Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων
Και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης
Και Υπολογιστων

Χρήση Τεχνικών Βαθιάς Μηχανικής Μάθησης για την Εύρεση Δεδομένων από το Twitter που Σχετίζονται με Καταστροφές

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Κωνσταντίνας Καραΐσκου

Επιβλέπων: Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.
Συνεπιβλέπων: Παρασκευή Τζούβελη
Ε.ΔΙ.Π. Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων
Και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης
Και Υπολογιστων

Χρήση Τεχνικών Βαθιάς Μηχανικής Μάθησης για την Εύρεση Δεδομένων από το Twitter που Σχετίζονται με Καταστροφές

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Κωνσταντίνας Καραΐσκου

Επιβλέπων: Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.
Συνεπιβλέπων: Παρασκευή Τζούβελη
Ε.ΔΙ.Π. Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Ιουλίου 2021.

.....	.....	.....
Γιώργος Στάμου	Στέφανος Κόλλιας	Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

.....
Κωνσταντίνα Καραΐσκου
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright ©Κωνσταντίνα Καραΐσκου, 2021.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Τα μέσα κοινωνικής δικτύωσης, και ειδικότερα το Twitter, παίζουν έναν αναπόσπαστο ρόλο κατά τη διάρκεια μιας καταστροφής, παρέχοντας άμεσα πληροφορίες που είναι πολύ σημαντικές για την ενημέρωση της εξέλιξης των γεγονότων και για τις επιχειρήσεις διάσωσης. Για το λόγο αυτό είναι πολύ σημαντική η ανάκτηση και ταξινόμηση των σχετικών μηνυμάτων. Στόχος λοιπόν της παρούσας διπλωματικής εργασίας είναι η εφαρμογή και σύγκριση μεθόδων βαθιάς μηχανικής μάθησης για την εύρεση δεδομένων από το Twitter που σχετίζονται με καταστροφές. Αρχικά, κάναμε μία βιβλιογραφική ανασκόπηση για τα δεδομένα και τα μοντέλα που χρησιμοποιούνται συνήθως σε παρόμοια προβλήματα. Ύστερα, χρησιμοποιώντας δεδομένα από πολλαπλές πηγές, δημιουργήσαμε πέντε σύνολα δεδομένων για διαφορετικούς τύπους καταστροφών. Στη συνέχεια, εκτελέσαμε πειράματα με επτά διαφορετικά νευρωνικά δίκτυα και με πέντε σύγχρονα προ-εκπαιδευμένα γλωσσικά μοντέλα σε καθένα από τα σύνολα δεδομένων ξεχωριστά. Τα αποτελέσματα για τα νευρωνικά δίκτυα έδειξαν ότι η υψηλότερη ακρίβεια επιτεύχθηκε από τα μοντέλα LSTM ή CNN σε όλα τα σύνολα δεδομένων. Όσον αφορά τα προ-εκπαιδευμένα μοντέλα, η κατάταξή τους ως προς την αποτελεσματικότητα τους διέφερε ανά σύνολο δεδομένων. Συνολικά, στις περισσότερες περιπτώσεις τα προ-εκπαιδευμένα μοντέλα με τη χαμηλότερη ακρίβεια έδωσαν καλύτερα αποτελέσματα από τα νευρωνικά δίκτυα με την υψηλότερη ακρίβεια.

Λέξεις-Κλειδιά: Βαθιά μηχανική μάθηση, Νευρωνικά Δίκτυα, Μεταφορά μάθησης, BERT, Διαχείριση καταστροφών, Twitter, Αυτόματη ανάλυση κειμένου

Abstract

Social media, especially Twitter, play an integral part during disasters by providing real-time updates that are crucial for developing situational awareness and for rescue operations. Therefore, the retrieval and classification of the relevant messages is very important. The aim of this diploma thesis is the application and comparison of several deep learning methods for the detection of disaster-related messages from Twitter. Firstly, we conducted a literature review of the datasets and models that are usually used in similar problems. Then, we created five datasets, each referring to a different disaster, using data from multiple sources. Next, we executed experiments using seven different neural networks and five state-of-the-art pre-trained language models on each dataset separately. For the neural networks, the results showed that the highest accuracy was achieved by the models LSTM or CNN in all datasets. As far as the pre-trained models are concerned, their ranking in respect to their efficacy was different per dataset. Finally, in most occasions the pre-trained models with the lowest accuracy outperformed the neural networks with the highest accuracy.

