που κατανοούν τις αλληλεπιδράσεις με τους ασθενείς και τους αξιολογούν με διαφορετικό τρόπο. Παρόλο που υπάρχει μια καθιερωμένη δομή με την οποία γίνονται τα παραπάνω, συχνά υπάρχει διαφοροποίηση στα EHRs μεταξύ διαφορετικών ιατρικών ειδικοτήτων και επαγγελματιών υγείας. Ωστόσο, αυτή η τυποποίηση της ορολογίας που χρησιμοποιείται, έχει βοηθήσει στη μείωση του θορύβου και κατ' επέκταση στη μείωση της απώλειας του NLP μοντέλου (Häyrinen et al., 2008).

2.2.2.4 Έλλειψη από «annotations»

Αρκετά μοντέλα μηχανικής μάθησης είναι εποπτευόμενα μοντέλα και επομένως απαιτούν τα δεδομένα να είναι με ετικέτα (label) για την εκπαίδευση. Από την άλλη, για να πραγματοποιηθούν annotations στα δεδομένα που υπάρχουν από τα EHRs είναι μια πολύπλοκη, χρονοβόρα και μεγάλου κόστους διαδικασία. Επίσης, είναι δύσκολο να διασφαλιστεί η ποιότητα των annotations και εκτός αυτού τα περισσότερα annotated δεδομένα στον τομέα της ιατρικής επιστήμης είναι στα αγγλικά, καθιστώντας πιο δύσκολη την έρευνα σε άλλες γλώσσες (Li et al., 2022).

2.2.2.5 Ερμηνευσιμότητα

Ένα μοντέλο νευρωνικών δικτύων αποτελείται από ένα μεγάλο αριθμό παραμέτρων για εκπαίδευση, γεγονός που προκαλεί δυσκολίες στην ερμηνευσιμότητα του μοντέλου. Σε αντίθεση με τα γραμμικά μοντέλα που είναι πιο απλά και εύκολα στην εξήγηση, τα νευρωνικά δίκτυα αποτελούνται από μη γραμμικά επίπεδα και πολύπλοκες αρχιτεκτονικές που εμποδίζουν την ερμηνεία τους. Πρόσφατα, αναπτύχθηκαν μέθοδοι για την προσπάθεια εξήγησης των μοντέλων που παράγονται από τα νευρωνικά δίκτυα (Che et al., 2015; Mullenbach et al., 2018).

2.2.3 Εφαρμογές της NLP

Βάσει της ανασκόπησης της βιβλιογραφίας σχετικά με τις τεχνικές βαθιάς μάθησης σε δεδομένα που αφορούν κείμενο ιατρικού περιεχομένου (clinical text data), οι εφαρμογές της NLP ποικίλουν και στην πλειονότητα τους εκτελούνται πολύ κοντά στην τελευταία λέξη της τεχνολογίας. Αναφορικά μερικές αποτελούν ταξινόμηση και πρόβλεψη κειμένου, εξαγωγή πληροφοριών, εργασίες «γέννησης» αλλά και εφαρμογές ερώτησης- απάντησης.

2.2.3.1 Ταξινόμηση και πρόβλεψη κειμένου – Text Classification and Prediction

Για την γρηγορότερη επεξεργασία του μεγάλου όγκου κειμένων που υπάρχουν για την διεκπεραίωση εργασιών όπως υποστήριξη κλινικών αποφάσεων, έρευνα και βελτιστοποίηση διαδικασιών, η ταξινόμηση και η πρόβλεψη κειμένων κρίνεται απαραίτητη. Μερικές δευτερεύουσες εργασίες που καλύπτει η ταξινόμηση ιατρικών κειμένων και κλινικών σημειώσεων είναι ο προσδιορισμός μιας σειράς διαγνώσεων και διαδικασιών στη Μονάδα Εντατικής Θεραπείας (ΜΕΘ) (Marafino et al., 2014) όπως και ο προσδιορισμός του αρχικού λόγου για την αναζήτηση ιατρικής φροντίδας ή «chief complaint», που αφορά το ιατρικό ιστορικό του ασθενούς, τα παρόντα συμπτώματα και τον λόγο επίσκεψης (Valmianski et al., 2019).

Επιπρόσθετα, η ιατρική κωδικοποίηση, που αντιστοιχεί το κείμενο από τα EHRs σε κωδικούς Διεθνούς Ταξινόμησης Νοσημάτων (Huang et al., 2019) και η πρόβλεψη των ιατρικών αποτελεσμάτων, για παράδειγμα η διάρκεια παραμονής, εξέλιξη κατάστασης ασθενούς κλπ. (Lyu et al., 2018) αποτελούν σημαντικές εργασίες στην κατηγορία αυτή.

2.2.3.2 Εξαγωγή πληροφοριών- Information extraction

Η διαδικασία του αυτόματου εντοπισμού ενός σημαντικού περιεχομένου σε κείμενο μη δομημένης γλώσσας είναι γνωστή ως εξαγωγή πληροφοριών. Περιλαμβάνει έναν αριθμό σχετικών εργασιών, όπως η αναγνώριση ονομαστικών οντοτήτων (named entity recognition) (Cho & Lee, 2019), η σύνδεση οντοτήτων (entity linking), η οποία είναι χρήσιμη στην αυτόματη σύνδεση των δεδομένων με ιατρικές οντότητες και βοηθάει στη διάγνωση, στη λήψη αποφάσεων κ.α. (Chen et al., 2021) και η εξαγωγή σχέσης και γεγονότος (relation and event extraction), με το πρώτο να προσδιορίζει τις οντότητες που συνδέονται μέσω μιας σχέσης που ταιριάζει σε συγκεκριμένους τύπους και το δεύτερο στον εντοπισμό διαφορετικών γεγονότων (ShafieiBavani et al., 2020; Y. Zhang, Lin, et al., 2018).

2.2.3.3 Εργασίες «γέννησης»- Generation

Τα μοντέλα που έχουν δημιουργηθεί για εφαρμογές generation χωρίζονται σε τρεις κατηγορίες, στη δημιουργία ιατρικού κειμένου (clinical text generation), περίληψη (summarization) και μετάφραση ιατρικής γλώσσας (medical language translation). Μια εφαρμογή που εμπίπτει στην πρώτη κατηγορία είναι αυτή της δημιουργίας αυτοματοποιημένης ακτινολογικής αναφοράς (Alfarghaly et al., 2021). Στη δεύτερη, η περίληψη ευρημάτων της ακτινολογίας (Y. Zhang, Ding, et al., 2018) και στην τελευταία η απλοποίηση κειμένου με σκοπό την μετατροπή ιατρικών κειμένων σε ένα ύφος πιο εύκολα κατανοητό από τους κοινούς ανθρώπους (Weng et al., 2019).

2.2.3.4 Άλλες εφαρμογές

Σε αυτήν την ενότητα, θα εξεταστούν πρόσθετα θέματα στα οποία δίνεται λιγότερη έμφαση, αλλά είναι ωστόσο κρίσιμα για τα EHRs και άλλους σχετικούς τομείς. Ο σκοπός της εφαρμογής ερώτησης-απάντησης (Question Answering-QA) είναι να δώσει μια απάντηση στο ερώτημα που έχει τεθεί (Xia et al., 2022), ωστόσο τα σύνολα δεδομένων ιατρικής για QA είναι μικρού μεγέθους και η δημιουργία νέων συνόλων είναι απαγορευτική από πλευράς κόστους. Γι' αυτόν τον λόγο έχουν χρησιμοποιηθεί κυρίως τα προ-εκπαιδευμένα μοντέλα BERT (Yoon et al., 2020). Έπειτα, υπάρχει το «phenotyping» ασθενών που αποτελεί έναν συγκεκριμένο τομέα εφαρμογής των EHRs. Με βάση τις παραμέτρους που προσδιορίζουν την ιατρική κατάσταση και τα συμπτώματα ενός ασθενούς, ορίζονται οι φαινότυποι του ασθενούς. Ο σκοπός είναι να προσδιοριστεί με ακρίβεια εάν ένας ασθενής έχει μια συγκεκριμένη ιατρική ασθένεια ή εάν κινδυνεύει να αναπτύξει. Έτσι, οι (Yang et al., 2020), προέβλεψαν 10 διαφορετικούς φαινοτύπους ασθενών και είχαν την μεγαλύτερη επίδραση στον προσδιορισμό του φαινοτύπου για τον χρόνιο πόνο.

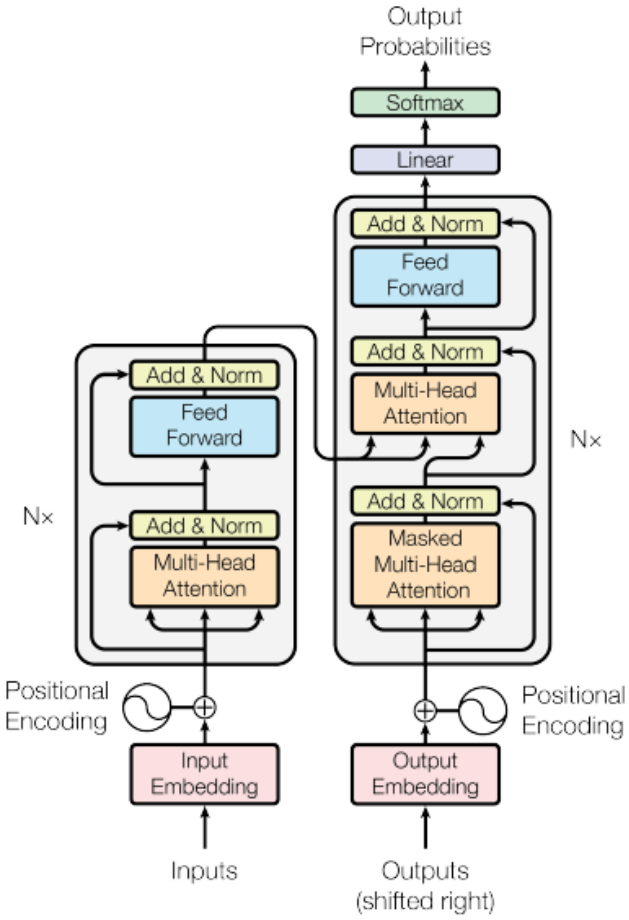
2.3 Προ-εκπαιδευμένα μοντέλα Transformer

Για την αντιμετώπιση των προκλήσεων της επεξεργασίας φυσικής γλώσσας έχουν χρησιμοποιηθεί διάφορα νευρωνικά δίκτυα όπως συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks- CNN), επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks- RNN), νευρωνικά δίκτυα που βασίζονται σε γραφήματα (Graph based Networks- GNNs) και μηχανισμοί προσοχής (attention mechanisms). Πρόσφατες έρευνες όμως, έχουν δείξει ότι τα προ-εκπαιδευμένα μοντέλα (Pre-trained models- PTMs) επεξεργασίας φυσικής γλώσσας έχουν σημειώσει καλύτερες αποδόσεις (Qiu et al., 2020).

Οι λόγοι για τους οποίους τα PTMs έχουν αντικαταστήσει τα προηγούμενα μοντέλα είναι καθώς η προ-εκπαίδευση πραγματοποιείται σε ένα μεγάλο σώμα κειμένου, με απόρροια τη μείωση του κόστους εκπαίδευσης και την αποφυγή υπερβολικής προσαρμογής (overfitting) σε μικρά σύνολα δεδομένων. Αυτό συμβαίνει καθώς η δημιουργία μεγάλων συνόλων δεδομένων με ετικέτα, που απαιτούνται για την εκπαίδευση των παραμέτρων και την αποφυγή του overfitting, αποτελεί μια πρόκληση για τις εργασίες της NLP. Πέρα από αυτό, παρέχεται καλύτερη προετοιμασία στο μοντέλο, η οποία οδηγεί σε καλύτερη γενίκευση του και γρηγορότερη επίτευξη του στόχου του προβλήματος.

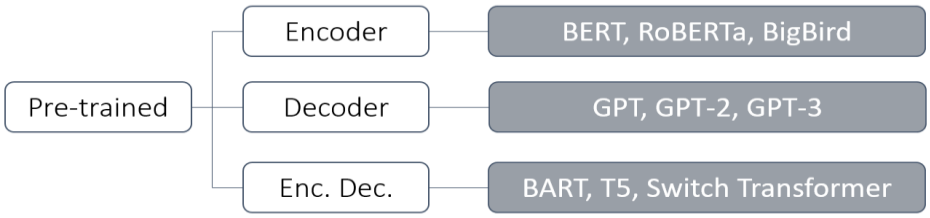
Τα πρώτα PTMs επικεντρώνονταν στην εκμάθηση καλών ενσωματώσεων λέξεων (word embeddings), όπως τα μοντέλα «Skip-Gram» και «GloVe» ενώ στη συνέχεια τα μοντέλα στόχευαν στην εκμάθηση ενσωματώσεων λέξεων με βάση τα συμφραζόμενα (contextual word embedding), όπως «CoVe», «ELMo», «OpenAI GTP» και «BERT».

Η κατηγοριοποίηση των PTMs γίνεται βάσει του τρόπου αναπαράστασης τους, της αρχιτεκτονικής τους, τον τρόπο με τον οποίο έχει γίνει η προ-εκπαίδευση τους αλλά και τις επεκτάσεις τους, για παράδειγμα υπάρχουν PTMs για πολλές γλώσσες (multilingual PTMs), PTMs πολλαπλών μοντέλων (multimodal PTMs) κ.α.. Οι Transformers με την αυτό-προσοχή ως βασικό συστατικό έχουν γίνει η κύρια επιλογή αρχιτεκτονικής για τα προ-εκπαιδευμένα μοντέλα γλώσσας στην επεξεργασία φυσικής γλώσσας (Qiu et al., 2020).



Εικόνα 1: Η αρχιτεκτονική του Transformer, αριστερά ο κωδικοποιητής και δεξιά ο αποκωδικοποιητής. Πηγή: Attention is all you need (Vaswani et al., 2017)

Ο μηχανισμός αυτό-προσοχής οδήγησε σε έναν νέο τρόπο σκέψης και ενέπνευσε μια πληθώρα νέων αρχιτεκτονικών. Όπως έχει ήδη αναφερθεί, η αρχιτεκτονική των Transformers βασίζεται στον κωδικοποιητή και αποκωδικοποιητή (encoder,decoder) και μπορεί να χρησιμοποιηθεί με 3 διαφορετικούς τρόπους: (Han et al., 2023)



Εικόνα 2: Ταξινομία των προ-εκπαιδευμένων μοντέλων Transformer

- **Encoder-Decoder:** Χρησιμοποιείται η πλήρης αρχιτεκτονική των Transformers. Ως συνήθως γίνεται χρήση στα μοντέλα «sequence-to-sequence». Όπως, στο μοντέλο BART και T5 (Text-To-Text Transfer Transformer).

- **Μόνο encoder:** Χρησιμοποιείται μόνο ο κωδικοποιητής και οι έξοδοι του κωδικοποιητή χρησιμοποιούνται ως αναπαράσταση για την ακολουθία εισόδου. Η επίλυση προβλημάτων ταξινόμησης βασίζεται σε αυτόν τον τρόπο. Προ-εκπαιδευμένα μοντέλα που χρησιμοποιούν μόνο τον κωδικοποιητή είναι το BERT και το RoBERTa.
- **Μόνο decoder:** Χρησιμοποιείται μόνο ο αποκωδικοποιητής, όπου αφαιρείται η μονάδα διασταυρούμενης προσοχής κωδικοποιητή-αποκωδικοποιητή. Βοηθάει στη μοντελοποίηση γλώσσας. Για παράδειγμα, τα «Generative Pre-trained Transformers (GPT)» μοντέλα GPT,GPT-2 και GPT-3.

Πίνακας 1:Πηγές εκπαίδευσης, μέγεθος συνόλου δεδομένων και ο αριθμός των παραμέτρων των μοντέλων για τα πιο διαδεδομένα Transformer PTMs.

Μοντέλο	*Πηγές προ-εκπαίδευσης*	*Μέγεθος σώματος προ-εκπαίδευσης*	*Αριθμός παραμέτρων*
BERT (BASE) (Devlin et al., 2019)	Wikipedia, BookCorpus	3.3B tokens	110M
BERT (LARGE) (Devlin et al., 2019)	Wikipedia, BookCorpus	3.3B tokens	340M
RoBERTa (Y. Liu et al., 2019)	Wikipedia, BookCorpus, web crawl*	161 GB data	340M
GPT (Radford, Narasimhan, et al., n.d.)	Web crawl	800M tokens	117M
GPT-2 (Radford, Wu, et al., n.d.)	Web crawl	40GB data	1.5B
GPT-3 (Brown et al., 2020)	Wikipedia, BookCorpus, web crawl	500B tokens	175B
BART (Lewis et al., 2020)	Wikipedia, BookCorpus	3.3B tokens	~370M

T5 (Raffel et al., 2020)	Web crawl	200B tokens	11B

***Web crawl : πρόγραμμα του διαδικτύου που συλλέγει και ανιχνεύει συνδέσμους και ιστοσελίδες**

Υπάρχει μια μεγάλη οικογένεια μοντέλων που προέρχεται από το BERT συμπεριλαμβανομένου του RoBERTa, XLNet, DeBERTa, ELECTRA κ.α.. Τα μοντέλα αυτά χρησιμοποιούνται συνήθως για εργασίες κατανόησης φυσικής γλώσσας και αλλάζοντας βασικά σημεία της αρχιτεκτονικής του BERT, βελτιώνεται η απόδοση σε διάφορες εργασίες της NLP. Για παράδειγμα, το T5 επιτυγχάνει την τελευταία λέξη της τεχνολογίας σε αρκετές εργασίες της NLP, όπως στα προβλήματα ταξινόμησης, το μοντέλο μαθαίνει να δημιουργεί σωστές ετικέτες (labels) με βάση την είσοδο (input). Από την άλλη, τα μοντέλα GPT έχουν επιδείξει καλύτερες αποδόσεις στην κατανόηση του περιβάλλοντος των δεδομένων κειμένου και μπορούν να εφαρμοστούν στη μηχανική μετάφραση έως τις εργασίες generation (Han et al., 2023).

Τα μοντέλα Transformer έχουν γίνει γρήγορα το νέο πρότυπο στις περισσότερες εργασίες της NLP. Οι αρχιτεκτονικές τους, με ή χωρίς προσαρμογές, χρησιμοποιούνται ως βάση για πιο πολύπλοκα συστήματα. Αυτή η βάση, είναι το προ-εκπαιδευμένο μοντέλο BERT και εν συνεχεία οι διάφορες βελτιώσεις του έχουν χρησιμοποιηθεί σε ένα ευρύ φάσμα εργασιών και είναι δύσκολο να ξεπεραστούν σε απόδοση.

2.4 Μοντέλο BERT

2.4.1 Εισαγωγή

Το μοντέλο BERT ή αλλιώς Bidirectional Encoder Representations από το Transformer είναι ένα μοντέλο γλώσσας που δημιουργήθηκε από μια ομάδα της Google το 2018 (Devlin et al., 2019) και η εκπαίδευση του έγινε σε ένα μεγάλο και δομημένο σύνολο κειμένων, όπως έχει ειπωθεί και στην ενότητα 1.4. Είναι ένα από τα πιο δημοφιλή και συχνά εφαρμοσμένα μοντέλα της NLP. Ο κωδικοποιητής πολλαπλών επιπέδων και διπλής κατεύθυνσης που χρησιμοποιείται στο μοντέλο BERT βασίζεται στην αρχιτεκτονική του Transformer.

Το μοντέλο BERT εκπαιδεύεται αμφίδρομα, επιτρέποντας να μαθαίνει το πλαίσιο τόσο πριν όσο και μετά από μια συγκεκριμένη λέξη και η διαδικασία αυτή πραγματοποιείται κατά τη διάρκεια της προ-εκπαίδευσης. Η αρχιτεκτονική του αμφίδρομου κωδικοποιητή είναι προ-εκπαιδευμένη με δύο κύριες εργασίες: το γλωσσικό μοντέλο απόκρυψης (Masked Language Model- MLM) και την πρόβλεψη της επόμενης

πρότασης (Next Sentence Prediction-NSP). Το MLM αποτελείται από την τυχαία απόκρυψη ενός ποσοστού των δεδομένων εισόδου με στόχο να προβλέψει τις καλυμμένες λέξεις. Κατά αυτόν τον τρόπο αναγκάζεται το μοντέλο να επικεντρωθεί περισσότερο στις λέξεις που περιβάλλουν την κρυμμένη λέξη έτσι ώστε να συλλάβει σωστά το νόημα. Από την άλλη, το NSP είναι μια εργασία δυαδικής ταξινόμησης και εστιάζει περισσότερο σε εργασίες σε επίπεδο πρότασης. Το μοντέλο σε αυτή τη διαδικασία, προσπαθεί να συμπεράνει εάν δυο προτάσεις είναι διαδοχικές. Στη συνέχεια, το προ-εκπαιδευμένο μοντέλο BERT μπορεί να ακολουθήσει τη διαδικασία του fine-tuning για να τελειοποιήσει την επίδοση του σε διάφορες εργασίες. Λεπτομερέστερη αναφορά στην αρχιτεκτονική και στον τρόπο προ-εκπαίδευσης του μοντέλου γίνεται στο Κεφάλαιο 4 (Devlin et al., 2019).

Προσφάτως, το μοντέλο BERT έχει επιτύχει αποτελέσματα τελευταίας τεχνολογίας σε αρκετές εργασίες της NLP με συνέπεια οι προηγούμενες τεχνικές να μην εξακολουθούν να είναι χρήσιμες συγκριτικά με τις επιδόσεις του BERT και των μοντέλων που αναπτύχθηκαν βάσει αυτού. Η επιτυχία των μοντέλων ευθύνεται κυρίως στην προ-εκπαίδευσή τους. Το BERT δοκιμάστηκε σε πολλά σύνολα δεδομένων για να συγκριθεί η απόδοση του με άλλα δημοσιευμένα μοντέλα και παρουσίασε βελτιστοποίηση αποτελεσμάτων σε 11 εργασίες της NLP, συμπεριλαμβανομένου του GLUE, SQuAD και SWAG. Αναλυτικότερα, στη δοκιμή με το General Language Understanding ή αλλιώς GLUE, το οποίο είναι μια συλλογή που δοκιμάζουν την κατανόηση της φυσικής γλώσσας, τόσο το $\text{BERT}_{\text{BASE}}$ όσο και το $\text{BERT}_{\text{LARGE}}$ έδειξαν υπεροχή σε όλες τις εργασίες επιτυγχάνοντας 4,5% και 7,0% αντίστοιχα μέση βελτίωση σε σύγκριση με τα πιο γνωστά μοντέλα. Στο μοντέλο SQuAD (The Stanford Question Answering Dataset) τίθεται μια ερώτηση και δίνεται ένα απόσπασμα από τη Wikipedia που περιέχει την απάντηση και το μοντέλο πρέπει να επιλέξει από το κείμενο το απόσπασμα που απαντά στην ερώτηση. Το BERT παρουσίασε καλύτερα αποτελέσματα σύμφωνα με τη μέτρηση F1 σε σχέση με το SQuAD v1.1 και το SQuAD v2.0, που αποτελεί μια επέκταση του προηγούμενου και επιτρέπει να μην υπάρχει σύντομη απάντηση στην δοθείσα παράγραφο, καθιστώντας το πρόβλημα πιο ρεαλιστικό. Τέλος, συγκριτικά με το Situation With Adversarial Generations (SWAG), το οποίο είναι ένα σύνολο δεδομένων που περιέχει 113 χιλιάδες ζεύγη προτάσεων, και πιο συγκεκριμένα είναι ένα σύνολο ερωτήσεων με τέσσερις πιθανές απαντήσεις για να επιλεγεί η πιο εύστοχη συνέχεια της πρότασης, το BERT ξεπέρασε το βασικό σύστημα κατά 27,1%

παρουσιάζοντας ακρίβεια 86,3% που είναι υψηλότερη και από αυτή ενός ανθρώπου με συναφή ειδικότητα (Devlin et al., 2019).

2.4.2 BERT σε ιατρικά δεδομένα

Τα προ-εκπαιδευμένα γλωσσικά μοντέλα, όπως έχει ήδη αναφερθεί, είναι αποτελεσματικά σε εργασίες στον γενικό τομέα, γεγονός που πυροδότησε τους ερευνητές για το εάν αυτά τα μοντέλα ή παραλλαγές τους είναι εξίσου αποτελεσματικά σε εργασίες στον ιατρικό τομέα. Επειδή υπάρχουν αρκετές πληροφορίες για τους ασθενείς, η χρήση ιατρικών δεδομένων κειμένου παρουσιάζει δυσκολία. Απόρροια αυτού, εργασίες όπως η ανάκτηση πληροφοριών που εκτελούνται από επαγγελματίες του τομέα της υγείας να είναι επαναλαμβανόμενες, χρονοβόρες και κουραστικές. Η χρήση αποτελεσματικών μεθόδων μέσω της αξιοποίησης των προ-εκπαιδευμένων μοντέλων αποτελεί μια λύση για την υποστήριξη της ανάκτησης πληροφοριών στον ιατρικό τομέα.

Τα PTMs και συγκεκριμένα το μοντέλο BERT εκπαιδεύεται και μαθαίνει αμφίδρομα χρησιμοποιώντας το MLM και το NSP στο σύνολο των δεδομένων εκπαίδευσης του, Wikipedia και BookCorpus. Ωστόσο, οι λέξεις συχνά έχουν διαφορετική σημασία ανάλογα με το πλαίσιο στο οποίο χρησιμοποιούνται. Αρκετές έρευνες έχουν δείξει ότι η προ-εκπαίδευση νέων μοντέλων γλώσσας που χρησιμοποιούν δεδομένα για συγκεκριμένο τομέα ή η βελτίωση και εξέλιξη των ήδη υπάρχον μπορεί να βελτιώσει την απόδοση σε εργασίες που αφορούν έναν συγκεκριμένο τομέα, όπως η ιατρική (Lamproudis et al., n.d.; Lee et al., 2020).

Το μοντέλο BERT μπορεί να εκπαιδευτεί σε σύνολο ιατρικών δεδομένων και να είναι αποτελεσματικότερο σε εργασίες ιατρικού κειμένου. Αρχικά, η εκπαίδευση του BioBERT γίνεται σε σώματα ιατρικού τομέα, όπως περιλήψεις από το PubMed και σε ολόκληρα άρθρα από το PMC. Στη συνέχεια, το μοντέλο αυτό περνάει στη διαδικασία του fine-tuning που βασίζεται σε τρεις εργασίες εξόρυξης ιατρικών κειμένων, Named Entity Recognition, Relation Extraction και Question Answering. Η έκδοση αυτή έχει ήδη αποδειχθεί αρκετά αποτελεσματική σε πολλές εργασίες εξόρυξης κειμένου όπως NER για κλινικές σημειώσεις, RE του ανθρώπινου φαινοτύπου κα.(Lee et al., 2020). Εκτός από το BioBERT έχουν αναπτυχθεί και άλλα μοντέλα, λόγου χάρη το BioClinicalBERT, αποτελεί μια παραλλαγή που ειδικεύεται στις κλινικές σημειώσεις, και το Bio-discharge-summary, που βασίστηκε στο BioBERT αλλά εκπαιδεύτηκε επιπλέον στις κλινικές σημειώσεις από το σύνολο δεδομένων MIMIC.

2.4.3 BERT και Information Extraction

Το μοντέλο BERT έχει αξιοποιηθεί στις περισσότερες από τις εργασίες που αναφέρθηκαν στο 2.2.3. Στη παρούσα έρευνα θα ασχοληθούμε κυρίως με τις εφαρμογές που αφορούν το Information Extraction σε ιατρικά δεδομένα. Η εξαγωγή πληροφοριών στην ουσία αποτελεί την αυτόματη εύρεση σημαντικών πληροφοριών σε κείμενο μη δομημένης γλώσσας, με εργασίες όπως named entity recognition, event και relation extraction κ.α..

Δεδομένου ότι οι περισσότερες πληροφορίες ιατρικής φύσης περιέχονται στα ιατρικά αρχεία ως ελεύθερο κείμενο, απαιτείται η χρήση προηγμένης τεχνολογίας για την αυτόματη εξαγωγή πληροφοριών. Ωστόσο, για τη δημιουργία τέτοιων μοντέλων απαιτούνται λεπτομερέστερα annotations, τα οποία περιγράφουν τις πληροφορίες στο σώμα εκπαίδευσης, αλλά αυτή η διαδικασία θέλει χρόνο και κόπο. Η NLP με τα PTMs, μπορεί να αναλύσει και να οργανώσει αυτόματα πληροφορίες από ελεύθερο κείμενο και να δώσει λύση στην αυτοματοποίηση διαφόρων διαδικασιών. Τέτοιες διαδικασίες είναι η αυτόματη εξαγωγή συμπτωμάτων, η αυτόματη ανίχνευση ασθενειών, η αυτόματη χαρτογράφηση των συμπτωμάτων και άλλες (Faris et al., 2022).

Οι ερευνητές ενδιαφέρονται να αυτοματοποιήσουν τη διαδικασία αναγνώρισης ιατρικών συμπτωμάτων αρκετά χρόνια. Γενικά στην NLP σε ιατρικό κείμενο και στο επίπεδο εξαγωγής χαρακτηριστικών, το μοντέλο BERT, ως το μοντέλο τελευταίας τεχνολογίας, χρησιμοποιήθηκε σε αρκετές μελέτες. Οι (Mu et al., 2021) εκπαίδευσαν ένα μοντέλο BERT για να χαρτογραφήσει και να εξάγει διαγνωστικά σχετικές πληροφορίες κειμένου από τις παθολογικές συνόψεις. Στην ουσία οι συνόψεις παθολογίας αποτελούνται από ημι-δομημένο ή αδόμητο κείμενο που συνοψίζει τις οπτικές πληροφορίες που προέρχονται από την παρατήρηση του ανθρώπινου ιστού. Αντίστοιχα, έχουν δημιουργηθεί προηγμένα μοντέλα βαθιάς μάθησης για την εξαγωγή κλινικών όρων από ακτινολογικές αναφορές, με ικανοποιητικά αποτελέσματα έχοντας F1-scores 95,36% και 94,62% σε δύο διαφορετικά σύνολα δεδομένων (Sugimoto et al., 2021).

Μια ακόμη έρευνα, ανέπτυξε ένα μοντέλο βασισμένο στο BioBERT για την πρόβλεψη της αλληλεπίδρασης φαρμάκου-φαρμάκου (Drug-Drug interaction) από σώμα κειμένου χρησιμοποιώντας πληροφορίες χημικής δομής. Η στρατηγική αυτή βελτίωσε άλλες ισχυρές αρχιτεκτονικές βασικών γραμμών κατά 3,4% σε βαθμολογία του F1 (Mondal, 2020). Εν συνεχεία, η κατανόηση του ιστορικού φαρμάκων ενός ασθενούς προκειμένου

οι γιατροί να του παρέχουν καλύτερες προτάσεις είναι σημαντική. Οι (Mahajan et al., 2021) παρουσίασαν μια αυτοματοποιημένη προσέγγιση για τον υπολογισμό της ημερήσιας δοσολογίας σε όλα τα φάρμακα, με τη χρήση του ClinicalBERT.

Υπάρχουν και άλλες αξιοσημείωτες εργασίες στη βιβλιογραφία που επικεντρώθηκαν στο ζήτημα της αναγνώρισης συμπτωμάτων σε διαφορετικά γλωσσικά πλαίσια. Στα αραβικά, οι (Faris et al., 2022) ανέπτυξαν ένα μοντέλο βαθιάς μάθησης για την αυτόματη αναγνώριση συμπτωμάτων. Για την εξαγωγή κλινικών οντοτήτων από τα κινέζικα δεδομένα που προέρχονται από τις αξονικές τομογραφίες, δημιουργήθηκε μια τεχνική βαθιάς μάθησης με προ-εκπαίδευση που ονομάζεται BERT-BTN. Η προσέγγιση αυτή έδειξε ότι μπορεί να διακρίνει αποτελεσματικά διαφορετικές κλινικές οντότητες σχετικά με τον προ-συμπτωματικό έλεγχο και τη σταδιοποίηση του καρκίνου του πνεύμονα (H. Zhang et al., 2021). Τέλος, σε γερμανικό γλωσσικό πλαίσιο, οι (Schäfer et al., 2020) πρότειναν ένα μοντέλο που βασίζεται στο BERT για την αναγνώριση και την εξαγωγή συμπτωμάτων χρησιμοποιώντας γερμανικούς μονολόγους ασθενών. Το μοντέλο αυτό, επίδειξε υποσχόμενη απόδοση ξεπερνώντας σημαντικά τις απλούστερες γραμμές βάσης.

2.5 Ερευνητικό κενό

Παρά τη σημαντική πρόσφατη πρόοδο, οι εφαρμογές της επεξεργασίας φυσικής γλώσσας επεκτείνονται καθημερινά και αυτή η ανάπτυξη δημιουργεί νέα εμπόδια. Η σημασία των λέξεων ή των προτάσεων μπορεί να είναι διαφορετική στον κλάδο της εκπαίδευσης, της υγείας, της νομοθεσίας οπότε είναι μια πρόκληση που τα μοντέλα της NLP πρέπει να αντιμετωπίσουν. Αν και οι τεχνικές βαθιάς μάθησης και τα PTMs είχαν επιτυχία στον γενικό τομέα της NLP, η εφαρμογή τους στη βιοϊατρική βιομηχανία εξακολουθεί να είναι προβληματική λόγω της σπανιότητας, καθώς παρατηρείται έλλειψη ανταλλαγής δεδομένων μεταξύ των οργανισμών υγειονομικής περίθαλψης, και της πολυπλοκότητας των δεδομένων κειμένου. Αρκετά PTMs μπορούν να βελτιωθούν περαιτέρω με περισσότερα βήματα εκπαίδευσης και μεγαλύτερα σύνολα δεδομένων. Επιπλέον, αντί η εκπαίδευση τους να πραγματοποιείται από την αρχή για κάθε εργασία, μπορεί να γίνει με βάση τα υπάρχοντα PTMs γενικής χρήσης χρησιμοποιώντας τεχνικές όπως η συμπίεση μοντέλων (Qiu et al., 2020).

Τέλος, ως μελλοντική εργασία μπορεί να θεωρηθεί και η ανάπτυξη προ-εκπαιδευμένων μοντέλων BERT αλλά και γενικότερα της οικογενείας που προέρχεται από αυτό το μοντέλο, για εφαρμογή στην ιατρική με χρήση κειμένου από διάφορες γλώσσες. Οι

περισσότερες έρευνες που έχουν πραγματοποιηθεί είναι στα αγγλικά και στα κινέζικα. Η μεγαλύτερη πρόκληση που έχουν να αντιμετωπίσουν οι ερευνητές για την ανάπτυξη μοντέλων σε διαφορετική γλώσσα είναι το μέγεθος του συνόλου δεδομένων. Τέτοια ανάπτυξη μοντέλου μπορεί να γίνει και στα ελληνικά, καθώς η αντίστοιχη έρευνα στη βιβλιογραφία είναι ελάχιστη με ένα παράδειγμα των (Papaioannou et al., 2022). Η ανάπτυξη αυτών των μοντέλων και οι συνεχείς βελτιώσεις τους, μπορούν να οδηγήσουν στην αυτοματοποίηση διάφορων εργασιών στον ιατρικό κλάδο.

2.6 Ερευνητική υπόθεση

Αυτή η έρευνα προτείνεται για τον αυτόματο προσδιορισμό κλινικών εννοιών από τις σημειώσεις που προέρχονται από τους εκπαιδευόμενους γιατρούς κατά την αλληλεπίδραση τους με τους ασθενείς. Αξίζει να σημειωθεί ότι οι σημειώσεις είναι γραμμένες στην αγγλική γλώσσα. Το Εθνικό Συμβούλιο Ιατρικών Εξεταστών έχει προτείνει την αξιοποίηση της τεχνολογίας της NLP έτσι ώστε να μετριαστούν οι προκλήσεις που σχετίζονται με τον ανθρώπινο παράγοντα. Η αυτοματοποίηση αυτή, θα επιταχύνει την εκπαίδευση των ασκούμενων ιατρών καθώς απαιτείται ένα κλάσμα του χρόνου που χρειάζεται ένας άνθρωπος για να βαθμολογήσει και κατ’ επέκταση θα συμβάλει στην αύξηση της συνέπειας, της αντικειμενικότητας και της αποτελεσματικότητας.

Για την επίτευξη αυτού, θα γίνει χρήση ενός προ-εκπαιδευμένου μοντέλου NLP που παρουσιάζει επιδόσεις κοντά στην τελευταία λέξη της τεχνολογίας. Το μοντέλο BERT, όπως έχει ήδη αναφερθεί, έχει παρουσιάσει βελτιστοποίηση της απόδοσης σε διάφορες εργασίες της NLP σε σύγκριση με προηγούμενα μοντέλα (Devlin et al., 2019) και θεωρείται η καταλληλότερη επιλογή για την εφαρμογή σε ιατρικά δεδομένα (Lee et al., 2020). Επομένως, η ερευνητική υπόθεση με την οποία θα ασχοληθούμε είναι «Το προ-εκπαιδευμένο μοντέλο BERT μπορεί να χρησιμοποιηθεί για την αυτοματοποίηση της διαδικασίας μάθησης και αξιολόγησης των εκπαιδευόμενων ιατρών της εξέτασης USMLE Step 2». Στην ουσία θέλουμε να δούμε εάν οι πρόσφατες εξελίξεις στην επιστήμη των υπολογιστών και στην NLP, μπορούν να ωθήσουν την ιατρική κοινότητα να εξετάσει τη χρήση της τεχνολογίας και ειδικότερα του προ-εκπαιδευμένου μοντέλου BERT, για την αξιολόγηση της απόδοσης των εκπαιδευόμενων ιατρών.

Κεφάλαιο 3- Μεθοδολογία

3.1 Ερευνητική στρατηγική

Για την υλοποίηση της παρούσας εργασίας επιλέχθηκε η εφαρμογή του μοντέλου διαδικασίας CRISP-DM (Cross-Industry Standard Process for Data Mining). Αρχικά, πραγματοποιήθηκε έρευνα σχετικά με το είδος της μεθοδολογίας που ταιριάζει και μπορεί να εφαρμοστεί στην ερευνητική υπόθεση μας. Ανάμεσα από τις μεθοδολογίες που υπάρχουν και παρουσιάζονται στο (Martinez et al., 2021), όπως CRIP-DM, Microsoft TDSP (Team Data Science Process), συστηματική έρευνα σε μεγάλα δεδομένα (Big Data) κ.α., επιλέχθηκε η πρώτη έτσι ώστε να ελέγξουμε εάν το προ-εκπαιδευμένο μοντέλο BERT μπορεί να χρησιμοποιηθεί στην ιατρική κοινότητα και να βοηθήσει στην αυτοματοποίηση της διαδικασίας εκπαίδευσης των εκπαιδευόμενων ιατρών.

Η μέθοδος CRISP-DM είναι ένα ανοιχτό μοντέλο διαδικασίας που αναπτύχθηκε το 1996 και αποτελεί μια καλά δομημένη και καθορισμένη διαδικασία. Ένα πρόβλημα της επιστήμης των δεδομένων αναλύεται σε έξι φάσεις σύμφωνα με το CRISP-DM: την κατανόηση της επιχειρηματικής πρόθεσης (Business Understanding), την κατανόηση των δεδομένων (Data Understanding), την προετοιμασία των δεδομένων (Data preparation), την μοντελοποίηση (Modeling) και τέλος την αξιολόγηση και ανάπτυξη του μοντέλου (Evaluation και Deployment αντιστοίχως). Αποτελεί την πιο διαδεδομένη διαδικασία, τόσο στην ακαδημαϊκή έρευνα όσο και στον επιχειρησιακό κλάδο. Ακόμη, άλλες τεχνικές χρησιμοποιούν συχνά ως αναφορά τα 6 στάδια του CRISP-DM και παράλληλα το αναπαράγουν με μικρές αλλαγές.

Ο παρακάτω πίνακας περιγράφει συνοπτικά τη κύρια ιδέα των φάσεων που χρησιμοποιεί η μεθοδολογία του CRISP-DM (Schröer et al., 2021).

Πίνακας 2: Σύντομα περιγραφή των 6 φάσεων της μεθοδολογίας CRISP-DM

Φάση	*Σύντομη περιγραφή*
Business Understanding	Περιγράφονται οι στόχοι και οι απαιτήσεις της έρευνας. Πρώτα εξηγείται ο τύπος του προβλήματος (π.χ. εξαγωγή πληροφοριών) και τα κριτήρια μέτρησης της ακρίβειας.
Data Understanding	Πραγματοποιείται συλλογή δεδομένων από διάφορες πηγές δεδομένων και περιγραφή και αξιολόγηση της ποιότητας τους. Χρησιμοποιείται στατιστική ανάλυση.

Data preparation	Καθαρισμός των δεδομένων και προετοιμασία τους για το επιλεγμένο μοντέλο της πρώτης φάσης.
Modeling	Αναπτύσσεται το δοκιμαστικό μοντέλο αλλά και το τελικό μοντέλο, ανάλογα με τα δεδομένα και την επιχειρηματική πρόταση. Αιτιολόγηση της επιλογής.
Evaluation	Τα αποτελέσματα συγκρίνονται με τους καθορισμένους επιχειρηματικούς στόχους. Είναι απαραίτητο να καθοριστούν πρόσθετες ενέργειες και να αξιολογηθούν τα ευρήματα. Έλεγχος εάν η διαδικασία πρέπει να αναθεωρηθεί.
Deployment	Αποτελεί την τελική αναφορά και περιλαμβάνει τον σχεδιασμό, την παρακολούθηση και τη συντήρηση.

Το CRISP-DM επιλέγεται από τους ερευνητές, και αξίζει να σημειωθεί ότι οι περισσότερες περιπτώσεις χρήσης εντοπίζονται στον τομέα της υγείας. Αυτό συμβαίνει καθώς αποτελεί το πρότυπο για την εφαρμογή ενός μοντέλου διαδικασίας σε έργα εξόρυξης και ως επακόλουθο αυτού θεωρείται αξιόπιστο και είναι ευρέως χρησιμοποιούμενο. Ωστόσο, ένα από τα μειονεκτήματα της επιλογής αυτής της μεθόδου αποτελεί το γεγονός της έλλειψης της έκτης και τελευταίας φάσης, της ανάπτυξης, από τις περισσότερες έρευνες που έχουν πραγματοποιηθεί. Οι (Schröer et al., 2021) έχουν προσπαθήσει να εντοπίσουν την αιτία της παράλειψης του τελευταίου σταδίου και αναφέρουν ότι μπορεί να οφείλεται στις ελλιπείς οδηγίες σχετικά με τον τρόπο διεξαγωγής της ανάπτυξης αλλά και ότι οι περισσότερες εργασίες στοχεύουν μόνο στην κατασκευή και στην αξιολόγηση μοντέλων και όχι στην ανάπτυξη τους. Ένα άλλο μειονέκτημα του CRISP-DM είναι πως δεν περιγράφει πώς πρέπει να οργανωθούν οι ομάδες έρευνας για να πραγματοποιήσουν τις προβλεπόμενες διαδικασίες, συγκριτικά με νεότερες μεθοδολογίες. Απαιτείται λοιπόν, καλύτερη ενοποίηση με της διαδικασίες διαχείρισης και καλύτερη καθοδήγηση των μεθόδων για συγκεκριμένες εργασίες εντός των σταδίων και όχι απλές λίστες ελέγχου. Τέλος, αν και αποτελεί ένα ολοκληρωμένο πλαίσιο το γεγονός ότι δεν έχει ενημερωθεί από το 1996, έχει ως συνέπεια να μην αντικατοπτρίζει τις πιο πρόσφατες εξελίξεις στην τεχνολογία της επιστήμης των δεδομένων (Martinez et al., 2021).

3.2 Ερευνητική διαδικασία

Η ερευνητική διαδικασία την οποία θα ακολουθήσουμε είναι η χρήση και ανάλυση δεδομένων από το διαδίκτυο και για την ακρίβεια θα χρησιμοποιήσουμε σύνολα δεδομένων από τη σελίδα Kaggle. Το Kaggle, εν συντομία, είναι μια διαδικτυακή πλατφόρμα που χρησιμοποιείται κυρίως από επιστήμονες για ανάλυση δεδομένων και μηχανική μάθηση. Επιτρέπει τους χρήστες να συνεργάζονται με άλλους χρήστες, να βρίσκουν και να δημοσιεύουν σύνολα δεδομένων, να χρησιμοποιούν σημειωματάρια (notebooks) και να ανταγωνίζονται με άλλους επιστήμονες κατά τη διεξαγωγή διαγωνισμών.