Keywords: Deep learning, Neural networks, Transfer learning, BERT, Disaster management, Twitter, Automatic text analysis

Ευχαριστίες

Η παρούσα διπλωματική εργασία ολοκληρώνει τις σπουδές μου στο Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών στο επιστημονικό πεδίο "Επιστήμη Δεδομένων και Μηχανική Μάθηση" του Εθνικού Μετσόβιου Πολυτεχνείου και είναι αποτέλεσμα της συνεργασίας με διάφορους ανθρώπους που με βοήθησαν καθ' όλη τη διάρκεια της εκπόνησης της.

Καταρχήν, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή, κ.Γιώργο Στάμου, για την εμπιστοσύνη που μου έδειξε και τη βοήθεια που μου προσέφερε σε αυτήν την πολύ ενδιαφέρουσα διπλωματική εργασία που εκπόνησα στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Επίσης, οφείλω ένα μεγάλο ευχαριστώ στην κα.Παρασκευή Τζούβελη, Ε.ΔΙ.Π. Ε.Μ.Π., για την εξαιρετική συνεργασία που είχαμε όλο αυτό το διάστημα και για τον πολύτιμο χρόνο που διέθεσε καθοδηγώντας με σε όλα τα στάδια της εργασίας.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου για την οικογένεια και τους φίλους μου, για την υπομονή και τη συμπαράσταση που έδειξαν καθ' όλη τη διάρκεια των σπουδών μου και κυρίως στους γονείς μου που δε σταμάτησαν να με στηρίζουν.

Κωνσταντίνα Καραΐσκου
Αθήνα, Ιούλιος 2021

Περιεχόμενα

Ευρετήριο Εικόνων

Ευρετήριο Πινάκων

1 Εισαγωγή

1.1 Σκοπός της Εργασίας

Κατά τη διάρκεια μιας καταστροφής, η άμεση εξασφάλιση όσο περισσότερων πληροφοριών γίνεται για την εξέλιξη των γεγονότων είναι τεράστιας σημασίας. Η πρόσβαση σε πληροφορίες είναι απαραίτητη για την ανάπτυξη της επίγνωσης της κατάστασης και μπορεί να αποτελέσει ζήτημα μεταξύ ζωής και θανάτου. Μία πηγή τέτοιων πληροφοριών που έχει μπει στο επίκεντρο του ενδιαφέροντος τα τελευταία χρόνια είναι τα μέσα κοινωνικής δικτύωσης. Οι χρήστες του Twitter, για παράδειγμα, γράφουν για τις εξελίξεις και την ανάκαμψη από μία καταστροφή σε συνδυασμό με πολλά άλλα διαφορετικά θέματα. Η ανάκτηση αυτών των πληροφοριών μπορεί να οδηγήσει σε σημαντικές βελτιώσεις στη διαχείριση καταστροφών. Σε αντίθεση με άλλες πηγές πληροφοριών, οι δημοσιεύσεις στα μέσα κοινωνικής δικτύωσης εμφανίζονται σχεδόν αμέσως όποτε υπάρχει μία νέα καταστροφή, και συνεπώς η πληροφορία μπορεί να μεταδοθεί πολύ γρήγορα με αυτόν τον τρόπο. Εκτός από γεγονότα που αφορούν την καταστροφή, τα μέσα κοινωνικής δικτύωσης προσφέρουν προσωπικές αντιλήψεις στα συμβάντα, πληροφορίες για οργανισμούς παροχής περίθαλψης και άλλες επίσημες συστάσεις. Από την πλευρά των χρηστών, το 69% των Αμερικανών θεωρούν ότι οι υπηρεσίες έκτακτης ανάγκης πρέπει να απαντούν σε κλήσεις βοηθείας που στέλνονται μέσω των δικτύων κοινωνικής δικτύωσης. Σε αυτήν την εργασία επικεντρωνόμαστε στο Twitter καθώς τα περισσότερα από τα υπόλοιπα μέσα κοινωνικής δικτύωσης δεν προσφέρουν τη δυνατότητα πρόσβασης σε μεγάλη ποσότητα των δεδομένων τους σε εξωτερικούς ερευνητές, ή δεν χρησιμοποιούνται με τρόπο που να διευκολύνει την άμεση απόκτηση πληροφοριών κατά τη διάρκεια μιας καταστροφής [1].