Η πλατφόρμα του Kaggle είναι η πιο δημοφιλής ανάμεσα στις πλατφόρμες επιστήμης δεδομένων που διαθέτουν διαγωνισμούς και προκλήσεις. Οι 10 εκατομμύρια μοναδικοί χρήστες της, κυμαίνονται από απολύτως αρχάριοι έως ειδικοί σε διάφορους κλάδους. Επιπλέον, είναι διαδεδομένη καθώς τα μέλη μπορούν να μοιράζονται τα σύνολα δεδομένων με την κοινότητα, με συνέπεια να διατίθενται δημόσια και η χρήση τους να είναι δωρεάν. Αυτή η δυνατότητα διευκολύνει την πρόσβαση σε διάφορα σύνολα δεδομένων. Πάνω από 40,000 σύνολα δεδομένων είναι δημόσια διαθέσιμα στην πλατφόρμα του Kaggle (Zimmermann, 2021).

Τέλος, το περιβάλλον της Kaggle είναι οργανωμένο, με επαρκή αιτιολόγηση των επιμέρους στοιχείων του εκάστοτε συνόλου δεδομένων και παράλληλα προσφέρει και τη δυνατότητα επεξεργασίας και ανάλυσης τους. Τα οφέλη της αξιοποίησης της συγκεκριμένης πλατφόρμας είναι η εύκολη πρόσβαση στα διαθέσιμα δεδομένα, η πρόσβαση σε έτοιμους κώδικες, διαγωνισμούς και μαθήματα αλλά και η δυνατότητα να γίνεις μέλος μιας κοινότητας με σκοπό τη μάθηση, ανάπτυξη γνώσεων και δικτύωση με άλλους επιστήμονες.

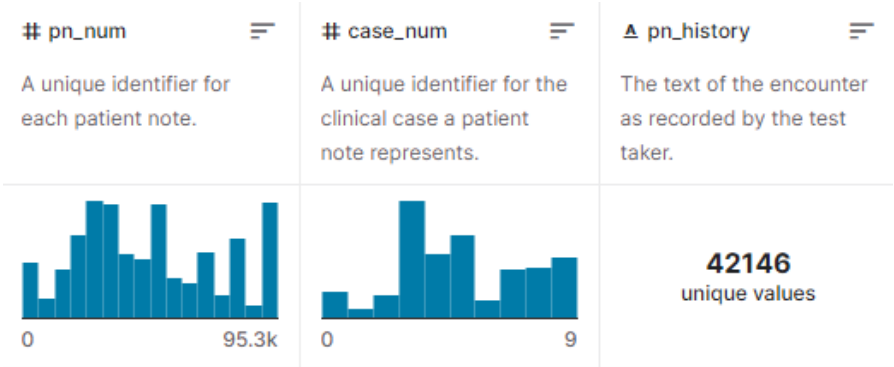
3.3 Σύνολο δεδομένων

Τα δεδομένα που θα χρησιμοποιήσουμε στην παρούσα διπλωματική, έγιναν διαθέσιμα από το Εθνικό Συμβούλιο Ιατρικών Εξεταστών (National Board of Medical Examiners-NBME) με τη μορφή διαγωνισμού, μέσα στη πλατφόρμα του Kaggle, τον Φεβρουάριο του 2022. Το NBME προσφέρει αξιολογήσεις και εκπαιδευτικές υπηρεσίες στους επαγγελματίες υγείας, στους φοιτητές αλλά και σε άλλους φορείς για την βελτίωση των γνώσεων τους στις εξελισσόμενες ανάγκες τις ιατρικής εκπαίδευσης και της υγειονομικής περίθαλψης.

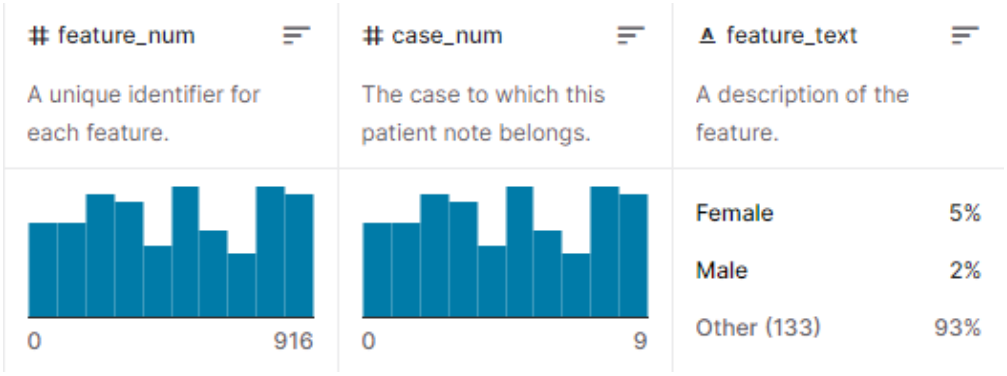
Τα δεδομένα κειμένου που έγιναν διαθέσιμα προέρχονται από την εξέταση «USMLE Step 2 Clinical Skills», η οποία αποτελεί προϋπόθεση για την απόκτηση ιατρικής άδειας. Η εξέταση αυτή, αξιολογεί την ικανότητα του ασκούμενου να αποκτά πληροφορίες, να διεξάγει φυσικές εξετάσεις στον ασθενή και να αναλύει τα δεδομένα. Κάθε εξεταζόμενος συναντά έναν τυποποιημένο ασθενή, που απεικονίζει μια ιατρική περίπτωση (clinical case). Καταγράφει σε ένα σημείωμα ασθενούς (patient note) τις λεπτομέρειες της αλληλεπίδρασης με τον ασθενή και τέλος, ένας εξειδικευμένος γιατρός βαθμολογεί κάθε σημείωση ασθενή αναζητώντας την ύπαρξη συγκεκριμένων βασικών ιδεών ή χαρακτηριστικών (features) που σχετίζονται με την περίπτωση. Για παράδειγμα, για μια ιατρική περίπτωση σχετικά με έναν ασθενή με συνεχείς πονοκεφάλους, μπορεί να είναι σημαντικό ο εξεταζόμενος ιατρός να κάνει ερωτήσεις που οδηγούν στην πληροφορία ότι ο ασθενής έχει φωτοφοβία. Σε μια περίπτωση όπως αυτή, η φωτοφοβία θα αναφέρεται ως ένα από τα σημαντικά χαρακτηριστικά και οι σημειώσεις που δεν αναφέρουν αυτό το συγκεκριμένο σύμπτωμα ή κάποια έκφρασή του, όπως ευαίσθητο στο φως, θα λαμβάνουν χαμηλότερη βαθμολογία (Yaneva et al., 2022).

Το σύνολο δεδομένων μας αποτελείται από 4 αρχεία, το patient_notes.csv, features.csv, train.csv τα οποία συνθέτουν τα δεδομένα εκπαίδευσης και το test.csv , που είναι ένα παράδειγμα δεδομένων και στο τέλος θα αντικατασταθεί από τα πραγματικά δεδομένα δοκιμής.

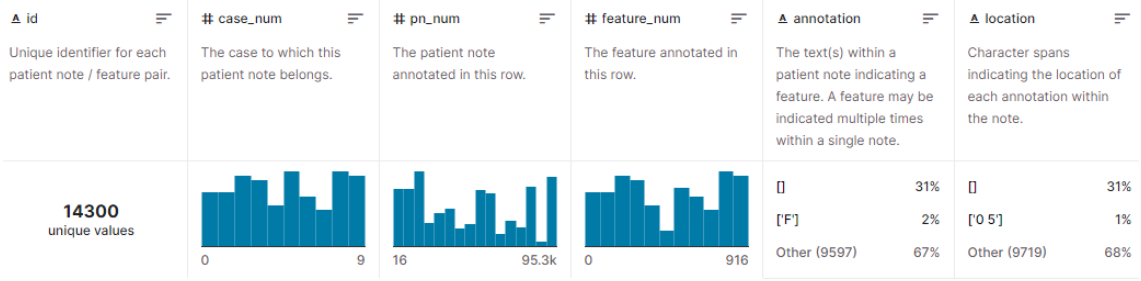
Το patient_notes.csv αποτελείται από περίπου 40,000 τμήματα ιστορικού σημειώματος των ασθενών (Εικόνα 3), το features.csv είναι τα χαρακτηριστικά ή οι βασικές έννοιες για κάθε ιατρική περίπτωση (Εικόνα 4). Τέλος, το train.csv περιέχει τα annotations για τις 1000 σημειώσεις ασθενών (Εικόνα 5).



Εικόνα 3: Το patient_notes.csv όπως παρουσιάζεται στις οδηγίες του διαγωνισμού. Πηγή: https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/data



Εικόνα 4: Το features.csv όπως παρουσιάζεται στις οδηγίες του διαγωνισμού. Πηγή: https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/data



Εικόνα 5: Το train.csv όπως παρουσιάζεται στις οδηγίες του διαγωνισμού. Πηγή: https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/data

Εκτενέστερη αναφορά στο σύνολο δεδομένων μας θα πραγματοποιηθεί στην ενότητα 4.2.

3.4 Ανάλυση δεδομένων

Η αφορμή αυτής της έρευνας είναι να αυτοματοποιήσει την διαδικασία εξαγωγής ιατρικών εννοιών μέσα από τις σημειώσεις που γράφονται από εκπαιδευόμενους ιατρούς, έτσι ώστε η αξιολόγηση τους να γίνεται γρηγορότερα χωρίς την παρουσία έμπειρου ιατρικού προσωπικού. Για την επίτευξη αυτού, καθώς αποτελεί ένα πρόβλημα της επεξεργασίας φυσικής γλώσσας, θα γίνει χρήση των προ-εκπαιδευμένων μοντέλων Transformers και ειδικότερα του μοντέλου BERT. Η επιλογή του μοντέλου BERT έγινε καθώς αποτελεί μοντέλο ανοιχτού κώδικα και επειδή έχει εκπαιδευτεί σε ένα τεράστιο σύνολο δεδομένων. Απόρροια αυτών, η μείωση της χρονοβόρας διαδικασίας εκπαίδευσης ενός μοντέλου από την αρχή και η βελτιστοποίηση της απόδοσης του.

Η μέτρηση της απόδοσης ενός μοντέλου μπορεί να αξιολογηθεί με ένα ευρύ πλήθος μετρικών και η επιλογή της καταλληλότερης μετρικής εξαρτάται από το πρόβλημα που έχουμε να αντιμετωπίσουμε. Στην παρούσα έρευνα, το αποτέλεσμα θα αξιολογηθεί με

την μετρική micro-averaged F1 score. Ο κώδικας είναι στην γλώσσα προγραμματισμού Python και ακολουθεί τη δομή του PyTorch. Τέτοια, αυτοματοποιημένα συστήματα πρέπει να είναι εξαιρετικά ακριβή προκειμένου να είναι λειτουργικά χρησιμοποιήσιμα και ισοδύναμα με τις επιδόσεις του ανθρώπου. Βάσει των (Yaneva et al., 2022), οι βαθμολογίες F1 που πρέπει να ξεπεραστούν είναι 0,84, με βάση την επικάλυψη θέσης χαρακτήρων, και για τη δυαδική βαθμολογία F1 είναι 0,97, δηλαδή για το εάν ένα δεδομένο χαρακτηριστικό εκφράστηκε σε ένα σημείωμα ασθενούς (1 εάν βρέθηκε, 0 διαφορετικά).

Σύμφωνα με τις οδηγίες του διαγωνισμού έχουμε ότι κάθε χαρακτήρας βαθμολογείται ως: True Positive (TP), εάν είναι μέσα σε μια βασική αλήθεια όσο και σε μια πρόβλεψη, False Negative (FN), εάν είναι μέσα σε μια βασική αλήθεια αλλά όχι σε μια πρόβλεψη και, False Positive (FP), αν είναι μέσα σε μια πρόβλεψη αλλά όχι μια βασική αλήθεια. Η μετρική F1 score υπολογίζεται σε συνδυασμό από τα TP, FN και FP που συγκεντρώνονται σε όλες τις περιπτώσεις.

Ο τύπος της μετρικής F1 score είναι:

$$F_1 = 2 \times \frac{Precison \times Recall}{Precision + Recall}$$

Με τον τύπο του Precision και Recall να είναι

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

και

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Η χρήση της μετρικής F1 score είναι απαραίτητη όταν αναζητάμε ισορροπία μεταξύ του Precision και του Recall και όταν παρατηρείται διαφορά στην κατανομή των κλάσεων. Ένας άλλος διαχωρισμός που υπάρχει για την μετρική αυτή είναι το averaging, το οποίο μπορεί να είναι macro, weighted και micro και η επιλογή της καταλληλότερης μεθόδου κρίνεται με βάση το σύνολο δεδομένων. Στην προκείμενη περίπτωση, διαθέτουμε ένα ισορροπημένο σύνολο δεδομένων και αναζητάμε μια μετρική για τη συνολική απόδοση που είναι ανεξάρτητη από τις κατηγορίες.

Ο μικρό-μέσος όρος δίνει σε κάθε ζεύγος κλάσης του δείγματος ίση συνεισφορά στη συνολική μέτρηση. Αναλυτικότερα, αθροίζει τα μερίσματα και τους διαιρέτες που

συνθέτουν τις μετρήσεις ανά κατηγορία για τον υπολογισμό ενός συνολικού πηλίκου. Προτιμάται όταν υπάρχουν πολλαπλές ετικέτες (multilabel). Γι' αυτό τον λόγο, η αξιολόγηση της προβλεπτικής ακρίβειας πραγματοποιείται με το F1 micro score (*3.3. Metrics and Scoring*, n.d.).

Ολοκληρώνοντας, το τελικό αποτέλεσμα του μοντέλου θα αποθηκευτεί σε ένα καινούργιο σύνολο δεδομένων, το οποίο θα έχει 2 στήλες, η πρώτη θα δηλώνει το id του χαρακτηριστικού που έχει γίνει annotation και η δεύτερη, τη τοποθεσία/τοποθεσίες αυτού του χαρακτηριστικού μέσα στο patient note, οι οποίες θα χωρίζονται με ερωτηματικό (;) (Εικόνα 6).

id Unique identifier for this instance, a feature within a patient note.	location Character spans indicating the location(s) of the feature within the note.
00016_000	0 100
00016_001	
00016_002	200 250;300 400
00016_003	
00016_004	75 110

Εικόνα 6: Δείγμα της μορφής του τελικού αποτελέσματος του μοντέλου

Για παράδειγμα, για το id 00016_000, πρέπει το μοντέλο να δώσει πρόβλεψη για το χαρακτηριστικό (feature) 000, στη σημείωση ασθενούς (patient note) 00016 και αυτή να είναι στη θέση 0 έως 100.

Κεφάλαιο 4- Ανάπτυξη θέματος

4.1 Εισαγωγή

Το Μάρτιο του 2020, λόγω της πανδημίας COVID-19, διακόπηκε η εξέταση «USMLE Step 2 Clinical Skills» για την προστασία της δημόσιας υγείας. Η εξέταση αυτή, απαιτεί στενή επαφή μεταξύ των εξεταζόμενων και του προσωπικού των εξετάσεων και ειδικά με τους τυποποιημένους ασθενείς. Η επανέναρξης αυτής της εξέτασης έγινε με σκοπό τη μείωση του κινδύνου μόλυνσης και της γενικότερης πρόθεσης του NBME για την αξιοποίηση της τεχνολογίας, η οποία μπορεί να περιλαμβάνει ακουστικά και οπτικά μέσα, επεξεργασία φυσικής γλώσσας, τεχνητή νοημοσύνη και άλλους συνδυασμούς επιστήμης και τεχνολογίας.

Το NBME από το 2003, προσπαθεί να χρησιμοποιήσει τις τεχνολογικές επιτεύξεις και την NLP για να μετριαστούν οι προκλήσεις που σχετίζονται με τους ανθρώπους που αξιολογούν τις εξετάσεις. Πιο συγκεκριμένα, ερεύνησε την σκοπιμότητα χρήσης της NLP για την εξαγωγή πληροφοριών από ιατρικό κείμενο, π.χ. σημειώσεις ασθενών από την εξέταση «Step 2 CS». Οι αυτοματοποιημένοι αλγόριθμοι που αναπτύχθηκαν δεν είχαν ομοιόμορφη απόδοση σε όλες τις περιπτώσεις, ωστόσο ο αλγόριθμος που βασίζεται στην παλινδρόμηση παρείχε προκαταρκτικά στοιχεία για την αντικατάσταση των ειδικών από τον συγκεκριμένο αλγόριθμο (Swygert et al., 2003).

Παραπάνω από μια δεκαετία λοιπόν, το NBME, στο κομμάτι των σημειώσεων των ασθενών από την εξέταση «USMLE Step 2 Clinical Skills», προσπαθεί να υιοθετήσει την τεχνολογία χωρίς να επηρεαστεί ο χρόνος βαθμολόγησης ή τα τέλη εξέτασης. Ωστόσο, αυτές οι τεχνικές βαθμολόγησης με την αξιοποίηση της NLP δεν έχουν εγκριθεί για την πλήρη εφαρμογή τους από τις κυβερνητικές επιτροπές, οι οποίες αποτελούνται από εκπροσώπους του ιατρικού κανονισμού, της ιατρικής εκπαίδευσης και του κοινού (Salt et al., 2019).

Στόχος πλέον του NBME είναι η προώθηση της αυτοματοποιημένης βαθμολογίας των σημειώσεων των ασθενών και η χρησιμοποίηση της παραπάνω αυτοματοποίησης στην εξέταση «USMLE Step 2 Clinical Skills». Γι’ αυτό το λόγο, προχώρησε στη δημόσια κυκλοφορία ενός μεγάλου σώματος δεδομένων από σημειώσεις ασθενών γραμμένες από τους εξεταζόμενους ιατρούς για ερευνητικούς σκοπούς. Έτσι, οδήγησε στην προσπάθεια γεφύρωσης του χάσματος που υπάρχει μεταξύ της βιοϊατρικής NLP και της έλλειψης διαθέσιμων ιατρικών σημειώσεων. Το σώμα αυτό αποτελείται από 42.146

τμήματα ιστορικού από σημειώσεις ασθενών για 10 ιατρικές περιπτώσεις. Συνολικά παρατηρείται ποικιλομορφία λόγω της δημιουργίας των σημειώσεων από πολλούς εξεταζόμενους ιατρούς, η οποία περιέχει διάφορες ιατρικές ορολογίες, τυπογραφικά λάθη, συντομογραφίες μεταξύ άλλων χαρακτηριστικών.

4.2 Notebooks που αξιοποιήθηκαν

Τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εργασία έγιναν διαθέσιμα με τη μορφή διαγωνισμού από την πλατφόρμα του Kaggle. Μετά το πέρας του διαγωνισμού, οι διάφορες λύσεις που προτάθηκαν έγιναν διαθέσιμες και στο ευρύ κοινό. Απόρροια αυτού, να είχαμε στην κατοχή μας έτοιμους κώδικες που θα βοηθούσαν στη μελέτη και στην δημιουργία του δικού μας μοντέλου.

Στο διαγωνισμό συμμετείχαν 1471 ομάδες και κατ' επέκταση υπάρχουν αρκετές διαφορετικές λύσεις για τον αυτόματο προσδιορισμό ιατρικών εννοιών. Τα notebooks, δηλαδή οι κώδικες, που σημείωσαν μεγαλύτερο σκορ αξιοποιούσαν τα προ-εκπαιδευμένα μοντέλα DeBERTa και Roberta. Παράλληλα, βελτιωμένη απόδοση παρουσίασαν notebooks τα οποία αποτελούν συνδυασμό διαφορετικών λύσεων ή και ακόμα notebooks που προτείνουν μικρές διορθωτικές κινήσεις σε κάποιο υπάρχων.

Στην περιγραφική στατιστική των δεδομένων βασιστήκαμε στο notebook του Sanskar Hasija. Το συγκεκριμένο notebook βοήθησε στην κατανόηση των συνόλων δεδομένων και επιλέχθηκε έναντι των υπολοίπων εξαιτίας της υψηλής προτίμησής του από τους χρήστες. Ύστερα, για την περιγραφή των δέκα διαφορετικών ιατρικών περιπτώσεων (clinical cases), που στην ουσία περιγράφουν τις απαραίτητες ιατρικές πληροφορίες, αξιοποιήσαμε το εξής notebook.

Από την άλλη, για την ανάπτυξη του μοντέλου BERT προχωρήσαμε στον συνδυασμό δύο notebooks. Πιο συγκεκριμένα, η κύρια δομή του κώδικα προήλθε από το notebook του Tomo Hiroh και κατόπιν στο παρακάτω notebook , το οποίο πρότεινε μια αλλαγή στον προηγούμενο κώδικα με αποτέλεσμα τη βελτίωση του αποτελέσματος. Έπειτα, για την συνάρτηση του tokenizer χρησιμοποιήσαμε τον κώδικα του Shudipto Trafder σε συνδυασμό με το notebook, που αποτελεί μια επεξήγηση του BERT tokenizer που εφάρμοσε στο μοντέλο του.

4.3 Περιγραφή των συνόλων δεδομένων

Το NBME και η ομοσπονδία Κρατικών Ιατρικών Συμβουλίων ανέπτυξαν την εξέταση «USMLE», η οποία είναι αποτελεί προϋπόθεση για την απόκτηση ιατρικής άδειας. Η

εξέταση «USMLE Step 2 Clinical Skills» αξιολογεί τις ικανότητες του εξεταζόμενου να αποκτά πληροφορίες, να διεξάγει φυσικές εξετάσεις και να αναλύει δεδομένα μέσα από τις σημειώσεις των ασθενών που συμπλήρωνε μετά από κάθε συνάντηση.

Ο στόχος του διαγωνισμού που δημιουργήθηκε από το NBME είναι ο αυτόματος προσδιορισμός συγκεκριμένων ιατρικών εννοιών. Αναλυτικότερα, ζητήθηκε η ανάπτυξη μιας αυτοματοποιημένης μεθόδου για την αναγνώριση των σχετικών χαρακτηριστικών σε κάθε σημείωμα ασθενούς, όπως τεκμηριώνονται στη συνέντευξη τους από έναν φοιτητή ιατρικής.