Η ουσία του ζητήματος βρίσκεται στην ανάκτηση και ταξινόμηση των σχετικών μηνυμάτων. Οι χρήστες του Twitter παγκοσμίως παράγουν περίπου 5,800 tweets ανά δευτερόλεπτο. Σε κάθε περιστατικό που συμβαίνει, η πλειοψηφία αυτών των tweets δε θα είναι σχετική με το γεγονός ή χρήσιμη στους πάροχους υπηρεσιών [1]. Ένας ευθύς τρόπος συλλογής μηνυμάτων που σχετίζονται με καταστροφές είναι το φιλτράρισμα, όπου χρησιμοποιείται ένα λεξικό με σχετικές λέξεις κλειδιά. Αυτή η προσέγγιση αποτυγχάνει στις περιπτώσεις όπου η ορολογία σχετική με καταστροφές είναι ποικίλη και διφορούμενη και όπου οι περιγραφικοί όροι, όπως είναι τα hashtags, επιλέγονται ατομικά από τους χρήστες, ενώ πολλά μηνύματα χρησιμοποιούν λανθασμένη ορολογία. Συνεπώς, η εύρεση μηνυμάτων σχετικών με καταστροφές μοντελοποιείται πολύ συχνά ως πρόβλημα ταξινόμησης [2]. Αυτό το πρόβλημα αποτελεί το πρώτο βήμα στην ταξινόμηση tweets και μπορεί να ακολουθηθεί από άλλων ειδών ταξινόμησης, όπως είναι η ανάλυση συναισθημάτων σε μία καταστροφή ή η ταξινόμηση των tweets σε κλάσεις πληροφορίας, όπως είναι ο εθελοντισμός ή οι υλικές ζημιές.

Σκοπός λοιπόν της παρούσας εργασίας είναι η εφαρμογή και σύγκριση πολλών μεθόδων βαθιάς μηχανικής μάθησης για την εύρεση δεδομένων από το Twitter που σχετίζονται με καταστροφές. Συγκεκριμένα, πραγματοποιούμε σύγκριση ανάμεσα σε νευρωνικά δίκτυα και σε σύγχρονα προ-εκπαιδευμένα γλωσσικά μοντέλα.

1.2 Δομή της Εργασίας

Η διπλωματική εργασία αποτελείται από άλλα πέντε κεφάλαια, τα οποία θα παρουσιαστούν συνοπτικά σε αυτήν την ενότητα για την καλύτερη εποπτεία του αναγνώστη.

Στο δεύτερο κεφάλαιο γίνεται μία βιβλιογραφική ανασκόπηση από σχετικές έρευνες που έχουν δημοσιευθεί τα τελευταία χρόνια. Αρχικά, αναλύονται μερικά σύνολα δεδομένων που έχουν χρησιμοποιηθεί από έρευνες και στη συνέχεια αναφέρονται οι μέθοδοι που χρησιμοποιούνται συνήθως για την ταξινόμηση πληροφοριών από το Twitter με βάση τη σχετικότητα τους με καταστροφές, με βάση το περιεχόμενο τους ή με βάση το συναίσθημα.