Τα σύνολα δεδομένων είναι 4 και αποτελούνται από τα δεδομένα εκπαίδευσης (training data) και από τα δεδομένα δοκιμής (test data), τα οποία αποτελούν παράδειγμα για τη μορφή που πρέπει να έχουν. Πρώτα όμως, είναι απαραίτητο να διευκρινιστούν ορισμένες έννοιες οι οποίες θα χρησιμοποιηθούν κατά την περιγραφή των δεδομένων μας, όπως ορίστηκαν στις οδηγίες του διαγωνισμού.

- **Ιατρική περίπτωση- Clinical case:** το σενάριο, συμπτώματα, παράπονα, ανησυχίες, που παρουσιάζει ο τυποποιημένος ασθενής στον εξεταζόμενο φοιτητή. Υπάρχουν 10 ιατρικές περιπτώσεις στο σύνολο δεδομένων.
- **Σημειώσεις ασθενούς- Patient note:** κείμενο που περιγράφει λεπτομερώς σημαντικές πληροφορίες που σχετίζονται με τον ασθενή κατά τη διάρκεια της συνάντησης με τον γιατρό, που περιλαμβάνει τη φυσική εξέταση και τη συνέντευξη (Εικόνα 7).
- **Χαρακτηριστικό- Feature:** Μια σχετική ιατρική έννοια.

History: Describe the history you just obtained from this patient. Include only information (pertinent positives and negatives) relevant to this patient's problem(s).

Karin Moore is a 45 yo F here for nervousness. A few weeks ago she noticed that she was feeling more nervous than usual and that it has been worsening. It is exacerbated by family and work. She feels espeically nervous on Sunday night and Monday morning when she is preparing for the week. She is unable to fall asleep and doesn't want to eat anything, though she does make herself eat. Nothing helps her nervousness. She otherwise denies significant changes in appetite, weight loss, or overall wellbeing. She denies fevers, chills, nausea, constipation, diarrhea, skin changes, racing heart, shortness of breath, dizziness, headaches or rashes.
ROS: otherwise negative
PMH: None; PSH: None
Meds: Tylenol for occasional HA
FHX: Father had an MI, died at 65yo
Allergies: NKDA
SH: Lives at home with husband, mother, and youngest son. Is an english literature professor at a local college. Has 2 drinks/mo, no tobacco or drug use.

Physical Examination: Describe any positive and negative findings relevant to this patient's problem(s). Be careful to include only those parts of examination you performed in this encounter.

VS: Blood Pressure: 130/85 mm Hg
Heart Rate: 96/min
Gen: No acute distress, conversational, thin
Neck: No thyromegaly, no lymphadcopathy
Heart: RRR, no murmurs, rubs or gallops. Radial pulses +2 bilaterally
Lungs: Clear to ascultation bilaterally, no wheezes
Psych: Well-groomed. Non-pressured speech, linear though process.

Data Interpretation: Based on what you have learned from the history and physical examination, list up to 3 diagnoses that might explain this patient's complaint(s). (...)

General anxiety disorder
Panic disorder
Hyperthyroidism

Εικόνα 7: Παράδειγμα ενός σημειώματος ασθενή. Το σύνολο δεδομένων περιλαμβάνει μόνο το τμήμα του ιστορικού του ασθενή. Πηγή: (Yaneva et al., 2022)

Τα δεδομένα εκπαίδευσης (training data) αποτελούνται από τα αρχεία patient_notes.csv, features.csv και train.csv.

Αρχικά, το patient_notes.csv αποτελείται από 3 στήλες και 42.146 σειρές και καμία ελλείπουσα τιμή. Μια σύντομη περιγραφή των στηλών περιγράφεται στον πίνακα 3:

Πίνακας 3: patient_notes.csv: μια συλλογή από περίπου 40.000 τμήματα ιστορικού από σημειώσεις ασθενών

Όνομα στήλης	*Περιγραφή*
pn_num	Ένα μοναδικό αναγνωριστικό για κάθε σημείωμα ασθενούς
case_num	Ένα μοναδικό αναγνωριστικό για την κλινική περίπτωση που αντιπροσωπεύει ένα σημείωμα ασθενούς
pn_history	Το κείμενο της συνάντησης όπως καταγράφηκε από τον εξεταζόμενο

Ακολούθως, υπάρχουν 3 στήλες και 143 γραμμές στο features.csv. Δεν έχει ελλείπουσα τιμή.

Πίνακας 4: features.csv: τα χαρακτηριστικά ή οι βασικές έννοιες για κάθε ιατρική περίπτωση.

Όνομα στήλης	*Περιγραφή*
feature_num	Ένα μοναδικό αναγνωριστικό για κάθε χαρακτηριστικό
case_num	Ένα μοναδικό αναγνωριστικό για κάθε περίπτωση
feature_text	Η περιγραφή του χαρακτηριστικού

Ολοκληρώνοντας, το train.csv αποτελείται από περίπου 14.300 γραμμές, 6 στήλες, χωρίς ελλείπουσες τιμές. Το σύνολο αυτό περιέχει τα annotations.

Πίνακας 5: train.csv: feature annotation για 1000 σημειώσεις ασθενών, 100 για κάθε μία από τις δέκα περιπτώσεις

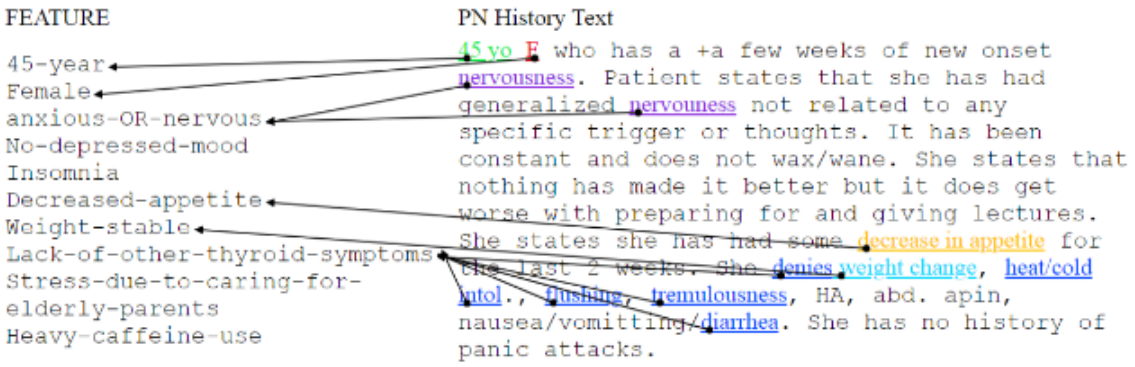
Όνομα στήλης	*Περιγραφή*
id	Μοναδικό αναγνωριστικό για κάθε ζεύγος σημειώσεων/ χαρακτηριστικών ασθενούς
pn_num	Η σημείωση ασθενούς annotated σε αυτή τη σειρά
feature_num	Το χαρακτηριστικό annotated σε αυτή τη σειρά
case_num	Η περίπτωση στην οποία ανήκει αυτό το σημείωμα ασθενούς
annotation	Το κείμενο ή τα κείμενα μέσα σε μια σημείωση ασθενούς που υποδεικνύει ένα χαρακτηριστικό. Ένα χαρακτηριστικό μπορεί να υποδειχθεί πολλές φορές μέσα σε μία μόνο σημείωση
location	Υποδεικνύει τη θέση κάθε annotation μέσα στη σημείωση

Οι σημειώσεις των ασθενών έγιναν annotated από έμπειρους ιατρούς των Ηνωμένων Πολιτειών της Αμερικής (Yaneva et al., 2022). Αρχικά, δόθηκε η οδηγία να διαβάσουν ολόκληρη τη σημείωση του ασθενή και έπειτα να αναγνωρίσουν όλες τις φράσεις που στην ουσία εκφράζουν ένα χαρακτηριστικό και να πραγματοποιήσουν τη σύνδεση τους (Εικόνα 8). Στη συνέχεια, έπρεπε να συμπεριλάβουν τα διαχωρισμένα annotations εξαιρώντας το κείμενο που δεν ήταν σχετικό με το χαρακτηριστικό. Για παράδειγμα, από το παρακάτω απόσπασμα έπρεπε να γίνει annotated μόνο το υπογραμμισμένο

κείμενο: «Έχει δοκιμάσει το Immodium (επιδεινωμένη κατάσταση) και το Cipro 250 mg BID (έχει λάβει 9 δισκία) από προηγούμενο επεισόδιο διάρροιας στην Κένυα, μικρότερης σοβαρότητας (καμία επίδραση)». Μία ακόμη οδηγία, είναι ότι το κάθε χαρακτηριστικό πρέπει να επισημαίνεται ως ξεχωριστό annotation, το οποίο θα αποτελείται από τις σημαντικότερες λέξεις κλειδιά και όχι ολόκληρο το κείμενο.

Οι σχολιασμοί περιλαμβάνουν ποσοτικούς δείκτες (δύο, μερικοί), επίθετα που προσδιορίζουν τον βαθμό (ήπιος, σοβαρός) και χρονικούς προσδιορισμούς (δύο βδομάδες, χρόνια). Αντιθέτως, δεν περιλαμβάνουν άρθρα ή αναφορές στον ασθενή (αυτός, αυτή) ή σημεία στίξης. Με άλλα λόγια, το φύλο είναι μια περίπτωση που πρέπει να γίνεται annotated μόνο μια φορά στην αρχή.

Τέλος, στις οδηγίες ανέφεραν ότι τα annotations μπορεί να επικαλύπτονται, δηλαδή στη φράση «Αρνητικός για πυρετό, ρίγη, ναυτία, εμετό» τα ουσιαστικά αναφέρονται σε διαφορετικά χαρακτηριστικά και πρέπει να γίνονται ως «Αρνητικός για πυρετό», «Αρνητικός για ρίγη» κ.λπ..



Εικόνα 8: Παράδειγμα μιας σημείωσης ασθενή, τα χαρακτηριστικά και η έκφραση τους στα annotations. Πηγή: (Yaneva et al., 2022)

Από την άλλη, τα test data αποτελούν παραδείγματα περιπτώσεων επιλεγμένα από το training set. Παρουσιάζεται η μορφή των δεδομένων έτσι όπως πρέπει να είναι στη διαδικασία της εκπαίδευσης. Αποτελείται από 5 γραμμές και 4 στήλες, σύνολο 20 παρατηρήσεις (χωρίς ελλείπουσες τιμές).

Πίνακας 6: test.csv: παραδείγματα περιπτώσεων επιλεγμένα από το training set

Όνομα στήλης	*Περιγραφή*
id	Μοναδικό αναγνωριστικό για κάθε ζεύγος σημειώσεων/ χαρακτηριστικών ασθενούς
case_num	Η περίπτωση στην οποία ανήκει αυτό το σημείωμα

	ασθενούς
pn_num	Η σημείωση ασθενούς annotated σε αυτή τη σειρά
feature_num	Το χαρακτηριστικό annotated σε αυτή τη σειρά

4.4 Περιγραφική στατιστική

Ο στόχος του διαγωνισμού είναι να προβλέψουμε το location, που στην ουσία αφορά το annotation, με τη βοήθεια των features και της στήλης feature text (περιγραφή του χαρακτηριστικού της εκάστοτε ιατρικής περίπτωσης) και των patient_notes και της στήλης pn_history (το κείμενο όπως καταγράφηκε από τον εξεταζόμενο στη διάρκεια της συνάντησης με τον ασθενή) από τα patient notes.

Αρχικά, αξίζει να γίνει αναφορά στα clinical cases, σύνολο 10, που υπάρχουν και καλύπτουν διάφορους ιατρικούς τομείς, όπως γυναικεία υγεία, γαστρεντερικές περιπτώσεις, νευρολογικές, ψυχιατρικές και καρδιαγγειακές (Yaneva et al., 2022). Ενώσαμε τα datasets των patient notes και features με το train για ευκολότερη κατανόηση και ομαδοποιήσαμε τα δεδομένα με βάση τα cases και μπορέσαμε να διακρίνουμε το γενικό πλαίσιο του κάθε ένα, το οποίο παρουσιάζεται στον παρακάτω πίνακα.

Πίνακας 7: Ιατρικές περιπτώσεις

Ιατρική περίπτωση	*Γενικό πλαίσιο*
Case 0	Περιπτώσεις σχετικές με τους παλμούς
Case 1	Κοιλιακό άλγος
Case 2	Συμπτώματα έμμηνου ρήσης
Case 3	Πόνος στην περιοχή του στομαχιού
Case 4	Υπερβολική νευρικότητα/άγχος
Case 5	Επεισόδια σχετικά με τους παλμούς και αισθήματα επικείμενης καταστροφής
Case 6	Πόνοι στο στήθος σε νεαρά άτομα
Case 7	Ακανόνιστος έμμηνος κύκλος
Case 8	Συμπτώματα θλίψης
Case 9	Πονοκέφαλος με επιπλέον συμπτώματα

Στο πρώτο βήμα γίνεται εξερεύνηση των training data που αποτελούνται από τα datasets των patient notes, features και train.

4.4.1 Patient Notes

	pn_num	case_num	pn_history
0	0	0	17-year-old male, has come to the student heal...
1	1	0	17 yo male with recurrent palpitations for the...
2	2	0	Dillon Cleveland is a 17 y.o. male patient wit...
3	3	0	a 17 yo m c/o palpitation started 3 mos ago; \...
4	4	0	17yo male with no pmh here for evaluation of p...

Εικόνα 9: Οι 5 πρώτες γραμμές του patient_notes

Ένα παράδειγμα του κειμένου που καταγράφηκε από τον εξεταζόμενο κατά τη συνάντηση του με τον τυποποιημένο ασθενή αποτελεί η Εικόνα 10:

```
HPI: 17yo M presents with palpitations. Patient reports 3-4 months of intermittent episodes of "heart
beating/pounding out of my chest." 2 days ago during a soccer game had an episode, but this time had
chest pressure and felt as if he were going to pass out (did not lose conciousness). Of note patient
endorses abusing adderall, primarily to study (1-3 times per week). Before recent soccer game, took a
dderall night before and morning of game. Denies shortness of breath, diaphoresis, fevers, chills, h
eadache, fatigue, changes in sleep, changes in vision/hearing, abdominal paun, changes in bowel or ur
inary habits.
PMHx: none
Rx: uses friends adderrall
FHx: mom with "thyroid disease," dad with recent heart attcak
All: none
Immunizations: up to date
SHx: Freshmen in college. Endorses 3-4 drinks 3 nights / week (on weekends), denies tabacco, endorses
trying marijuana. Sexually active with girlfriend x 1 year, uses condoms
```

Εικόνα 10: Patient history για τυχαίο ασθενή

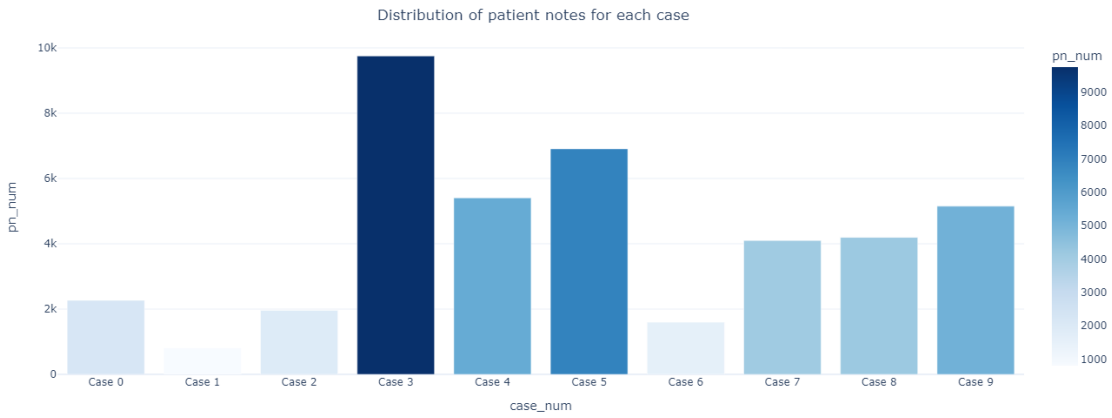
Η στήλη pn_history στην ουσία περιγράφει λεπτομερώς της πληροφορίες που προέρχονται τόσο από τη φυσική εξέταση του ασθενή όσο και από την συνέντευξη του στον εκπαιδευόμενο ιατρό. Ενδεικτικά, αναφέρεται εάν ο ασθενής έχει κάποια αλλεργία, αν ακολουθεί κάποια φαρμακευτική αγωγή, το σεξουαλικό ιστορικό και πιο εξειδικευμένες έννοιες όπως FH (Familial History), ROS (Review of Systems) και PMH (Past Medical History) και άλλες.

Ορισμένες φράσεις μπορούν να γραφτούν με διαφορετικούς τρόπους και χρειάζεται προσοχή κατά την επεξεργασία. Για παράδειγμα, η ηλικία μπορεί να αναφέρεται ως «year-old», «yo» ή «y.o». Για την καλύτερη οπτικοποίηση των δεδομένων κειμένου χρησιμοποιήσαμε την τεχνική Word Cloud στην οποία το μέγεθος κάθε λέξης υποδεικνύει τη συχνότητα της, με τις λέξεις: sexually, active, chest, pain, none, PSH,

PMH να αποτελούν τις συχνότερες από το σύνολο των λέξεων που υπάρχουν στο ιστορικό των ασθενών.

Εικόνα 11: Word Cloud για patient history

Στη συνέχεια, εξετάστηκε ο αριθμός των patient notes που υπάρχουν σε κάθε case αλλά και το μήκος της σημείωσης.



Εικόνα 12: Κατανομή patient notes ανά case

Παρατηρείται ότι ο αριθμός των patient notes ανά case είναι άνισα κατανεμημένος. Το case 3 εμφανίζεται τις περισσότερες φορές, 9753, ενώ το case 1 τις λιγότερες με 808 εμφανίσεις. Ακόμη, το μέσο μήκος μιας σημείωσης που καταγράφεται από τον

εκπαιδευόμενο γιατρό είναι 818,176 χαρακτήρες. Η μικρότερη σημείωση έχει 30 χαρακτήρες και η μεγαλύτερη αποτελείται από 950.

4.4.2 Features

Παρόμοια ανάλυση πραγματοποιήθηκε και στα features data. Τα features αποτελούν μια βασική ιατρική έννοια ή αλλιώς ένα συγκεκριμένο χαρακτηριστικό για κάθε ιατρική περίπτωση.

	feature_num	case_num	feature_text
0	0	0	Family-history-of-MI-OR-Family-history-of-myoc...
1	1	0	Family-history-of-thyroid-disorder
2	2	0	Chest-pressure
3	3	0	Intermittent-symptoms
4	4	0	Lightheaded

Εικόνα 13: Οι 5 πρώτες γραμμές του features

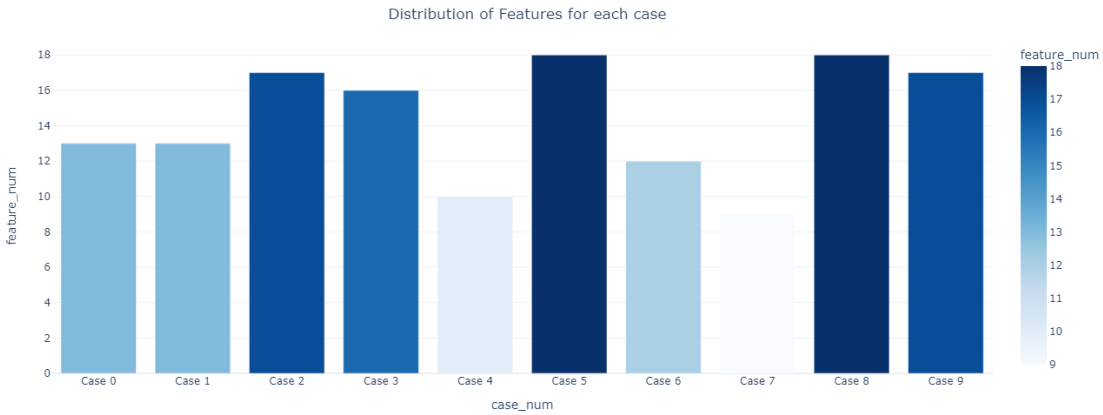
Ορισμένα παραδείγματα χαρακτηριστικών όπως αναγράφονται στο feature_text παρουσιάζονται στην Εικόνα 14 ενώ στην Εικόνα 15 δημιουργήσαμε ένα Word Cloud, με τις πιο συχνά εμφανιζόμενες λέξεις να είναι symptoms, year, months, ago.

```
5     No-hair-changes-OR-no-nail-changes-OR-no-tempe...
6                                          Adderall-use
7                                Shortness-of-breath
8                                          Caffeine-use
9                       heart-pounding-OR-heart-racing
10                               Few-months-duration
Name: feature_text, dtype: object
```

Εικόνα 14: Παραδείγματα της στήλης feature_text

Εικόνα 15: Word Cloud για features

Μέσα από την Εικόνα 16 παρατηρείται ότι ο αριθμός των χαρακτηριστικών ανά περίπτωση είναι λιγότερο άνισα κατανεμημένος συγκριτικά με τα patient notes. Το case 5 και 8 έχουν τη μέγιστη εμφάνιση χαρακτηριστικών που είναι 18 έναντι 9 που συγκεντρώνει το 7ο case. Επίσης, το μέσο μήκος των χαρακτηριστικών μέσα από την αντίστοιχη στήλη του feature_text είναι 23,209 , με ελάχιστο μήκος 3 και μέγιστο το 68.



Εικόνα 16: Κατανομή features ανά case

4.4.3 Train

Το σύνολο εκπαίδευσης αποτελείται από τα annotations που έχουν πραγματοποιηθεί στις σημειώσεις των ασθενών αλλά και την τοποθεσία τους.

	id	case_num	pn_num	feature_num	annotation	location
0	00016_000	0	16	0	['dad with recent heart attcak']	['696 724']
1	00016_001	0	16	1	['mom with "thyroid disease']	['668 693']
2	00016_002	0	16	2	['chest pressure']	['203 217']
3	00016_003	0	16	3	['intermittent episodes', 'episode']	['70 91', '176 183']
4	00016_004	0	16	4	['felt as if he were going to pass out']	['222 258']

Εικόνα 17: Οι 5 πρώτες γραμμές του train

4.4.4 Annotations

Συναντάμε 12.234 annotations και 4.399 κενά annotation (30,76%), πραγματοποιώντας έλεγχο για κενές λίστες ([ ]) στη στήλη location. Ακόμη, το μέσο μήκος των annotations είναι 16,528 χαρακτήρες, με μέγιστους χαρακτήρες σε ένα annotation 198 και εμφανίζεται μία φορά, όπως και όσα έχουν μήκος πάνω από 70 εμφανίζονται το πολύ 4 φορές. Οι πιο κοινές λέξεις που εμφανίζονται στα annotations είναι ago, months, day και weeks (Εικόνα 18).