Στο τρίτο κεφάλαιο εξηγούνται όλες οι θεωρητικές έννοιες που αναφέρονται κατά μήκος όλης της εργασίας. Αρχικά, αναλύονται κάποιες γενικές έννοιες, όπως είναι οι συναρτήσεις ενεργοποίησης, ύστερα παρουσιάζονται μερικές συγκεκριμένες τεχνικές που χρησιμοποιούμε, όπως είναι οι τεχνικές ομαλοποίησης και η βελτιστοποίηση υπερ-παραμέτρων. Στη συνέχεια, εξηγούνται μερικές έννοιες της Επεξεργασίας Φυσικής Γλώσσας και τέλος περιγράφονται όλες οι αρχιτεκτονικές νευρωνικών δικτύων και προ-εκπαιδευμένων γλωσσικών μοντέλων που χρησιμοποιούμε.

Το τέταρτο κεφάλαιο αφορά τη μεθοδολογία που ακολουθήσαμε. Συγκεκριμένα, αναλύονται λεπτομερώς τα δεδομένα που χρησιμοποιούμε και η προ-επεξεργασία τους, καθώς και η δομή του κάθε μοντέλου ξεχωριστά. Δηλαδή, εξηγείται αναλυτικά η επιλογή των παραμέτρων του κάθε μοντέλου, ενώ παρουσιάζονται τα αποτελέσματα της βελτιστοποίησης των υπερ-παραμέτρων.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα από όλα τα πειράματα που εκτελέσαμε. Αρχικά, παρατίθενται τα αποτελέσματα από τα νευρωνικά δίκτυα, στη συνέχεια τα αποτελέσματα από τα προ-εκπαιδευμένα γλωσσικά μοντέλα και τέλος πραγματοποιείται μία σύγκριση μεταξύ των δύο αυτών κατηγοριών πειραμάτων.

Τέλος, στο έκτο κεφάλαιο γίνεται μία σύνοψη της εργασίας και εξάγονται τα σχετικά συμπεράσματα με βάση όλα τα προαναφερόμενα στοιχεία, ενώ προτείνονται ιδέες για μελλοντική εργασία.

2 Σχετική έρευνα

Οι υπάρχουσες έρευνες μελετούν το πρόβλημα ταξινόμησης των tweets, που αφορούν μία καταστροφή, συνήθως με δύο τρόπους. Στον πρώτο τρόπο, τα μηνύματα από το Twitter προέρχονται από μία συγκεκριμένη καταστροφή, όπως είναι οι πλημμύρες στην Alberta το 2013, και χρησιμοποιούνται για την εκπαίδευση και την αξιολόγηση του μοντέλου. Στον δεύτερο τρόπο, συνδυάζονται μηνύματα από πολλές καταστροφές του ίδιου τύπου για την εκπαίδευση, ενώ διαφορετικά μηνύματα από άλλες καταστροφές χρησιμοποιούνται για την αξιολόγηση. Ο πρώτος τρόπος είναι πιο εύκολος αλλά δεν γενικεύει στην πράξη όπως ο δεύτερος τρόπος, ενώ ο δεύτερος τρόπος χρειάζεται πολλά δεδομένα [2]. Σε αυτήν την εργασία, συνδυάζουμε πολλά σύνολα δεδομένων για κάθε τύπο καταστροφής.

Υπάρχουν μερικά διαθέσιμα σύνολα δεδομένων που περιέχουν δεδομένα από το Twitter και που συγκεντρώθηκαν κατά τη διάρκεια καταστροφών. Αυτά συνήθως επικεντρώνονται στο περιεχόμενο των μηνυμάτων, ενώ σπάνια θα περιέχουν και εικόνες. Οι κλάσεις ποικίλουν στα σύνολα δεδομένων. Κάποια περιέχουν πληροφορίες μόνο για τη σχετικότητα των δεδομένων με την καταστροφή, ενώ άλλα περιέχουν κλάσεις με βάση τον τύπο του περιεχομένου, την πηγή της πληροφορίας ή την προτεραιότητα που έχει το κάθε tweet. Το αρνητικό με αυτά τα δεδομένα είναι ότι περιέχουν ένα ποσοστό μόνο από τα συνολικά tweets που αποστέλνονται την αντίστοιχη περίοδο, αλλά είναι και πάλι πολύ περισσότερα από τα δεδομένα που είναι διαθέσιμα από άλλα μέσα κοινωνικής δικτύωσης. Επιπλέον, αυτά τα σύνολα δεδομένων είναι συνήθως στα αγγλικά, ενώ λιγότερα από το 1% όλων των tweets περιέχουν γεωγραφικές πληροφορίες, που συχνά είναι απαραίτητες για μία ανάλυση [1]. Μερικά από αυτά τα σύνολα δεδομένων είναι τα εξής:

Events2012 Αυτό το σύνολο δεδομένων περιέχει 120 εκατομμύρια tweets, από τα οποία τα 150,000 έχουν καταταχτεί σε ένα από τα 506 γεγονότα (που δεν είναι απαραίτητα καταστροφές) [3].

CrisisLex Το CrisisLex είναι μία συλλογή από σύνολα δεδομένων για καταστροφές. Τα σύνολα δεδομένων μπορεί να αφορούν είτε συγκεκριμένες καταστροφές είτε περισσότερα περιστατικά. Πολλά από τα δεδομένα έχουν καταταχτεί σε κλάσεις από ανθρώπους [4].

CrisisNLP Όπως και το CrisisLex έτσι και το CrisisNLP περιλαμβάνει μία συλλογή από σύνολα δεδομένων για καταστροφές. Συγκεκριμένα, η ομάδα που είναι υπεύθυνη για το CrisisNLP σύλλεξε tweets από 19 καταστροφές και τα δημοσίευσε για έρευνα. Από τα συνολικά 53 εκατομμύρια tweets που είναι διαθέσιμα σε αυτά τα δεδομένα, τα 50,000 έχουν ταξινομηθεί σε κλάσεις από ανθρώπους [5].

CrisisMMD Το CrisisMMD είναι μία ενδιαφέρουσα περίπτωση καθώς περιλαμβάνει μόνο tweets που περιέχουν και κείμενο και εικόνα. 16,000 tweets συλλέχθηκαν από εφτά γεγονότα που έγιναν το 2017 σε πέντε χώρες. Η ταξινόμηση έγινε από ανθρώπους για το κείμενο και τις εικόνες ξεχωριστά. Το κάθε tweet ταξινομείται σε τρεις κλάσεις, σχετική/μη σχετική πληροφορία, οχτώ θεματικές κατηγορίες, όπως εθελοντισμός ή πληγμένα άτομα, και σοβαρότητα της ζημιάς που εφαρμόζεται μόνο σε εικόνες [6].

Epic Αυτό το σύνολο δεδομένων είναι επικεντρωμένο στον τυφώνα Sandy και συλλέχθηκε με διαφορετικό τρόπο από τα υπόλοιπα. Η ομάδα πρώτα συγκέντρωσε tweets που περιλάμβαναν hashtags σχετικά με τον τυφώνα και στη συνέχεια τα συγκέντρωσαν ανά χρήστη. Από αυτούς τους χρήστες επέλεξαν αυτούς που είχαν γεωγραφικές πληροφορίες (geotagged) στα tweets για την περιοχή που πλήγηκε, θεωρώντας ότι αυτοί οι χρήστες επηρεάστηκαν από τον τυφώνα. Ύστερα, 105 από αυτούς τους χρήστες επιλέχθηκαν τυχαία, ενώ συγκεντρώθηκαν τα tweets τους από μία βδομάδα πριν την καταστροφή μέχρι μία βδομάδα μετά την καταστροφή. Αυτό είχε ως αποτέλεσμα τη δημιουργία ενός συνόλου δεδομένων που περιέχει τόσο σχετικά όσο και άσχετα tweets από

τους ίδιους χρήστες. Τα tweets κατατάχθηκαν με βάση τη σχετικότητά τους, καθώς και σε 17 θεματικές κατηγορίες και συναισθήματα [7].