Εικόνα 18: Word Cloud για τα annotations

4.4.5 Ανάλυση για τυχαίο ασθενή

Αναγκαία κρίθηκε και η ανάλυση για τους ασθενείς καθώς με αυτόν τον τρόπο είναι ευκολότερη η κατανόηση των annotations που υπάρχουν στα train δεδομένα. Υπάρχουν 1000 μοναδικοί τυποποιημένοι ασθενείς, παρακάτω θα ασχοληθούμε με τον ασθενή της Εικόνας 10.

	id	case_num	pn_num	feature_num	annotation	location
0	00016_000	0	16	0	['dad with recent heart attcak']	['696 724']
1	00016_001	0	16	1	['mom with "thyroid disease']	['668 693']
2	00016_002	0	16	2	['chest pressure']	['203 217']
3	00016_003	0	16	3	['intermittent episodes', 'episode']	['70 91', '176 183']
4	00016_004	0	16	4	['felt as if he were going to pass out']	['222 258']
5	00016_005	0	16	5	[]	[]
6	00016_006	0	16	6	['adderall', 'adderrall', 'adderrall']	['321 329', '404 413', '652 661']
7	00016_007	0	16	7	[]	[]
8	00016_008	0	16	8	[]	[]
9	00016_009	0	16	9	['palpitations', 'heart beating/pounding']	['26 38', '96 118']
10	00016_010	0	16	10	['3-4 months of']	['56 69']
11	00016_011	0	16	11	['17yo']	['5 9']
12	00016_012	0	16	12	['M']	['10 11']

Εικόνα 19: Ανάλυση ασθενή με pn_num 16

Για κάθε μοναδικό pn_num υπάρχουν πολλές σειρές που απεικονίζουν πολλά annotations στις σημειώσεις των ασθενών. Με τη στήλη location να υποδεικνύει τη θέση που έχει γίνει το annotation. Οι επόμενες δύο εικόνες, 20 και 21, δείχνουν συγκριτικά το κείμενο της συνάντησης όπως καταγράφηκε από τον εξεταζόμενο γιατρό κατά τη διάρκεια αλληλεπίδρασης του με τον ασθενή και τα annotations που έχουν πραγματοποιηθεί. Γίνεται εύκολα αντιληπτό ότι κάθε annotation αποτελεί μια σημαντική ιατρική έννοια/πληροφορία, η οποία θα βοηθήσει στην εκπαίδευση του μοντέλου για την αυτοματοποίηση της μεθόδου αναγνώρισης σχετικών χαρακτηριστικών σε κάθε σημείωμα ασθενή.

```
Patient Notes -
 HPI: 17yo M presents with palpitations. Patient reports 3-4 months of intermittent episodes of
"heart beating/pounding out of my chest." 2 days ago during a soccer game had an episode, but th
is time had chest pressure and felt as if he were going to pass out (did not lose conciousness).
Of note patient endorses abusing adderall, primarily to study (1-3 times per week). Before recen
t soccer game, took adderrall night before and morning of game. Denies shortness of breath, diap
horesis, fevers, chills, headache, fatigue, changes in sleep, changes in vision/hearing, abdomin
al paun, changes in bowel or urinary habits.
PMHx: none
Rx: uses friends adderrall
FHx: mom with "thyroid disease," dad with recent heart attcak
All: none
Immunizations: up to date
SHx: Freshmen in college. Endorses 3-4 drinks 3 nights / week (on weekends), denies tabacco, end
orses trying marijuana. Sexually active with girlfriend x 1 year, uses condoms
------------
Annotaions:
 ['dad with recent heart attcak']
 ['mom with "thyroid disease']
 ['chest pressure']
 ['intermittent episodes', 'episode']
 ['felt as if he were going to pass out']
 []
 ['adderall', 'adderrall', 'adderrall']
 []
 []
 ['palpitations', 'heart beating/pounding']
 ['3-4 months of']
 ['17yo']
 ['M']
```

Εικόνα 20: Patient note και τα αντίστοιχα annotations

HPI: 17yo Annotation M Annotation presents with palpitations Annotation . Patient reports 3-4 months of Annotation intermittent episodes Annotation of " heart beating/pounding Annotation out of my chest." 2 days ago during a soccer game had an episode Annotation , but this time had chest pressure Annotation and felt as if he were going to pass out Annotation (did not lose conciousness). Of note patient endorses abusing adderall Annotation , primarily to study (1-3 times per week). Before recent soccer game, took adderrall Annotation night before and morning of game. Denies shortness of breath, diaphoresis, fevers, chills, headache, fatigue, changes in sleep, changes in vision/hearing, abdominal paun, changes in bowel or urinary habits.

PMHx: none

Rx: uses friends adderrall Annotation

FHx: mom with "thyroid disease Annotation ," dad with recent heart attcak Annotation

All: none

Immunizations: up to date

SHx: Freshmen in college. Endorses 3-4 drinks 3 nights / week (on weekends), denies tabacco, endorses trying marijuana. Sexually active with girlfriend x 1 year, uses condoms

Εικόνα 21: Patient note και τα αντίστοιχα annotations (2)

4.5 Περιγραφή του μοντέλου BERT

4.5.1 Εισαγωγή

Σε αυτήν την ενότητα θα γίνει εκτενέστερη περιγραφή της αρχιτεκτονικής του προ-εκπαιδευμένου μοντέλου BERT. Θα αναλυθούν οι τεχνικές εκπαίδευσης του μοντέλου, δηλαδή οι διαδικασίες MLM και NSP και θα αναφερθούν τα πλεονεκτήματα της επιλογής του συγκεκριμένου μοντέλου. Με αυτόν τον τρόπο, θα γίνει ευκολότερα κατανοητός ο τρόπος λειτουργίας του προ-εκπαιδευμένου μοντέλου BERT και θα βοηθήσει στην ανάλυση δεδομένων που θα ακολουθήσει.

4.5.2 Αρχιτεκτονική του μοντέλου BERT

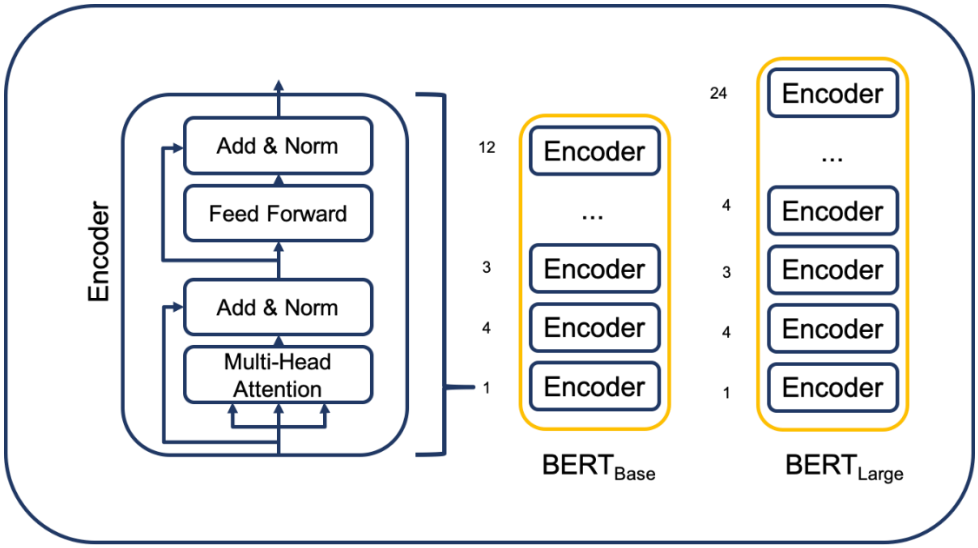
Σε όλες τις φυσικές γλώσσες η σημασία μιας λέξης ποικίλει και καθορίζεται από το πλαίσιο στο οποίο βρίσκεται. Η μοντελοποίηση αυτού του χαρακτηριστικού νοήματος ήταν και συνεχίζει να είναι μία από τις θεμελιώδεις δυσκολίες που έχει να αντιμετωπίσει η επεξεργασία φυσικής γλώσσας. Πρόσφατα η επεξεργασία φυσικής γλώσσας έκανε τεράστια άλματα στην κατανόηση της ανθρώπινης γλώσσας με την ανάπτυξη ενός μοντέλου, που δημιουργήθηκε από ερευνητές της Google το 2018, και ονομάζεται Bidirectional Encoder Representations from Transformer ή αλλιώς BERT.

Όπως έχει ήδη αναφερθεί σε προηγούμενη ενότητα και βάσει των (Devlin et al., 2019), το BERT είναι ένα γλωσσικό μοντέλο το οποίο έχει εκπαιδευτεί σε ένα μεγάλο σύνολο δεδομένων κειμένου και αποτελεί ένα από τα πιο διάσημα και ευρέως χρησιμοποιούμενα NLP μοντέλα. Είναι ένα μοντέλο ανοιχτού κώδικα που έχει κατορθώσει να δώσει λύση σε πολλά γλωσσικά προβλήματα. Μερικά από τα οποία είναι η γενική κατανόηση της γλώσσας όπως εξαγωγή συμπερασμάτων φυσικής γλώσσας (natural language inference), ανάλυση συναισθημάτων (sentiment analysis), απάντηση ερωτήσεων (question answering), ανίχνευση παράφρασης (paraphrase detection) και γλωσσική αποδοχή (linguistic acceptability).

Αυτό έχει επιτευχθεί καθώς εκπαιδεύεται, σε αντίθεση με προηγούμενα μοντέλα, αμφίδρομα μέσω πολλαπλών επιπέδων χρησιμοποιώντας νευρωνικά δίκτυα Transformer και έχει την ικανότητα να μαθαίνει τις λέξεις και τα συμφραζόμενα τους κατά της διάρκειας της προ-εκπαίδευσης. Από την άλλη, η δημιουργία του μοντέλου βάσει της αρχιτεκτονικής Transformer, έχει συνδράμει στην εφικτή εκπαίδευση του BERT σε μεγάλες ποσότητες δεδομένων σε σχετικά σύντομο χρονικό διάστημα (Xu et al., 2022). Ο μετασχηματιστής περιλαμβάνει, όπως έχει ειπωθεί στην ενότητα 1.4, έναν κωδικοποιητή και έναν αποκωδικοποιητή. Ο κωδικοποιητής διαβάζει το κείμενο

εισόδου και ο αποκωδικοποιητής προβλέπει για το συγκεκριμένο πρόβλημα. Το μοντέλο BERT όμως, κάνει χρήση μόνο του κωδικοποιητή, διαφοροποιώντας σε ένα βαθμό την αρχιτεκτονική του από αυτή των Transformers.

Αναλυτικότερα, η αρχιτεκτονική του μοντέλου είναι ένας πολύ-επίπεδος αμφίδρομος κωδικοποιητής (encoder) Transformer, όπως περιγράφεται στο «Attention is all you need» (Vaswani et al., 2017). Ένα χαρακτηριστικό που του δίνει τη δυνατότητα να μαθαίνει το περιεχόμενο της κάθε λέξης βάσει των λέξεων που βρίσκονται αριστερά και δεξιά αυτής. Πιο λεπτομερής αναφορά στην αρχιτεκτονική θα γίνει στα δύο βασικά μοντέλα BERT, στο $\text{BERT}_{\text{BASE}}$ και στο $\text{BERT}_{\text{LARGE}}$.



Εικόνα 22: Αρχιτεκτονική του μοντέλου BERT. Στοίβα του κωδικοποιητή. Πηγή: https://humboldt-wi.github.io/blog/research/information_systems_1920/bert_blog_post/

Ορολογία:

Transformer Layers: ο αριθμός μετασχηματιστών μπλοκ (block Transformer). Ένας μπλοκ μετασχηματιστής μετατρέπει μια ακολουθία αναπαραστάσεων λέξεων σε μια ακολουθία λέξεων με βάση τα συμφραζόμενα.

Hidden Size: τα επίπεδα μαθηματικών συναρτήσεων, που βρίσκονται μεταξύ της εισόδου και της εξόδου.

Attention Heads: το μέγεθος ενός μπλοκ μετασχηματιστή (Transformer).

Parameters: ο αριθμός των μεταβλητών/παραμέτρων που είναι διαθέσιμες κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

Processing: ο τύπος της μονάδας επεξεργασίας που χρησιμοποιείται για την εκπαίδευση του μοντέλου.

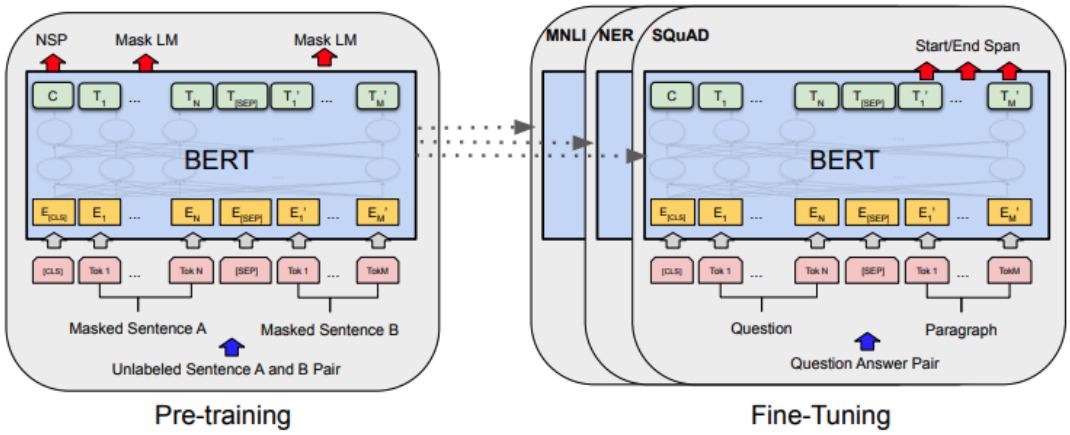
Length of Training: ο χρόνος που χρειάστηκε για να εκπαιδευτεί το μοντέλο.

Πίνακας 8: Αρχιτεκτονική του $BERT_{BASE}$ και του $BERT_{LARGE}$

	Transformer Layers	*Hidden Size*	*Attention Heads*	*Parameters*	*Processing*	*Length of Training*
$BERT_{BASE}$	12	768	12	110M	4 TPUs	4 days
$BERT_{LARGE}$	24	1024	16	340M	16TPUs	4 days

Η αρχιτεκτονική των δύο αυτών εκδόσεων είναι πανομοιότυπη με την αρχιτεκτονική των μετασχηματιστών (Vaswani et al., 2017) με μόνη διαφορά των αριθμό των στρωμάτων, δηλαδή των transformer layers, των hidden sizes και των attention heads. Ένας μετασχηματιστής έχει 6 transformers layers, 512 hidden sizes και 8 attention heads. Ενώ το $BERT_{BASE}$ διαθέτει 12, 768 και 12 και το $BERT_{LARGE}$ 24, 1024 και 16 αντιστοίχως.

Εκτός από τα παραπάνω, η επιτυχία του μοντέλου BERT βασίζεται και στην προ-εκπαίδευση του μοντέλου (pre-training) και στην προσαρμογή αυτού (fine-tuning) (Devlin et al., 2019).



Εικόνα 23: Συνολικές διαδικασίες pre-training και fine-tuning για το BERT. Εκτός από τα επίπεδα εξόδου (output layers) οι ίδιες αρχιτεκτονικές χρησιμοποιούνται και στα δύο στάδια. Για διαφορετικά μοντέλα χρησιμοποιούνται οι ίδιες pre-trained παράμετροι. Πηγή: (Devlin et al., 2019)

Για την προ-εκπαίδευση χρησιμοποιούνται οι παρακάτω δύο στρατηγικές, το γλωσσικό μοντέλο απόκρυψης και η πρόβλεψη της επόμενης πρότασης.

- **Γλωσσικό Μοντέλο Απόκρυψης-Masked Language Model (MLM)**
  Η μέθοδος αυτή καλύπτει ή αποκρύπτει μια λέξη σε μια φράση και απαιτεί από το μοντέλο BERT να χρησιμοποιήσει τις λέξεις σε κάθε πλευρά της κρυμμένης λέξης με στόχο να την προβλέψει. Στην ουσία επιβάλλει την αμφίδρομη μάθηση από το κείμενο. Κατά τη διάρκεια της εκπαίδευσης αντικαθιστά το 15% των αρχικών λέξεων με το σύμβολο [MASK], αποκρύπτει δηλαδή τη λέξη, και το BERT προσπαθεί να προβλέψει τις κρυμμένες λέξεις. Η στρατηγική που ακολουθείται για να εξαλειφθεί η αναντιστοιχία μεταξύ της προ-εκπαίδευσης και του fine-tuning είναι: κατά 80% η λέξη αντικαθίσταται πραγματικά με τη σωστή λέξη, 10% των φορών αντικαθίσταται με μία τυχαία λέξη και 10% των φορών η λέξη μένει αμετάβλητη.
- **Πρόβλεψη της Επόμενης Πρότασης-Next Sentence Prediction (NSP)**
  Η μέθοδος αυτή χρησιμοποιείται για να βοηθήσει το μοντέλο BERT να μάθει για τις σχέσεις μεταξύ των προτάσεων, προβλέποντας εάν μια δεδομένη πρόταση ακολουθεί την προηγούμενη ή όχι. Στη διάρκεια της εκπαίδευσης το μοντέλο τροφοδοτείται με δύο προτάσεις εισόδου έτσι ώστε η δεύτερη πρόταση να έρχεται μετά την πρώτη κατά το 50% των φορών και χαρακτηρίζεται ως «IsNext» ή μπορεί να είναι μία τυχαία πρόταση από το σύνολο δεδομένων κατά 50%, με την ένδειξη «NotNext». Κατά αυτόν τον τρόπο αυξάνεται η προβλεπτική ακρίβεια του μοντέλου περίπου 97-98%.

Από την άλλη, η διαδικασία του fine-tuning είναι απλή, και το μοντέλο μπορεί να βελτιωθεί με σχετικά λίγες παραμετροποιήσεις. Αυτό συμβαίνει καθώς οι περισσότερες υπερ-παράμετροι (hyperparameters) του μοντέλου είναι οι ίδιες με αυτές της προ-εκπαίδευσης του, με εξαίρεση το μέγεθος παρτίδας (batch size), το ρυθμό εκμάθησης (learning rate) και τον αριθμό των πλήρων περασμάτων του συνόλου δεδομένων εκπαίδευσης (training epochs). Η διαδικασία μπορεί να εκτελεστεί πολύ γρήγορα, αρκεί να γίνει μια εξαντλητική αναζήτηση στις παραπάνω παραμέτρους έτσι ώστε να επιλεγεί το μοντέλο που αποδίδει καλύτερα.

Εν κατακλείδι, το BERT είναι ένα ισχυρό και δημοφιλές εργαλείο της επεξεργασίας φυσικής γλώσσας που έχει χρησιμοποιηθεί από πολλούς ερευνητές. Μερικά από τα πιο δεδομένα προ-εκπαιδευμένα μοντέλα BERT είναι ανάλυση συναισθήματος των tweets μέσα από τη πλατφόρμα του Twitter, κατηγοριοποίηση συναισθημάτων, όπως θυμός, φόβος, χαρά κλπ., ανάλυση σε κλινικές σημειώσεις και μετάφραση ομιλίας σε κείμενο.

Τέλος, η μηχανική μάθηση απαιτεί τεράστιες ποσότητες δεδομένων για την επίτευξη καλύτερης προβλεπτικής ακρίβειας, κάτι το οποίο είναι ακριβό τόσο σε χρονικούς όσο και σε υπολογιστικούς πόρους. Ωστόσο, οι ερευνητές έχουν τη δυνατότητα να το αξιοποιήσουν γρήγορα το μοντέλο BERT χωρίς να δαπανήσουν πολύ χρόνο και χρήματα, καθώς ο κώδικας του διατίθεται δωρεάν στο GitHub.

Κεφάλαιο 5- Ανάλυση δεδομένων και ερμηνεία αποτελεσμάτων

5.1 Ανάπτυξη του μοντέλου BERT

Όπως έχει ήδη αναφερθεί ο σκοπός του προβλήματος είναι η αυτοματοποίηση του προσδιορισμού των κλινικών εννοιών όπως αναγράφονται στις σημειώσεις των ιατρών κατά τη διάρκεια αλληλεπίδρασής τους με τους ασθενείς. Ο στόχος είναι η πρόβλεψη της στήλης annotation και κατ' επέκταση της στήλης location. Η πρόβλεψη τους βασίζεται στο ιατρικό ιστορικό, pn_history, και στην περιγραφή του ιατρικού χαρακτηριστικού, feature_text. Για την καλύτερη κατανόηση των δεδομένων, ενώσαμε τα σύνολα δεδομένων, το train με το feature και το patient_notes, με αποτέλεσμα τα train data να είναι όπως η Εικόνα 24. Ίδια λογική ακολουθήθηκε και για τα test data.

	id	case_num	pn_num	feature_num	annotation	location	feature_text	pn_history
0	00016_000	0	16	0	['dad with recent heart attcak']	['696 724']	Family-history-of-MI-OR-Family-history-of-myoc...	HPI: 17yo M presents with palpitations. Patien...
1	00016_001	0	16	1	['mom with "thyroid disease']	['668 693']	Family-history-of-thyroid-disorder	HPI: 17yo M presents with palpitations. Patien...
2	00016_002	0	16	2	['chest pressure']	['203 217']	Chest-pressure	HPI: 17yo M presents with palpitations. Patien...