Florence Το σύνολο δεδομένων Florence περιέχει 600,000 tweets που συγκεντρώθηκαν στην περιοχή που πλήγηκε από τον τυφώνα Florence την εβδομάδα 10-17 Σεπτεμβρίου το 2018. Τα tweets δεν φιλτραρίστηκαν και συνεπώς ένα υποσύνολο τους σχετίζεται με τον τυφώνα. Τα αρχικά δεδομένα φιλτραρίστηκαν με διαφορετικούς τρόπους και η επικάλυψη αυτών των αποτελεσμάτων θεωρήθηκε ότι περιλαμβάνει με μεγάλη σιγουριά tweets που σχετίζονται με την καταστροφή. Το τελικό αποτέλεσμα περιλαμβάνει γύρω στα 20,000 tweets [8].

Disaster Tweet Corpus 2020 Αυτό το σύνολο δεδομένων περιέχει tweets που έχουν συλλεχθεί και καταταχτεί σε κλάσεις από αρκετές άλλες έρευνες και καλύπτει 48 διαφορετικές καταστροφές που ανήκουν σε 10 διαφορετικούς τύπους καταστροφών. Όλα τα tweets έχουν ταξινομηθει με βάση το αν είναι σχετικά ή όχι με την αντίστοιχη καταστροφή [9].

TREC-IS 2019A Ένα πρόβλημα ταξινόμησης σχετικό με καταστροφές που ονομάστηκε "Incident Streams" ήταν μέρος του Text REtrieval Conference (TREC) που οργανώθηκε από το NIST το 2018. Αρχικά, συγκεντρώθηκαν αυτόματα tweets από έξι γεγονότα με τη χρήση λέξεων κλειδιών και στη συνέχεια ταξινομήθηκαν στον έναν από τους 25 τύπους κλάσεων. Το σύνολο δεδομένων εμπλουτίστηκε το 2019 από περισσότερα δεδομένα, ενώ το κάθε tweet πλέον ανήκει σε πολλαπλές κλάσεις. Η πιο πρόσφατη έκδοση του συνόλου δεδομένων περιέχει περίπου 30,000 tweets από 15 γεγονότα [10].

Appen Disaster Response Messages Αυτό το σύνολο δεδομένων περιέχει 30,000 μηνύματα που συγκεντρώθηκαν κατά τη διάρκεια καταστροφών μεταξύ του 2010 και του 2012. Αυτά τα tweets ταξινομήθηκαν σε 36 κατηγορίες περιεχομένου, όπως είναι η ιατρική βοήθεια ή η διάσωση, καθώς και με την ετικέτα για το αν είναι σχετικά με την καταστροφή. Επίσης, τα μηνύματα αυτά περιλαμβάνουν πολλές γλώσσες, καθώς και την αγγλική τους μετάφραση [11].

Kaggle covid-19 Αυτά τα δεδομένα περιέχουν tweets από χρήστες που χρησιμοποίησαν ένα από τα εννιά hashtags σχετικά με τον covid-19 από τον Μάρτιο του 2020. Συγκεντρώθηκαν περίπου ένα εκατομμύριο tweets ανά μήνα από περισσότερες από 200 χώρες, ενώ έχουν γίνει ήδη κάποιες αναλύσεις σε αυτά τα δεδομένα μέσω του Kaggle [12].

covid19_twitter Αυτό είναι ένα μεγαλύτερο σύνολο δεδομένων από το προηγούμενο που αφορά tweets που είναι σχετικά με τον covid-19. Τα tweets συλλέχθηκαν παγκοσμίως με τη χρήση 13 λέξεων-κλειδιών από 238 χώρες. Το σύνολο δεδομένων περιέχει επίσης τους 1000 συχνότερους όρους, bigrams και trigrams [13].

GeoCoV19 Αυτό είναι ένα ακόμα μεγαλύτερο σύνολο δεδομένων από τα δύο προηγούμενα, όπου τα μηνύματα συλλέχθηκαν με τη χρήση 800 λέξεων-κλειδιών, από την $1^{η}$ Φεβρουαρίου 2020. Για τα περισσότερα από αυτά τα tweets ανακτήθηκαν οι γεωγραφικές πληροφορίες είτε από τα γεωγραφικά μεταδεδομένα του tweet, είτε από την τοποθεσία του χρήστη, είτε από κάποια αναφορά μέρους στο κείμενο του tweet [14].