Εικόνα 24: Οι 3 πρώτες εγγραφές του ενωμένου συνόλου δεδομένων train

Η επιθυμητή πρόβλεψη πρέπει να είναι όπως η Εικόνα 25. Για την ακρίβεια, για τον ασθενή με pn_num 16 έχουμε δει στις εικόνες 20 και 21 τη θέση που γίνεται το annotation. Μετά την ένωση, στο σύνολο δεδομένων train μπορούμε να διακρίνουμε πλέον και το ιατρικό χαρακτηριστικό (feature), δηλαδή την ιατρική έννοια, που συνδέεται με το συγκεκριμένο annotation. Για παράδειγμα, όπως φαίνεται και παρακάτω, για το annotation «intermittent episodes» αντιστοιχεί η ιατρική έννοια «intermittent symptoms», για τη φράση «felt as if he were going to pass out» η αντίστοιχη μετάφραση σε ιατρικό χαρακτηριστικό είναι «Lightheaded». Στην ουσία θέλουμε να προβλέψουμε τη θέση του «intermittent episodes» η οποία αναφέρεται ως «intermittent symptoms» στο σύνολο δεδομένων features.

Εικόνα 25: Τα annotations και τα αντίστοιχα feature_text. Πηγή: https://www.kaggle.com/code/theoviel/roberta-strikes-back

5.1.1 Παραλλαγές του μοντέλου BERT

Στην προηγούμενη ενότητα αναφέρθηκε ότι το μοντέλο BERT κυκλοφόρησε αρχικά σε δύο παραλλαγές, τη $\text{BERT}_{\text{BASE}}$ και $\text{BERT}_{\text{LARGE}}$, ενώ στη συνέχεια αναπτύχθηκαν εκδόσεις και σε άλλες γλώσσες αλλά και μικρότερα μοντέλα. Παράλληλα, υπάρχουν παραλλαγές που κάνουν διάκριση μεταξύ των πεζών και των κεφαλαίων γραμμάτων, cased και uncased. Για παράδειγμα εντοπίζεται διαφορά μεταξύ των λέξεων «Λέξη» και «λέξη» στο cased μοντέλο BERT. Σε αυτήν την εργασία επιλέξαμε να κάνουμε χρήση του $\text{BERT}_{\text{BASE}}$ uncased, το οποίο είναι διαθέσιμο μέσα από την ιστοσελίδα «Hugging Face». Τα uncased μοντέλα είναι ως συνήθως καλύτερα συγκριτικά με τα cased, εκτός εάν τίθεται από το πρόβλημα ότι τα κεφαλαία γράμματα είναι σημαντικά, που γίνεται συχνά στις εφαρμογές Named Entity Recognition. Το μοντέλο μας έχει 12 transformer layers, 768 hidden sizes, 12 attention heads και 110M παραμέτρους.

5.5.2 BERT Tokenizer

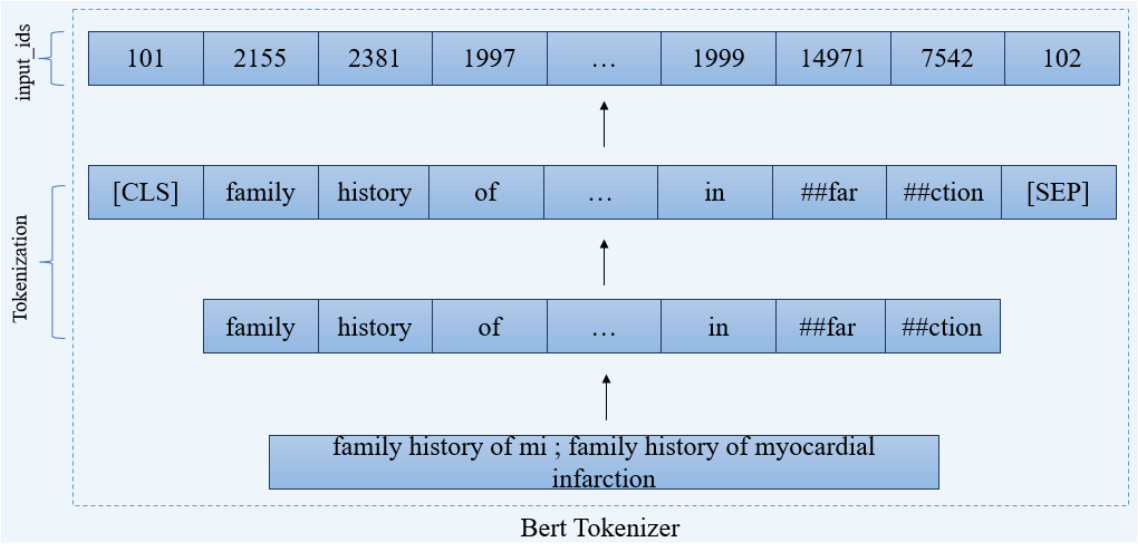
Αρχικά, είναι αναγκαίο να αναφέρουμε ότι το μοντέλο BERT μπορεί να χρησιμοποιηθεί με 3 διαφορετικούς τρόπους και το αποτέλεσμα της πρόβλεψης να παραμένει το ίδιο. Μπορεί να γίνει χρήση απευθείας με pipeline στον κώδικα για

Masked Language Modeling: $from\ transformers\ import\ pipeline$, μέσα στη δομή (framework) του PyTorch: $from\ transformers\ import\ BertTokenizer, BertModel$ και μέσα στη δομή του TensorFlow: $from\ transformers\ import\ BertTokenizer, TFBertModel$. Στην έρευνα μας, χρησιμοποιήσαμε τη δομή του PyTorch.

Προτού όμως εισάγουμε τα δεδομένα μας και πιο συγκεκριμένα τα δεδομένα κειμένου στο μοντέλο BERT, πρέπει να τα προ-επεξεργαστούμε έτσι ώστε να είναι στην μορφή που απαιτεί το μοντέλο. Το πρώτο βήμα είναι να χωρίσουμε σε λέξεις και υπο-λέξεις τις προτάσεις στη μορφή που το μοντέλο BERT τις αναγνωρίζει. Το tokenizer είναι υπεύθυνο για την προετοιμασία των δεδομένων εισόδου. Υπάρχουν δύο μορφές που μπορεί να υλοποιηθεί, η κανονική και η γρήγορη (Fast) που βασίζεται στη βιβλιοθήκη Rust Tokenizers. Εδώ θα κάνουμε χρήση του $BertTokenizerFast$, καθώς επιταχύνεται η διαδικασία και παρέχονται επιπρόσθετες μέθοδοι, για παράδειγμα η λήψη του εύρους των χαρακτήρων που αντιστοιχούν σε ένα συγκεκριμένο token. Επίσης, κληρονομεί από το PreTrainedTokenizerFast τις περισσότερες από τις κύριες μεθόδους, δηλαδή πραγματοποιεί tokenizing, δηλαδή διαχωρίζονται οι συμβολοσειρές (strings) σε συμβολοσειρές υπο-λέξεων, προσθέτει νέα tokens στο λεξιλόγιο και δίνει προσοχή στη διαχείριση των ειδικών tokens.

Το BertTokenizer όπως και το BertTokenizerFast, χειρίζονται όλες τις αλλαγές που απαιτούνται στα δεδομένα κειμένου για να εξασφαλίσουν ότι το κείμενο εισαγωγής είναι κατάλληλο για χρήση ως είσοδο στο μοντέλο BERT. Προσθέτει ταυτόχρονα, τα tokens [CLS], για να υποδείξει την αρχή της πρότασης, [SEP] για να χωρίσει πολλές προτάσεις και [PAD] για να μετατρέψει τις προτάσεις να έτσι ώστε να έχουν τον ίδιο αριθμό από tokens.

Μετά την εφαρμογή του BertTokenizerFast, εξασφαλίσαμε η κάθε πρόταση να έχει το ίδιο μέγεθος, δηλαδή να έχει τον ίδιο αριθμό tokens, με την εντολή padding και σε συνδυασμό με την εντολή truncation, εάν μια πρόταση είναι πολύ μεγάλη να κόβεται. Το αποτέλεσμα αυτής της διαδικασίας είναι η δημιουργία ενός λεξικού που περιέχει τα input_ids, τα οποία είναι τα id για κάθε token, τα token_type_ids, που παίρνει τις τιμές 0 και 1, ανάλογα σε ποια ακολουθία ανήκει το token και το attention_mask, που έχει τις τιμές 0 και 1, με το πρώτο να δηλώνει ότι το token είναι συμπλήρωμα/padding δηλαδή [PAD] και το δεύτερο να προσδιορίζει ότι ένα token περιέχει πραγματική λέξη ή [CLS] ή και [SEP].



Εικόνα 26: Παράδειγμα BERT Tokenizer

Στην συνέχεια, δημιουργήσαμε ετικέτες (labels) καθώς είναι ένα κρίσιμο βήμα που επηρεάζει άμεσα την προβλεπτική ακρίβεια του μοντέλου. Οι προβλέψεις του μοντέλου θα είναι λανθασμένες εάν ένα σημαντικό κλάσμα του συνόλου δεδομένων της εκπαίδευσης έχει εσφαλμένα labels.

5.5.3 Κατασκευή του μοντέλου

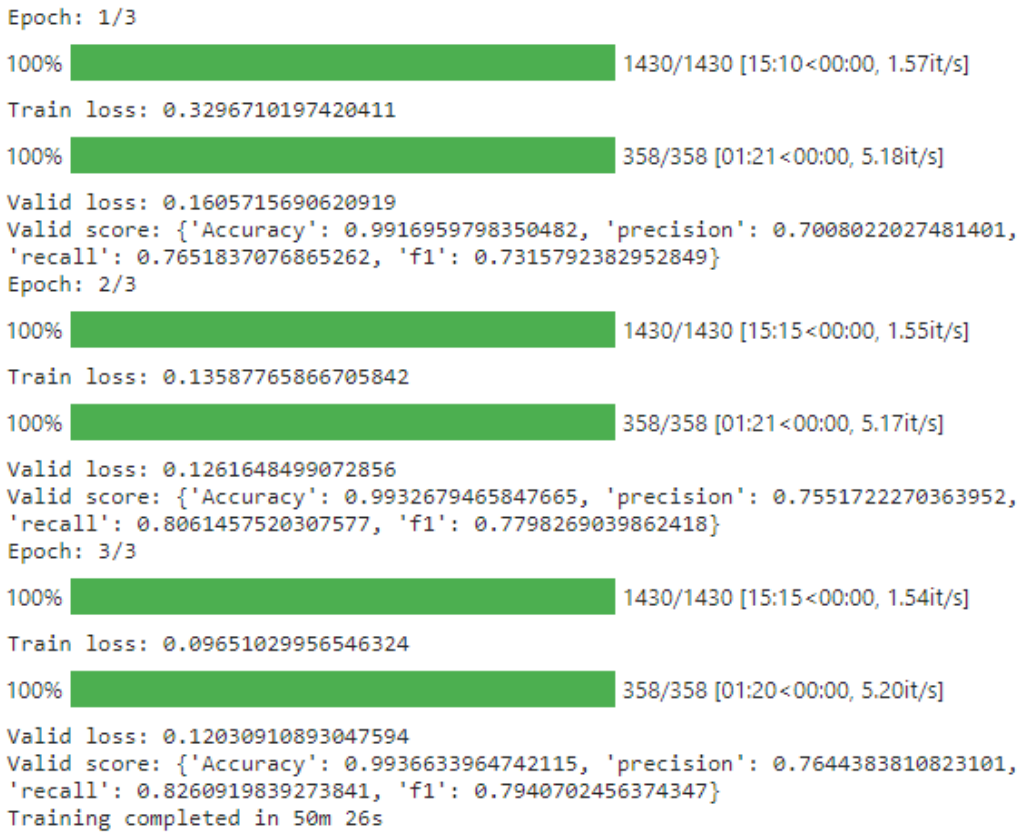
Έπειτα, κατασκευάσαμε το μοντέλο που χρησιμοποιήσαμε, το $\text{BERT}_{\text{BASE}}$ uncased, το οποίο είναι προ-εκπαιδευμένο και έχει 12 επίπεδα κωδικοποιητών Transformer. Κάναμε χρήση ενός γραμμικού ταξινομητή (linear classifier), στον οποίο δώσαμε ως είσοδο την έξοδο του μοντέλου BERT.

Το επόμενο βήμα, ήταν η εκπαίδευση του μοντέλου, σύμφωνα με τη διαδικασία εκπαίδευσης του PyTorch. Η διαδικασία του fine-tuning του μοντέλου, όπως έχει ειπωθεί και στην ενότητα 4.3, είχε διαφοροποιήσεις στις παραμέτρους που αφορούν το μέγεθος της παρτίδας (batch size), που ορίσαμε να είναι 8, του ρυθμού εκπαίδευσης (learning rate) και των αριθμό των πλήρων περασμάτων του συνόλου δεδομένων (epochs).

Ο συνολικός αριθμός των επαναλήψεων (epochs) των δεδομένων εκπαίδευσης ορίστηκε να είναι 3. Ως βελτιστοποιητή (optimizer) ορίσαμε τον AdamW, ο οποίος βασίζεται στον Adam και δίνει καλύτερα αποτελέσματα συγκριτικά με τους υπόλοιπους αλγορίθμους βελτιστοποίησης. Οι (Loshchilov & Hutter, 2019) δείξανε ότι η εκδοχή του Adam με αποσυνδεδεμένη μείωση βάρους (decoupled weight decay), δηλαδή ο

AdamW, έχει σημαντικά καλύτερη απόδοση γενίκευσης συγκριτικά με του Adam με κανικοποίηση L2. Ορίσαμε ως ρυθμό εκπαίδευσης 1e-5.

Επίσης, χρησιμοποιήσαμε την δυαδική διασταυρούμενη εντροπία (Binary Cross-entropy loss on logits) ως συνάρτηση απώλειας. Ένα σημαντικό μέρος ενός νευρωνικού δικτύου είναι συναρτήσεις απώλειας, οι οποίες στην ουσία υπολογίζουν πόσο κακή απόδοση έχει ένα μοντέλο, ή πόσο μεγάλη είναι η απώλεια του. Αυτή η συνάρτηση απώλειας συνδυάζει ένα Sigmoid layer και το BCEloss σε μία μόνο κατηγορία, πράγμα που δημιουργεί μια σταθερότητα καθώς οι πράξεις συνδυάζονται σε ένα επίπεδο και δεν ακολουθείται η μία από την άλλη (*BCEWithLogitsLoss — PyTorch 1.13 Documentation*, n.d.).



Εικόνα 27: Αποτελέσματα training loss και validation loss μετά από 3 epochs

Παρατηρούμε ότι στην αρχή το train loss (απώλεια εκπαίδευσης) ήταν μεγαλύτερο από το valid loss, που υποδηλώνει ότι το μοντέλο δεν μπορούσε να αποδώσει στο σύνολο των δεδομένων εκπαίδευσης (underfitting). Στη συνέχεια όμως των επαναλήψεων, παρατηρούμε ότι το train loss και το valid loss μειώνονται και τα δύο, και στην τελευταία επανάληψη, η οποία είναι και η βέλτιστη, είναι ίσο με 0,096 και 0.12 αντιστοίχως.

5.5.4 Αξιολόγηση του μοντέλου

Εφόσον το μοντέλο έχει εκπαιδευτεί, μπορούμε τώρα να χρησιμοποιήσουμε τα δεδομένα εκπαίδευσης για να αξιολογήσουμε την απόδοση του μοντέλου. Η συνάρτηση για την εκπαίδευση και αξιολόγηση της απόδοσης του μοντέλου στο δοκιμαστικό σύνολο φαίνεται παρακάτω.

```
best_loss = np.inf

for i in range(epochs):
    print("Epoch: {}/{}".format(i + 1, epochs))
    # first train model
    train_loss = train_model(model, train_dataloader, optimizer, criterion)
    train_loss_data.append(train_loss)
    print(f"Train loss: {train_loss}")
    # evaluate model
    valid_loss, score = eval_model(model, test_dataloader, criterion)
    valid_loss_data.append(valid_loss)
    score_data_list.append(score)
    print(f"Valid loss: {valid_loss}")
    print(f"Valid score: {score}")

    if valid_loss < best_loss:
        best_loss = valid_loss
        torch.save(model.state_dict(), "nbme_bert_v2.pth")
```

Εικόνα 28: Κώδικας για την εκπαίδευση και αξιολόγηση του μοντέλου

Η ακρίβεια του παραπάνω μοντέλου, σύμφωνα με την μετρική F1 score που έχει επιλεγεί, είναι 0,794 ή αλλιώς 79,4%.

```
{ 'precision': 0.7644383810823101,
'recall': 0.8260919839273841, 'f1': 0.7940702456374347}
```

Εικόνα 29: Αξιολόγηση του μοντέλου

Ολοκληρώνοντας, όπως έχει ήδη αναφερθεί, η τελική μορφή που πρέπει να έχει το αποτέλεσμα του μοντέλου είναι η δημιουργία ενός συνόλου δεδομένων που περιέχει 2 στήλες, id και location. Με τη πρώτη να δηλώνει το αναγνωριστικό του χαρακτηριστικού (feature) που έχει γίνει annotation και τη δεύτερη, τη τοποθεσία ή τοποθεσίες του annotation.

	id	location
0	00016_000	696 724
1	00016_001	668 693
2	00016_002	203 217
3	00016_003	70 91
4	00016_004	222 232; 236 258

Εικόνα 30: Η τοποθεσία των αντίστοιχων προβλεπόμενων annotation

Μέσα από την εικόνα 30, εύκολα συμπεραίνουμε ότι για το αναγνωριστικό 00016_000 βρήκαμε ότι για το χαρακτηριστικό 000, στη σημείωση ασθενούς 00016 , γίνεται annotation στη θέση 696 μέχρι 724, για το id 00016_001 γίνεται στη θέση 668 έως 693 και ούτε καθεξής.

5.2 Ερμηνεία αποτελεσμάτων

Μια σημαντική μέτρηση αξιολόγησης στη μηχανική μάθηση είναι η μετρική F1. Συνδυάζοντας το precision και το recall , συνοψίζει την ικανότητα πρόβλεψης ενός μοντέλου. Η μετρική F1 έχει εύρος από 0 έως 1, με το 0 να δηλώνει το χειρότερο δυνατό αποτέλεσμα και το 1 να σημαίνει ένα άψογο αποτέλεσμα, δηλαδή το μοντέλο προβλέπει με ακρίβεια κάθε παρατήρηση.

Το μοντέλο που έχει αναπτυχθεί στην παρούσα εργασία, δίνει F1 score 0,794 ή αλλιώς 79,4%. Σύμφωνα με τη θεωρία, η μετρική F1 είναι καλή, καθώς είναι πολύ κοντά στο 0.8 ή 80%. Με άλλα λόγια, το μοντέλο μας προβλέπει σωστά τα annotations και τη θέση τους, το 79,4% των φορών.

Με άλλα λόγια, υψηλό σκορ στο F1 σημαίνει ότι πιθανότατα το μοντέλο παρουσιάζει υψηλό precision και recall. Εδώ έχουμε precision 0,764 ή καλύτερα 76,4% και recall 0,826 ή 82,6%. Η χαμηλότερη τιμή στο precision σημαίνει ότι, από τις περιπτώσεις που προσδιορίστηκαν ως θετικές, το μοντέλο δεν πέτυχε πολλές από αυτές σωστά.

Κεφάλαιο 6- Συμπεράσματα και προτάσεις

6.1 Σύνοψη αποτελεσμάτων έρευνας

Η παρούσα έρευνα επικεντρώνεται στην αξιοποίηση του προ-εκπαιδευμένου μοντέλου BERT στον ιατρικό κλάδο και πιο συγκεκριμένα σε δεδομένα ιατρικού κειμένου. Σκοπός της, ο αυτόματος προσδιορισμός ιατρικών εννοιών μέσα από ιστορικό ιατρικό και τα ιατρικά συμπτώματα των ασθενών. Απόρροια αυτού, η γρηγορότερη και πιο έγκυρη διάγνωση των ασθενειών. Είναι γνωστό ότι η διάγνωση σε πρώιμο στάδιο μιας νόσου είναι πολύ σημαντική καθώς οι πιθανότητες ίασης είναι αυξημένες.

Χρησιμοποιήσαμε ένα σύνολο δεδομένων που έγινε διαθέσιμο από το NBM και είχε ως στόχο την προώθηση της βιοϊατρικής NLP. Το σύνολο αυτό περιείχε 10 ιατρικές περιπτώσεις και κάθε ιατρική έννοια ή χαρακτηριστικό εμπίπτει σε μία από αυτές. Τα δεδομένα κειμένου που αναλύθηκαν, προέρχονται από την εξέταση των ιατρών για την απόκτηση ιατρικής άδειας και περιγράφουν σημαντικές πληροφορίες σχετικές με τον ασθενή, οι οποίες προκύπτουν από την αλληλεπίδραση τους με τον εκπαιδευόμενο ιατρό, και περιλαμβάνει τη φυσική εξέταση και τη συνέντευξη. Για το πρακτικό κομμάτι αυτής της εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python σε συνδυασμό με τη βιβλιοθήκη PyTorch. Τέλος, το μοντέλο BERT, αφού αποτελεί ανοικτό κώδικα, το αντλήσαμε από το Hugging Face.

Το μοντέλο BERT συζητήθηκε ευρέως στην επιστημονική κοινότητα αμέσως μετά την εμφάνιση του και χρησιμοποιείται πλέον σε όλα τα ζητήματα επεξεργασίας κειμένου. Συντέλεσε στην ταχεία ανάπτυξη της επεξεργασίας φυσικής γλώσσας και εδραίωσε την επικράτηση της χρήσης των προ-εκπαιδευμένων μοντέλων σε τεράστια σύνολα δεδομένων. Αυτό συνέβη, καθώς βασίζεται στην αρχιτεκτονική των Transformer με μηχανισμό προσοχής και στην αμφίδρομη μοντελοποίηση. Ωστόσο, η εφαρμογή της NLP στην ιατρική κοινότητα εξακολουθεί να αποτελεί πρόκληση λόγω της σπανιότητας και πολυπλοκότητας των δεδομένων κειμένου.

Τα ευρήματα της έρευνας ήταν αρκετά ικανοποιητικά, η μετρική F1 είχε score 79,4%, ενώ Recall και Precision ήταν 82,6% και 76,4% αντιστοίχως. Ασχοληθήκαμε μόνο με τη χρήση του προ-εκπαιδευμένου μοντέλου BERT καθώς όπως έχουμε αναφέρει έχει επιτύχει αποτελέσματα τελευταίας τεχνολογίας στις περισσότερες εργασίες της NLP, οπότε κρίθηκε ότι δεν ήταν αναγκαία η σύγκριση του με προηγούμενες μεθοδολογίες.