Τα τελευταία χρόνια έχουν δημοσιευθεί πολλές έρευνες για την ταξινόμηση πληροφοριών από το Twitter με βάση τη σχετικότητα τους με καταστροφές, με βάση το περιεχόμενο τους ή με βάση το συναίσθημα. Σε κάποιες από αυτές τις έρευνες χρησιμοποιήθηκαν παραδοσιακές τεχνικές μηχανικής μάθησης, σε άλλες τεχνικές βαθιάς μηχανικής μάθησης, ενώ σε άλλες χρησιμοποιήθηκε η τεχνική μεταφορά μάθησης.

Οι πιο συνηθισμένες παραδοσιακές τεχνικές μηχανικής μάθησης που χρησιμοποιήθηκαν σε ταξινόμηση πληροφοριών από το Twitter είναι οι αλγόριθμοι Support Vector Machine (SVM), Random Forest, Naive Bayes και Logistic Regression. Οι J. Rexiline Ragini et al. (2018) χώρισαν αρχικά τα δεδομένα, που συλλέχθηκαν από τους συγγραφείς, σε κατηγορίες ανάλογα με τις ανάγκες των ανθρώπων, όπως νερο ή καταφύγιο, με τη χρήση λέξεων-κλειδιών και στη συνέχεια χρησιμοποίησαν τον αλγόριθμο SVM για την κατάταξη των συναισθημάτων των ανθρώπων στην κάθε κατηγορία [15]. Σε μία άλλη έρευνα, οι συγγραφείς (Gonzalo Ruz et al., 2020) χρησιμοποίησαν κυρίως μπεϋζιανά δίκτυα για την ανάλυση συναισθημάτων σε δύο σύνολα δεδομένων στα ισπανικά. Συγκεκριμένα, χρησιμοποίησαν τον Naive Bayes και τον Tree Augmented Naive Bayes, καθώς και τους αλγορίθμους SVM και Random Forest για σύγκριση, ενώ τα καλύτερα αποτελέσματα τα είχε το μοντέλο SVM [16]. Οι Jyoti Singh et al. (2019) χρησιμοποίησαν τους αλγορίθμος SVM, Gradient Boosting και Random Forest για την ταξινόμηση tweets σε μηνύματα υψηλής ή χαμηλής προτεραιότητας, ενώ τα δεδομένα συλλέχθηκαν και ταξινομήθηκαν χειροκίνητα από τους συγγραφείς. Το μοντέλο SVM είχε αρκετά χειρότερα αποτελέσματα από τα άλλα δύο μοντέλα, ενώ οι συγγραφείς ανέπτυξαν επίσης ένα σύστημα πρόβλεψης της τοποθεσίας των χρηστών που έγραψαν τα μηνύματα υψηλής προτεραιότητας με τη χρήση του μοντέλου Markov [17]. Σε μία άλλη έρευνα (Sukanya Manna et al., 2019), εφαρμόστηκαν οι αλγόριθμοι Naive Bayes, SVM και Logistic Regression, καθώς και ένα MLP στο σύνολο δεδομένων CrisisLexT6, όπου τα tweets είναι ταξινομημένα με βάση τη σχετικότητά τους με την αντίστοιχη καταστροφή, σε κάθε καταστροφή ξεχωριστά, ενώ τα καλύτερα αποτελέσματα τα είχαν τα μοντέλα SVM και Logistic Regression [18]. Τέλος, οι Beverly Parilla-Ferrer et al. (2014) χρησιμοποίησαν ένα σύνολο δεδομένων στο οποίο τα tweets ταξινομήθηκαν χειροκίνητα με βάση τη σχετικότητά τους με τη συγκεκριμένη καταστροφή, ενώ για την αυτόματη ταξινόμηση των tweets χρησιμοποιήθηκαν οι αλγόριθμοι Naive Bayes και SVM. Η έρευνα έδειξε ότι το μοντέλο SVM είχε καλύτερα αποτελέσματα [19].