Εν κατακλείδι, ένα γενικό συμπέρασμα από το μοντέλο που δημιουργήσαμε είναι πως αν και φαίνεται να παρουσιάζει καλή απόδοση, τέτοια αυτοματοποιημένα συστήματα πρέπει να είναι εξαιρετικά ακριβή προκειμένου να εγκριθούν και να εφαρμοστούν στην πράξη. Σύμφωνα με όσα αναφέρθηκαν, η απόδοση του μοντέλου θα έπρεπε να ήταν ισοδύναμη ή και καλύτερη με αυτή του ανθρώπινου παράγοντα. Εξαιτίας αυτού, δεν κατατείνει στην απόδειξη της ερευνητικής υπόθεσης που θέσαμε στο Κεφάλαιο 2, πράγμα που σημαίνει ότι το προ-εκπαιδευμένο μοντέλο BERT που αναπτύχθηκε δεν μπορεί να χρησιμοποιηθεί για την αυτοματοποίηση της διαδικασίας μάθησης και αξιολόγησης των εκπαιδευμένων ιατρών.

6.2 Περιορισμοί και μελλοντική εργασία

Η επίτευξη της ανάλυσης των δεδομένων ιατρικού κειμένου σε αυτή τη διπλωματική έγινε εφικτή, καθώς το Εθνικό Συμβούλιο Ιατρικών Εξεταστών έκανε διαθέσιμα δεδομένα που προσομοιώνουν την εξέταση «USMLE Step 2 Clinical Skills» στους συμμετέχοντες του διαγωνισμού μέσα στη πλατφόρμα Kaggle και στη συνέχεια στο ευρύ κοινό. Τα δεδομένα αυτά είναι ανώνυμα και αποτρέπει τους κινδύνους για αναγνώριση του εξεταζόμενου ή εξαγωγή συμπερασμάτων σχετικά με την ατομική απόδοση. Παράλληλα, η χρήση αυτών των δεδομένων για ερευνητικούς σκοπούς διασφαλίζεται με τη διανομή τους μέσω συμφωνιών. Τα παραπάνω αναφέρονται διότι η μεγαλύτερη πρόκληση που έχει να αντιμετωπίσει η επεξεργασία φυσικής γλώσσας σε δεδομένα ιατρικού περιεχομένου είναι η ιδιωτικότητα των δεδομένων, καθώς κρίνεται αναγκαίο να λαμβάνονται μέτρα για τη διατήρηση του απορρήτου.

Στη διάρκεια αυτής της έρευνας, επικεντρωθήκαμε μόνο στην αξιοποίηση του προ-εκπαιδευμένου μοντέλου BERT. Σκοπός ήταν η εισαγωγή στις έννοιες και στις εφαρμογές της επεξεργασίας φυσικής γλώσσας με την χρήση του προ-εκπαιδευμένου μοντέλου BERT και η εξοικείωση του ερευνητή με τη χρήση του σε δεδομένα ιατρικού περιεχομένου. Από την άλλη, όπως έχει ήδη ειπωθεί, υπάρχει μια οικογένεια μοντέλων που προέρχονται από αυτό το μοντέλο, τα οποία αλλάζοντας βασικά σημεία στην αρχιτεκτονική τους πετυχαίνουν βελτίωση στην απόδοση συγκριτικά με το βασικό μοντέλο BERT.

Μια μελλοντική έρευνα, θα μπορούσε να αξιοποιήσει μερικά από τα μοντέλα που βασίζονται στο BERT, όπως το RoBERTa και DeBERTa. Στην ιστοσελίδα Kaggle υπάρχουν διαθέσιμοι κώδικες που έχουν χρησιμοποιήσει τα παραπάνω μοντέλα στα δεδομένα του διαγωνισμού και έχουν παρουσιάσει καλύτερη απόδοση, 88% και 86%

αντιστοίχως (*NBME / Deberta-Base Baseline [Inference]*, n.d.; *Roberta Strikes Back !*, n.d.).

Επιπρόσθετα, μια μελλοντική έρευνα θα μπορούσε να είναι η βελτίωση του μοντέλου που έχει αναπτυχθεί στην παρούσα έρευνα αλλά και των αντίστοιχων που είναι διαθέσιμα στο Kaggle. Πράγματι, για το μοντέλο που έχει αναπτυχθεί εδώ, η αλλαγή των παραμέτρων στο fine-tuning κατά τη διαδικασία της εκπαίδευσης, ενδέχεται να οδηγήσει σε σημαντική αύξηση της απόδοσης του μοντέλου, όπως αντίστοιχες αλλαγές μπορούν να πραγματοποιηθούν και κατά τη διαδικασία του tokenization.

Η απόδοση του μοντέλου επηρεάζεται αρνητικά από την ασάφεια της γλώσσας που χρησιμοποιείται στις σημειώσεις των ασθενών. Παράλληλα, λόγω της μη τυποποίησης των ιατρικών ακρωνύμιων και της μεγάλης ποικιλίας εκφράσεων για την επεξήγηση ενός ιατρικού φαινομένου, το λεξιλόγιο που έπρεπε να μάθει το μοντέλο ήταν αρκετά διευρυμένο. Οι (Lu et al., n.d.) απέδειξαν ότι η χρήση εκτός των βασικών μεθόδων προ-επεξεργασίας κειμένου, μπορεί να προκαλέσει μικρές έως σημαντικές αυξομειώσεις στην απόδοση του μοντέλου. Η μεγαλύτερη μείωση της απόδοσης γίνεται στην αφαίρεση της διαδικασίας «stopword» και «stemming». Απεναντίας, η χρήση μόνο πεζών γραμμάτων αυξάνει την απόδοση του μοντέλου, όπως και η αφαίρεση των παυλών από το «-OR-». Ωστόσο η αφαίρεση όλων των παυλών είχε ως αποτέλεσμα τη μείωση της απόδοσης. Μελλοντικά πειράματα μπορούν να επικεντρωθούν στη δημιουργία ενός συνόλου από λέξεις για τη διαδικασία «stopword» για συγκεκριμένους τομείς, και ειδικότερα για την ιατρική.

6.3 Προτάσεις

Η δημόσια δημοσίευση των δεδομένων από το NBME που χρησιμοποιήθηκαν και σε αυτή τη διπλωματική, σηματοδότησε την έναρξη της ανάπτυξης λύσεων για εργασίες σχετικά με τις ιατρικές σημειώσεις, βελτιώνοντας την εκπαιδευτική αξιολόγηση στον τομέα της ιατρικής με τη χρήση της τεχνολογίας. Συνοψίζοντας, η μηχανική μάθηση σε συνδυασμό με την επεξεργασία φυσικής γλώσσας μπορούν να συμβάλουν στην επίτευξη αυτοματοποιημένων διαδικασιών.

Σύμφωνα με το μοντέλο που αναπτύχθηκε, προβλέπει σωστά τις σημαντικές ιατρικές έννοιες από το κείμενο με απόδοση F1 79,4%. Η απόδοση αυτή, δεν είναι αρκετή έτσι ώστε να αποδεχτούμε την ερευνητική υπόθεση που είχε τεθεί ότι το μοντέλο BERT μπορεί να αξιοποιηθεί για την παραπάνω εφαρμογή. Μολονότι η απόδοση είναι

ικανοποιητική, επειδή η εφαρμογή αφορά ιατρικούς σκοπούς η βελτίωση της κρίνεται αναγκαία. Η χρήση διαφορετικών μοντέλων, όπως αναφερθήκαμε, δίνει καλύτερα αποτελέσματα συγκριτικά με το μοντέλο BERT.

Η ανάγκη της επιτάχυνσης της χρονοβόρας διαδικασίας μάθησης και αξιολόγησης των ασκούμενων ιατρών, η οποία προς το παρών εξαρτάται από τη συμβολή έμπειρων ιατρών, έχει ξεκινήσει τη διαδικασία της μοντελοποίησης αλλά μόνο στην αγγλική γλώσσα. Τα ηλεκτρονικά ιατρικά δεδομένα σε άλλες γλώσσες δεν είναι διαθέσιμα σε αντίστοιχες ποσότητες με αποτέλεσμα όποιες έρευνες έχουν πραγματοποιηθεί να αφορούν μόνο κείμενο στα αγγλικά. Υπάρχουν εξαιρέσεις εφαρμογών της επεξεργασίας φυσικής γλώσσας, που εντοπίζονται και μοντέλα που χρησιμοποιούν δεδομένα σε διάφορες γλώσσες όπως κινέζικα, αραβικά κ.α.

Αντίστοιχη έρευνα, θα μπορούσε να πραγματοποιηθεί και στην ελληνική γλώσσα. Υπάρχει διαθέσιμη η αντίστοιχη ελληνική έκδοση του προ-εκπαιδευμένου γλωσσικού μοντέλου BERT, το οποίο έχει εκπαιδευτεί στο ελληνικό μέρος της Wikipedia, στο ελληνικό τμήμα του Σώματος Πρακτικών του Ευρωπαϊκού Κοινοβουλίου και στο ελληνικό μέρος του Oscar (Open Super-large Crawled Aggregated corpus). Αν και υπάρχουν ελάχιστα διαθέσιμα σύνολο δεδομένων κειμένου στα ελληνικά και άρα και σύνολα δεδομένων ιατρικού περιεχομένου, οι (Koutsikakis et al., 2020) έχουν δείξει ότι το μοντέλο αυτό επιτυγχάνει κορυφαίες αποδόσεις, κυρίως σε εφαρμογές αναγνώρισης ονομαστικών οντοτήτων και συμπέρασμα φυσικής γλώσσας.

Βιβλιογραφία

3.3. Metrics and scoring: Quantifying the quality of predictions. (n.d.). Scikit-Learn. Retrieved November 22, 2022, from https://scikit-learn/stable/modules/model_evaluation.html

Alfarghaly, O., Khaled, R., Elkorany, A., Helal, M., & Fahmy, A. (2021). Automated radiology report generation using conditioned transformers. *Informatics in Medicine Unlocked*, *24*, 100557. https://doi.org/10.1016/j.imu.2021.100557

BCEWithLogitsLoss—PyTorch 1.13 documentation. (n.d.). Retrieved November 21, 2022, from https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2015). *Distilling Knowledge from Deep Networks with Applications to Healthcare Domain* (arXiv:1512.03542). arXiv. http://arxiv.org/abs/1512.03542

Chen, L., Varoquaux, G., & Suchanek, F. M. (2021). A Lightweight Neural Model for Biomedical Entity Linking. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(14), Article 14. https://doi.org/10.1609/aaai.v35i14.17499

Cho, H., & Lee, H. (2019). Biomedical named entity recognition using deep neural networks with contextual information. *BMC Bioinformatics*, *20*(1), 735. https://doi.org/10.1186/s12859-019-3321-4

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Faris, H., Faris, M., Habib, M., & Alomari, A. (2022). Automatic symptoms identification from a massive volume of unstructured medical consultations using deep neural and BERT models. *Heliyon*, *8*(6), e09683. https://doi.org/10.1016/j.heliyon.2022.e09683

Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., & Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. *Journal of Biomedical Informatics*, *46*(3), 541–562. https://doi.org/10.1016/j.jbi.2012.12.003

Han, X., Wang, Y.-T., Feng, J.-L., Deng, C., Chen, Z.-H., Huang, Y.-A., Su, H., Hu, L., & Hu, P.-W. (2023). A survey of transformer-based multimodal pre-trained modals. *Neurocomputing*, *515*, 89–106. https://doi.org/10.1016/j.neucom.2022.09.136

Häyrinen, K., Saranto, K., & Nykänen, P. (2008). Definition, structure, content, use and impacts of electronic health records: A review of the research literature. *International Journal of Medical Informatics*, *77*(5), 291–304. https://doi.org/10.1016/j.ijmedinf.2007.09.001

Huang, J., Osorio, C., & Sy, L. W. (2019). An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes. *Computer Methods and Programs in Biomedicine*, *177*, 141–153. https://doi.org/10.1016/j.cmpb.2019.05.024

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2022). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-022-13428-4

Koutsikakis, J., Chalkidis, I., Malakasiotis, P., & Androutsopoulos, I. (2020). GREEK-BERT: The Greeks visiting Sesame Street. *11th Hellenic Conference on Artificial Intelligence*, 110–117. https://doi.org/10.1145/3411408.3411440

Lamproudis, A., Henriksson, A., & Dalianis, H. (n.d.). *Evaluating Pretraining Strategies for Clinical BERT Models*. 7.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Li, I., Pan, J., Goldwasser, J., Verma, N., Wong, W. P., Nuzumlalı, M. Y., Rosand, B., Li, Y., Zhang, M., Chang, D., Taylor, R. A., Krumholz, H. M., & Radev, D. (2022). Neural Natural Language Processing for unstructured data in electronic health records: A review. *Computer Science Review*, *46*, 100511. https://doi.org/10.1016/j.cosrev.2022.100511

Liu, J., Capurro, D., Nguyen, A., & Verspoor, K. (2022). "Note Bloat" impacts deep learning-based NLP models for clinical prediction tasks. *Journal of Biomedical Informatics*, *133*, 104149. https://doi.org/10.1016/j.jbi.2022.104149

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (arXiv:1907.11692). arXiv. https://doi.org/10.48550/arXiv.1907.11692

Locke, S., Bashall, A., Al-Adely, S., Moore, J., Wilson, A., & Kitchen, G. B. (2021). Natural language processing in medicine: A review. *Trends in Anaesthesia and Critical Care*, *38*, 4–9. https://doi.org/10.1016/j.tacc.2021.02.007

Loshchilov, I., & Hutter, F. (2019). *Decoupled Weight Decay Regularization* (arXiv:1711.05101). arXiv. https://doi.org/10.48550/arXiv.1711.05101

Lu, S. Y. F., Balaji, S., Shenoy, N., Bakhtawar, M., Chan, J. H., & Thanapattheerakul, T. (n.d.). *The Impact of Preprocessing on the Automated Scoring of the USMLE Step 2 Clinical Skills Exam*. 2.

Lyu, X., Hueser, M., Hyland, S. L., Zerveas, G., & Raetsch, G. (2018). *Improving Clinical Predictions through Unsupervised Time Series Representation Learning* (arXiv:1812.00490). arXiv. http://arxiv.org/abs/1812.00490

Mahajan, D., Liang, J. J., & Tsou, C.-H. (2021). *Extracting Daily Dosage from Medication Instructions in EHRs: An Automated Approach and Lessons Learned* (arXiv:2005.10899). arXiv. http://arxiv.org/abs/2005.10899

Marafino, B. J., Davies, J. M., Bardach, N. S., Dean, M. L., & Dudley, R. A. (2014). N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *Journal of the American Medical Informatics Association*, *21*(5), 871–875. https://doi.org/10.1136/amiajnl-2014-002694

Martinez, I., Viles, E., & G. Olaizola, I. (2021). Data Science Methodologies: Current Challenges and Future Approaches. *Big Data Research*, *24*, 100183. https://doi.org/10.1016/j.bdr.2020.100183

Mondal, I. (2020). *BERTChem-DDI: Improved Drug-Drug Interaction Prediction from text using Chemical Structure Information* (arXiv:2012.11599). arXiv. http://arxiv.org/abs/2012.11599

Mu, Y., Tizhoosh, H. R., Tayebi, R. M., Ross, C., Sur, M., Leber, B., & Campbell, C. J. V. (2021). A BERT model generates diagnostically relevant semantic embeddings from pathology synopses with active learning. *Communications Medicine*, *1*(1), Article 1. https://doi.org/10.1038/s43856-021-00008-0

Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., & Eisenstein, J. (2018). *Explainable Prediction of Medical Codes from Clinical Text* (arXiv:1802.05695). arXiv. http://arxiv.org/abs/1802.05695

NBME / Deberta-base baseline [inference]. (n.d.). Retrieved November 25, 2022, from https://kaggle.com/code/yasufuminakama/nbme-deberta-base-baseline-inference

Papaioannou, J.-M., Grundmann, P., van Aken, B., Samaras, A., Kyparissidis, I., Giannakoulas, G., Gers, F., & Löser, A. (2022). *Cross-Lingual Knowledge Transfer for Clinical Phenotyping* (arXiv:2208.01912). arXiv. https://doi.org/10.48550/arXiv.2208.01912

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, *63*(10), 1872–1897. https://doi.org/10.1007/s11431-020-1647-3

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (n.d.). *Improving Language Understanding by Generative Pre-Training*. 12.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (n.d.). *Language Models are Unsupervised Multitask Learners*. 24.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Raina, V., & Krishnamurthy, S. (2022). Natural Language Processing. In V. Raina & S. Krishnamurthy (Eds.), *Building an Effective Data Science Practice: A Framework to Bootstrap and Manage a Successful Data Science Practice* (pp. 63–73). Apress. https://doi.org/10.1007/978-1-4842-7419-4_6

Roberta Strikes Back ! (n.d.). Retrieved November 25, 2022, from https://kaggle.com/code/theoviel/roberta-strikes-back

Rosett, C. M., & Hagerty, A. (2021). Introducing Machine Learning. In C. M. Rosett & A. Hagerty (Eds.), *Introducing HR Analytics with Machine Learning: Empowering Practitioners, Psychologists, and Organizations* (pp. 107–127). Springer International Publishing. https://doi.org/10.1007/978-3-030-67626-1_8

Salman, S., & Liu, X. (2019). *Overfitting Mechanism and Avoidance in Deep Neural Networks* (arXiv:1901.06566). arXiv. http://arxiv.org/abs/1901.06566

Salt, J., Harik, P., & Barone, M. A. (2019). Leveraging Natural Language Processing: Toward Computer-Assisted Scoring of Patient Notes in the USMLE Step 2 Clinical Skills Exam. *Academic Medicine*, *94*(3), 314–316. https://doi.org/10.1097/ACM.0000000000002558

Schäfer, A., Blach, N., Rausch, O., Warm, M., & Krüger, N. (2020). *Towards Automated Anamnesis Summarization: BERT-based Models for Symptom Extraction* (arXiv:2011.01696). arXiv. http://arxiv.org/abs/2011.01696

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

ShafieiBavani, E., Jimeno Yepes, A., Zhong, X., & Martinez Iraola, D. (2020). Global Locality in Biomedical Relation and Event Extraction. *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, 195–204. https://doi.org/10.18653/v1/2020.bionlp-1.21

Sugimoto, K., Takeda, T., Oh, J.-H., Wada, S., Konishi, S., Yamahata, A., Manabe, S., Tomiyama, N., Matsunaga, T., Nakanishi, K., & Matsumura, Y. (2021). Extracting clinical terms from radiology reports with deep learning. *Journal of Biomedical Informatics*, *116*, 103729. https://doi.org/10.1016/j.jbi.2021.103729

Swygert, K., Margolis, M., King, A., Siftar, T., Clyman, S., Hawkins, R., & Clauser, B. (2003). Evaluation of an Automated Procedure for Scoring Patient Notes as Part of a Clinical Skills Examination: *Academic Medicine*, *78*(Supplement), S75–S77. https://doi.org/10.1097/00001888-200310001-00024

Valmianski, I., Goodwin, C., Finn, I. M., Khan, N., & Zisook, D. S. (2019). *Evaluating robustness of language models for chief complaint extraction from patient-generated text* (arXiv:1911.06915). arXiv. https://doi.org/10.48550/arXiv.1911.06915

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. http://arxiv.org/abs/1706.03762

Weng, W.-H., Chung, Y.-A., & Szolovits, P. (2019). Unsupervised Clinical Language Translation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3121–3131. https://doi.org/10.1145/3292500.3330710

Xia, Y., Li, F., Liu, Q., Jin, L., Zhang, Z., Sun, X., & Shao, L. (2022). ReasonFuse: Reason Path Driven and Global-Local Fusion Network for Numerical Table-Text Question Answering. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2022.09.046

Xu, S., Zhang, C., & Hong, D. (2022). BERT-based NLP techniques for classification and severity modeling in basic warranty data study. *Insurance: Mathematics and Economics*, *107*, 57–67. https://doi.org/10.1016/j.insmatheco.2022.07.013

Yaneva, V., Mee, J., Ha, L. A., Harik, P., Jodoin, M., & Mechaber, A. (2022). *The USMLE® Step 2 clinical skills patient note corpus*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.208

Yang, Z., Dehmer, M., Yli-Harja, O., & Emmert-Streib, F. (2020). Combining deep learning with token selection for patient phenotyping from electronic health records. *Scientific Reports*, *10*(1), Article 1. https://doi.org/10.1038/s41598-020-58178-1

Yoon, W., Lee, J., Kim, D., Jeong, M., & Kang, J. (2020). Pre-trained Language Model for Biomedical Question Answering. In P. Cellier & K. Driessens (Eds.), *Machine Learning and Knowledge Discovery in Databases* (pp. 727–740). Springer International Publishing. https://doi.org/10.1007/978-3-030-43887-6_64

Zhang, H., Hu, D., Duan, H., Li, S., Wu, N., & Lu, X. (2021). A novel deep learning approach to extract Chinese clinical entities for lung cancer screening and staging. *BMC Medical Informatics and Decision Making*, *21*(2), 214. https://doi.org/10.1186/s12911-021-01575-x

Zhang, Y., Ding, D. Y., Qian, T., Manning, C. D., & Langlotz, C. P. (2018). Learning to Summarize Radiology Findings. *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, 204–213. https://doi.org/10.18653/v1/W18-5623

Zhang, Y., Lin, H., Yang, Z., Wang, J., Zhang, S., Sun, Y., & Yang, L. (2018). A hybrid model based on neural networks for biomedical relation extraction. *Journal of Biomedical Informatics*, *81*, 83–92. https://doi.org/10.1016/j.jbi.2018.03.011

Zimmermann, J. L. (2021). Data Competitions: Crowdsourcing with Data Science Platforms. In M. Einhorn, M. Löffler, E. de Bellis, A. Herrmann, & P. Burghartz (Eds.), *The Machine Age of Customer Insight* (pp. 183–197). Emerald Publishing Limited. https://doi.org/10.1108/978-1-83909-694-520211017

Παράρτημα Ι- Ακρώνυμα

Ακρώνυμο	*Επεξήγηση*
BERT	Bidirectional Encoder Representations from Transformers
NBME	National Board of Medical Examiners
NLP	Natural Language Processing
EHR	Electronic Health Records
QA	Question Answering
PTM	Pre-Trained Model
T5	Text-To-Text Transfer Transformer
GPT	Generative Pre-Trained Transformers
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
GLUE	General Language Understanding
SQuAD	The Stanford Question Answering Dataset
SWAG	Situation With Adversarial Generations
NER	Named Entity Recognition
RE	Relation Extraction
CRISP-DM	Cross-Industry Standard Process for Data Mining

Παράρτημα ΙΙ- Κώδικας Ανάλυσης Δεδομένων και ανάπτυξης μοντέλου BERT

Περιγραφική στατιστική

Αρχικά προσθέσαμε όλες τις απαραίτητες βιβλιοθήκες και φορτώσαμε τα δεδομένα μας σε Pandas data frame.

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under
the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as ou
tput when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the cu
rrent session
```

```
import os
import spacy
import warnings
import wordcloud
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
import plotly.graph_objects as go
```

```
train = pd.read_csv("../input/nbme-score-clinical-patient-notes/train.csv")
test = pd.read_csv("../input/nbme-score-clinical-patient-notes/test.csv")
features = pd.read_csv("../input/nbme-score-clinical-patient-notes/features.csv")
patient_notes = pd.read_csv("../input/nbme-score-clinical-patient-notes/patient_notes.csv")
submission = pd.read_csv("../input/nbme-score-clinical-patient-notes/sample_submission.csv")
```

Στη συνέχεια, προχωρήσαμε στην περιγραφή των συνόλων δεδομένων μας, δημιουργώντας τα κατάλληλα γραφήματα έτσι ώστε να κατανοήσουμε καλύτερα το περιεχόμενο τους. Πιο συγκεκριμένα, στο Κεφάλαιο 4 για κάθε data set δημιουργήσαμε

ένα γράφημα Word Cloud και ένα ραβδόγραμμα για την κατανομή του εκάστοτε data set ανά case.

Train data

```
print(f'\033[92mNumber of rows in train data: {train.shape[0]}')
print(f'\033[94mNumber of columns in train data: {train.shape[1]}')
print(f'\033[91mNumber of values in train data: {train.count().sum()}')
print(f'\033[91mNumber missing values in train data: {sum(train.isna().sum())}')
train.head()
```

Test data

```
print(f'\033[92mNumber of rows in test data: {test.shape[0]}')
print(f'\033[94mNumber of columns in test data: {test.shape[1]}')
print(f'\033[91mNumber of values in train data: {test.count().sum()}')
print(f'\033[91mNo of rows with missing values  in test data: {sum(test.isna().sum())}')
test.head()
```

Patient notes data

```
print(f'\033[92mNumber of rows in test data: {patient_notes.shape[0]}')
print(f'\033[94mNumber of columns in test data: {patient_notes.shape[1]}')
print(f'\033[91mNumber of values in train data: {patient_notes.count().sum()}')
patient_notes.head(15)
```

```
RANDOM_IDX = 16
warnings.filterwarnings('ignore')
```

```
print(patient_notes["pn_history"].iloc[RANDOM_IDX])
```

```
print("Minimun length of Patient History -",np.min(all_notes_len))
print("Maximum lenght of Patient History -", np.max(all_notes_len))
```

```
notes_counts = patient_notes.groupby("case_num").count()
fig = px.bar(data_frame =notes_counts,
             x = notes_counts.index,
             y = 'pn_num' ,
             color = "pn_num",
             color_continuous_scale="blues")
fig.update_layout(title = {
        'text': 'Distribution of patient notes for each case',
        'y':0.95,
        'x':0.48,
        'xanchor': 'center',
        'yanchor': 'top'} ,
                  xaxis = dict(
        tickmode = 'array',
        tickvals = [0, 1,2, 3, 4,5, 6,7,8,9],
        ticktext = ['Case 0', 'Case 1', 'Case 2', 'Case 3', 'Case 4', 'Case 5', 'Case 6', 'Cas
e 7', 'Case 8', 'Case 9']),
                  template = "plotly_white")
fig.show()
```

```
wordcloud_notes = wordcloud.WordCloud(stopwords=wordcloud.STOPWORDS, max_font_size=120, max_wo
rds=5000,
                      width = 600, height = 400,
                      background_color='white',colormap='PuBu').generate(" ".join(all_notes))
fig, ax = plt.subplots(figsize=(14,10))
ax.imshow(wordcloud_notes, interpolation='bilinear')
ax.set_axis_off()
plt.imshow(wordcloud_notes);
```

Feature data

```
print(f'\033[92mNumber of rows in test data: {features.shape[0]}')
print(f'\033[94mNumber of columns in test data: {features.shape[1]}')
print(f'\033[91mNumber of values in train data: {features.count().sum()}')
features.head()
```

```
print("Minimun length of Feature text -",np.min(all_feat_len))
print("Maximum lenght of Feature text -", np.max(all_feat_len))
```

```
feature_counts = features.groupby("case_num").count()
fig = px.bar(data_frame =feature_counts,
             x = feature_counts.index,
             y = 'feature_num' ,
             color = "feature_num",
             color_continuous_scale="blues")
fig.update_layout(title = {
        'text': 'Distribution of Features for each case',
        'y':0.95,
        'x':0.48,
        'xanchor': 'center',
        'yanchor': 'top'} ,
                  xaxis = dict(
        tickmode = 'array',
        tickvals = [0, 1,2, 3, 4,5, 6,7,8,9],
        ticktext = ['Case 0', 'Case 1', 'Case 2', 'Case 3', 'Case 4', 'Case 5', 'Case 6', 'Cas
e 7', 'Case 8', 'Case 9']),
                  template = "plotly_white")
fig.show()
```

```
wordcloud_feat = wordcloud.WordCloud(stopwords=wordcloud.STOPWORDS, max_font_size=120, max_wor
ds=5000,
                     width = 600, height = 400,
                     background_color='white',colormap='PuBu').generate(" ".join(all_feat))
fig, ax = plt.subplots(figsize=(14,10))
ax.imshow(wordcloud_feat, interpolation='bilinear')
ax.set_axis_off()
plt.imshow(wordcloud_feat);
```

Annotation analysis

```
print("Minimun length of Feature text -",np.min(annot_lengths))
print("Maximum lenght of Feature text -", np.max(annot_lengths))
```

```
wordcloud_annot = wordcloud.WordCloud(stopwords=wordcloud.STOPWORDS, max_font_size=120, max_wo
rds=5000,
                     width = 600, height = 400,
                     background_color='white',colormap='PuBu').generate(" ".join(all_annot_wo
rds))
fig, ax = plt.subplots(figsize=(14,10))
ax.imshow(wordcloud_annot, interpolation='bilinear')
ax.set_axis_off()
plt.imshow(wordcloud_annot);
```

Patient analysis

```
print("Unique Patient Count in train data : ",len(train["pn_num"].value_counts()))
```

```
PATIENT_IDX = 16
patient_df = train[train["pn_num"] == PATIENT_IDX]
patient_df
```

```
print(f"\033[94mPatient Notes - ")
print(f'\033[94m',patient_notes[patient_notes["pn_num"] == PATIENT_IDX]["pn_history"].iloc[0])
print("------------")
print(f'\033[92mAnnotations:')
for i in range(len(patient_df)):
    print(f'\033[92m',patient_df["annotation"].iloc[i])
```

Cases genre

Ενώσαμε τα data sets με στόχο να εντοπίσουμε τον ιατρικό τομέα που εμπίπτει η κάθε ιατρική περίπτωση, που περιγράφει το σενάριο, τα συμπτώματα και τα παράπονα του τυποποιημένου ασθενή.

```
train = train.merge(features, on=['feature_num', 'case_num'], how='left')
train = train.merge(patient_notes, on=['pn_num', 'case_num'], how='left')
display(train.head())
```

```
train_case_num_group = train.groupby(train['case_num'])
train_feature_num_group = train.groupby(train['feature_num'])
```

```
for case_num, each_case in train_case_num_group:
    print('case_num:', case_num)
    display(each_case.head(3))
```

Μοντέλο BERT

Για το επόμενο μέρος της ανάλυσης, δημιουργήσαμε ένα δεύτερο notebook που ακολουθήσαμε παρόμοια διαδικασία με την περιγραφική στατιστική. Αναλυτικότερα, η δομή που ακολουθήσαμε είναι: εισαγωγή των απαραίτητων βιβλιοθηκών, δημιουργία βοηθητικών συναρτήσεων, εφαρμογή του Tokenizer, εκπαίδευση του μοντέλου, επιλογή βέλτιστου μοντέλου και συμπεράσματα.

```
from ast import literal_eval
from itertools import chain

import numpy as np
import pandas as pd
import torch
import torch.nn as nn
from sklearn.metrics import precision_recall_fscore_support
from sklearn.model_selection import train_test_split
from torch import optim
from torch.utils.data import DataLoader
from torch.utils.data import Dataset
from tqdm.notebook import tqdm
from transformers import AutoModel, AutoTokenizer
from sklearn.model_selection import StratifiedKFold
```

```
ROOT = "../input/nbme-score-clinical-patient-notes"
```

```
def create_train_df(debug = False):
    feats = pd.read_csv(f"{ROOT}/features.csv")
    notes = pd.read_csv(f"{ROOT}/patient_notes.csv")
    train = pd.read_csv(f"{ROOT}/train.csv")

    train["annotation_list"] = [literal_eval(x) for x in train["annotation"]]
    train["location_list"] = [literal_eval(x) for x in train["location"]]
    merged = train.merge(notes, how = "left")
    merged = merged.merge(feats, how = "left")


    def process_feature_text(text):
        return text.replace("-OR-", ";-").replace("-", " ")
    merged["feature_text"] = [process_feature_text(x) for x in merged["feature_text"]]

    merged["feature_text"] = merged["feature_text"].apply(lambda x: x.lower())
    merged["pn_history"] = merged["pn_history"].apply(lambda x: x.lower())


    print(merged.shape)

    return merged

df = create_train_df()
```

```
first = df.loc[0]
example = {
    "feature_text": first.feature_text,
    "pn_history": first.pn_history,
    "location_list": first.location_list,
    "annotation_list": first.annotation_list
}
for key in example.keys():
    print(key)
    print(example[key])
    print("=" * 100)
```

Βοηθητικές συναρτήσεις

Για το data set:

```
class CustomDataset(Dataset):
    def __init__(self, data, tokenizer, config):
        self.data = data
        self.tokenizer = tokenizer
        self.config = config

    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        data = self.data.iloc[idx]
        tokens = tokenize_and_add_labels(self.tokenizer, data, self.config)

        input_ids = np.array(tokens["input_ids"])
        attention_mask = np.array(tokens["attention_mask"])
        token_type_ids = np.array(tokens["token_type_ids"])

        labels = np.array(tokens["labels"])
        offset_mapping = np.array(tokens['offset_mapping'])
        sequence_ids = np.array(tokens['sequence_ids']).astype("float16")

        return input_ids, attention_mask, token_type_ids, labels, offset_mapping, sequence_ids
```

Για το μοντέλο:

```
class CustomModel(nn.Module):
    def __init__(self, config):
        super().__init__()
        self.bert = AutoModel.from_pretrained(config['model_name'])  # BERT model
        self.dropout = nn.Dropout(p=config['dropout'])
        self.config = config
        self.fc1 = nn.Linear(768, 512)
        self.fc2 = nn.Linear(512, 512)
        self.fc3 = nn.Linear(512, 1)

    def forward(self, input_ids, attention_mask, token_type_ids):
        outputs = self.bert(input_ids=input_ids, attention_mask=attention_mask, token_type_ids
=token_type_ids)
        logits = self.fc1(outputs[0])
        logits = self.fc2(self.dropout(logits))
        logits = self.fc3(self.dropout(logits)).squeeze(-1)
        return logits
```

Για τις υπερ-παραμέτρους:

```
hyperparameters = {
    "max_length": 416,
    "padding": "max_length",
    "return_offsets_mapping": True,
    "truncation": "only_second",
    "model_name": "../input/huggingface-bert/bert-base-uncased",
    "dropout": 0.2,
    "lr": 1e-5,
    "test_size": 0.2,
    "seed": 1268,
    "batch_size": 8
}
```

Για την προετοιμασία του data set:

```
train_df = create_train_df()

X_train, X_test = train_test_split(train_df, test_size=hyperparameters['test_size'],
                                   random_state=hyperparameters['seed'])


print("Train size", len(X_train))
print("Test Size", len(X_test))
```

Tokenizer

```
tokenizer = AutoTokenizer.from_pretrained(hyperparameters['model_name'])

training_data = CustomDataset(X_train, tokenizer, hyperparameters)
train_dataloader = DataLoader(training_data, batch_size=hyperparameters['batch_size'], shuffle
=True)

test_data = CustomDataset(X_test, tokenizer, hyperparameters)
test_dataloader = DataLoader(test_data, batch_size=hyperparameters['batch_size'], shuffle=Fals
e)
```

```
def loc_list_to_ints(loc_list):
    to_return = []
    for loc_str in loc_list:
        loc_strs = loc_str.split(";")
        for loc in loc_strs:
            start, end = loc.split()
            to_return.append((int(start), int(end)))
    return to_return


def tokenize_and_add_labels(tokenizer, data, config):
    out = tokenizer(
        data["feature_text"],
        data["pn_history"],
        truncation=config['truncation'],
        max_length=config['max_length'],
        padding=config['padding'],
        return_offsets_mapping=config['return_offsets_mapping']
    )
    labels = [0.0] * len(out["input_ids"])
    out["location_int"] = loc_list_to_ints(data["location_list"])
    out["sequence_ids"] = out.sequence_ids()

    for idx, (seq_id, offsets) in enumerate(zip(out["sequence_ids"], out["offset_mapping"])):
        if not seq_id or seq_id == 0:
            labels[idx] = -1
            continue

        token_start, token_end = offsets
        for feature_start, feature_end in out["location_int"]:
            if token_start >= feature_start and token_end <= feature_end:
                labels[idx] = 1.0
                break

    out["labels"] = labels

    return out
```

```
out = tokenize_and_add_labels(tokenizer, example, hyperparameters )
for key in out.keys():
    print(key)
    print(out[key])
    print("=" * 100)
```

Παράδειγμα του tokenizer

```
i=0
while i<16:
 example=tokenizer.decode(out.input_ids[i])
 print(example)
 i=i+1
```

```
i=0
while i<16:
 example=tokenizer.decode(out.input_ids[i])
 print(example)
 i=i+1
```

Training

```
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

model = CustomModel(hyperparameters).to(DEVICE)

criterion = torch.nn.BCEWithLogitsLoss(reduction = "none")
optimizer = optim.AdamW(model.parameters(), lr=hyperparameters['lr'])
```

```
from sklearn.metrics import accuracy_score

def get_location_predictions(preds, offset_mapping, sequence_ids, test=False):
    all_predictions = []
    for pred, offsets, seq_ids in zip(preds, offset_mapping, sequence_ids):
        pred = 1 / (1 + np.exp(-pred))
        start_idx = None
        end_idx = None
        current_preds = []
        for pred, offset, seq_id in zip(pred, offsets, seq_ids):
            if seq_id is None or seq_id == 0:
                continue

            if pred > 0.5:
                if start_idx is None:
                    start_idx = offset[0]
                end_idx = offset[1]
            elif start_idx is not None:
                if test:
                    current_preds.append(f"{start_idx} {end_idx}")
                else:
                    current_preds.append((start_idx, end_idx))
                start_idx = None
        if test:
            all_predictions.append("; ".join(current_preds))
        else:
            all_predictions.append(current_preds)

    return all_predictions
```

```
def calculate_char_cv(predictions, offset_mapping, sequence_ids, labels):
    all_labels = []
    all_preds = []
    for preds, offsets, seq_ids, labels in zip(predictions, offset_mapping, sequence_ids, labe
ls):

        num_chars = max(list(chain(*offsets)))
        char_labels = np.zeros(num_chars)

        for o, s_id, label in zip(offsets, seq_ids, labels):
            if s_id is None or s_id == 0:
                continue
            if int(label) == 1:
                char_labels[o[0]:o[1]] = 1

        char_preds = np.zeros(num_chars)

        for start_idx, end_idx in preds:
            char_preds[start_idx:end_idx] = 1

        all_labels.extend(char_labels)
        all_preds.extend(char_preds)

    results = precision_recall_fscore_support(all_labels, all_preds, average="binary", labels=
np.unique(all_preds))
    accuracy = accuracy_score(all_labels, all_preds)


    return {
        "Accuracy": accuracy,
        "precision": results[0],
        "recall": results[1],
        "f1": results[2]
    }
```

Συνάρτηση training:

```
def train_model(model, dataloader, optimizer, criterion):
        model.train()
        train_loss = []

        for batch in tqdm(dataloader):
            optimizer.zero_grad()
            input_ids = batch[0].to(DEVICE)
            attention_mask = batch[1].to(DEVICE)
            token_type_ids = batch[2].to(DEVICE)
            labels = batch[3].to(DEVICE)

            logits = model(input_ids, attention_mask, token_type_ids)
            loss = criterion(logits, labels)
            # since, we have
            loss = torch.masked_select(loss, labels > -1.0).mean()
            train_loss.append(loss.item() * input_ids.size(0))
            loss.backward()
            # clip the the gradients to 1.0. It helps in preventing the exploding gradient probl
em
            # it's also improve f1 accuracy slightly
            nn.utils.clip_grad_norm_(model.parameters(), 1.0)
            optimizer.step()

        return sum(train_loss)/len(train_loss)
```

Συνάρτηση Evaluation:

```
def eval_model(model, dataloader, criterion):
        model.eval()
        valid_loss = []
        preds = []
        offsets = []
        seq_ids = []
        valid_labels = []

        for batch in tqdm(dataloader):
            input_ids = batch[0].to(DEVICE)
            attention_mask = batch[1].to(DEVICE)
            token_type_ids = batch[2].to(DEVICE)
            labels = batch[3].to(DEVICE)
            offset_mapping = batch[4]
            sequence_ids = batch[5]

            logits = model(input_ids, attention_mask, token_type_ids)
            loss = criterion(logits, labels)
            loss = torch.masked_select(loss, labels > -1.0).mean()
            valid_loss.append(loss.item() * input_ids.size(0))

            preds.append(logits.detach().cpu().numpy())
            offsets.append(offset_mapping.numpy())
            seq_ids.append(sequence_ids.numpy())
            valid_labels.append(labels.detach().cpu().numpy())

        preds = np.concatenate(preds, axis=0)
        offsets = np.concatenate(offsets, axis=0)
        seq_ids = np.concatenate(seq_ids, axis=0)
        valid_labels = np.concatenate(valid_labels, axis=0)
        location_preds = get_location_predictions(preds, offsets, seq_ids, test=False)
        score = calculate_char_cv(location_preds, offsets, seq_ids, valid_labels)

        return sum(valid_loss)/len(valid_loss), score
```

Επιλογή βέλτιστου μοντέλου:

```
import time

train_loss_data, valid_loss_data = [], []
score_data_list = []
valid_loss_min = np.Inf
since = time.time()
epochs = 3
```

```
best_loss = np.inf

for i in range(epochs):
    print("Epoch: {}/{}".format(i + 1, epochs))
    # first train model
    train_loss = train_model(model, train_dataloader, optimizer, criterion)
    train_loss_data.append(train_loss)
    print(f"Train loss: {train_loss}")
    # evaluate model
    valid_loss, score = eval_model(model, test_dataloader, criterion)
    valid_loss_data.append(valid_loss)
    score_data_list.append(score)
    print(f"Valid loss: {valid_loss}")
    print(f"Valid score: {score}")

    if valid_loss < best_loss:
        best_loss = valid_loss
        torch.save(model.state_dict(), "nbme_bert_v2.pth")


time_elapsed = time.time() - since
print('Training completed in {:.0f}m {:.0f}s'.format(
    time_elapsed // 60, time_elapsed % 60))
```

Συμπεράσματα:

```
model.load_state_dict(torch.load("nbme_bert_v2.pth", map_location = DEVICE))
```

```
def create_test_df():
    feats = pd.read_csv(f"{ROOT}/features.csv")
    notes = pd.read_csv(f"{ROOT}/patient_notes.csv")
    test = pd.read_csv(f"{ROOT}/test.csv")

    merged = test.merge(notes, how = "left")
    merged = merged.merge(feats, how = "left")

    def process_feature_text(text):
        return text.replace("-OR-", ";-").replace("-", " ")

    merged["feature_text"] = [process_feature_text(x) for x in merged["feature_text"]]

    return merged
```

```
class SubmissionDataset(Dataset):
    def __init__(self, data, tokenizer, config):
        self.data = data
        self.tokenizer = tokenizer
        self.config = config

    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        example = self.data.loc[idx]
        tokenized = self.tokenizer(
            example["feature_text"],
            example["pn_history"],
            truncation = self.config['truncation'],
            max_length = self.config['max_length'],
            padding = self.config['padding'],
            return_offsets_mapping = self.config['return_offsets_mapping']
        )
        tokenized["sequence_ids"] = tokenized.sequence_ids()

        input_ids = np.array(tokenized["input_ids"])
        attention_mask = np.array(tokenized["attention_mask"])
        token_type_ids = np.array(tokenized["token_type_ids"])
        offset_mapping = np.array(tokenized["offset_mapping"])
        sequence_ids = np.array(tokenized["sequence_ids"]).astype("float16")

        return input_ids, attention_mask, token_type_ids, offset_mapping, sequence_ids


test_df = create_test_df()

submission_data = SubmissionDataset(test_df, tokenizer, hyperparameters)
submission_dataloader = DataLoader(submission_data, batch_size=hyperparameters['batch_size'],
shuffle=False)
```

```
model.eval()
preds = []
offsets = []
seq_ids = []

for batch in tqdm(submission_dataloader):
    input_ids = batch[0].to(DEVICE)
    attention_mask = batch[1].to(DEVICE)
    token_type_ids = batch[2].to(DEVICE)
    offset_mapping = batch[3]
    sequence_ids = batch[4]

    logits = model(input_ids, attention_mask, token_type_ids)

    preds.append(logits.detach().cpu().numpy())
    offsets.append(offset_mapping.numpy())
    seq_ids.append(sequence_ids.numpy())

preds = np.concatenate(preds, axis=0)
offsets = np.concatenate(offsets, axis=0)
seq_ids = np.concatenate(seq_ids, axis=0)
```

```
print(score)
```

```
location_preds = get_location_predictions(preds, offsets, seq_ids, test=True)
```

```
test_df["location"] = location_preds
```

```
test_df[["id", "location"]].to_csv("submission.csv", index = False)
pd.read_csv("submission.csv").head()
```