Οι συχνότερες τεχνικές βαθιάς μηχανικής μάθησης που χρησιμοποιήθηκαν σε ταξινόμηση πληροφοριών από το Twitter είναι τα μοντέλα CNN και LSTM. Οι Abhinav Kumar et al. (2020) πρότειναν ένα μοντέλο για τα σύνολα δεδομένων που εμπεριέχονται CrisisMMD, τα οποία αποτελούνται και από κείμενο και από εικόνα. Το μοντέλο αυτό αποτελούνταν από δύο διαδοχικά LSTMs για το κείμενο και από το μοντέλο VGG-16, που αποτελείται από 13 επίπεδα συνέλιξης, για τις εικόνες, ενώ οι έξοδοι από τα δύο μοντέλα συνενώνονταν και περνούσαν από ένα πλήρες συνδεδεμένο επίπεδο [20]. Σε μία άλλη έρευνα (Simon O'Keefe et al., 2018), εφαρμόστηκαν τα μοντέλα CNN και BiLSTM με δύο διαφορετικά embeddings σε σύνολα δεδομένων της συλλογής CrisisNLP, όπου τα tweets είναι ταξινομημένα σε κατηγορίες με βάση το περιεχόμενο τους, ενώ το BiLSTM με τα GloVe embeddings είχε τα καλύτερα αποτελέσματα [21]. Οι Pradip Bhere et al. (2020) πρότειναν ένα CNN μοντέλο το οποίο εφάρμοσαν στο σύνολο δεδομένων Social Media Disaster Tweets του Kaggle, όπου τα tweets είναι ταξινομημένα με βάση τη σχετικότητά τους με την καταστροφή [22]. Σε μία άλλη έρευνα οι συγγραφείς (Sreenivasulu Madichetty et al., 2020) χρησιμοποίησαν διαφορετικά σύνολα δεδομένων για διαφορετικές καταστροφές, όπου τα tweets είναι ταξινομημένα με βάση τη σχετικότητά τους με την αντίστοιχη καταστροφή, και εφάρμοσαν τα μοντέλα CNN, LSTM, BiLSTM και BiLSTM με μηχανισμό προσοχής με τη χρήση διαφορετικών embeddings [23]. Τέλος, ο Sreenivasulu Madichetty (2020) ανέπτυξε ένα μοντέλο που αποτελεί συνδυασμός ενός CNN και ενός KNN μοντέλου, τα οποία εκπαιδεύονται ξεχωριστά και στη συνέχεια η έξοδος των δύο αυτών μοντέλων συνενώνεται και περνάει από έναν SVM ταξινομητή. Το μοντέλο αυτό εφαρμόστηκε σε δεδομένα που έχουν ως σκοπό την ανίχνευση των tweets όπου είτε ζητούνται ή προσφέρονται είδη πρώτης ανάγκης, όπως νερό και φαγητό [24].

Σε πολλές έρευνες οι συγγραφείς σύγκριναν παραδοσιακές τεχνικές μηχανικής μάθησης με τεχνικές βαθιάς μηχανικής μάθησης, ενώ σε όλες τα μοντέλα βαθιάς μηχανικής μάθησης είχαν τα καλύτερα αποτελέσματα. Οι Venkata Kishore Neppalli et al. (2018) σύγκριναν τον Naive Bayes με δύο νευρωνικά δίκτυα, ένα CNN και ένα GRU, στο σύνολο δεδομένων CrisisLexT26, όπου τα tweets είναι ταξινομημένα με βάση τη σχετικότητά τους με την καταστροφή, το περιεχόμενο της πληροφορίας και

την πηγή, ενώ το CNN είχε τα καλύτερα αποτελέσματα [25]. Οι Md. Yasin Kabir et al. (2019) πρότειναν ένα μοντέλο που συνδυάζει το BiLSTM με μηχανισμό attention και με ένα CNN πριν την έξοδο και το αξιολόγησαν σε ένα έτοιμο σύνολο δεδομένων, όπου ταξινόμησαν χειροκίνητα ένα μερος του, καθώς και σε σύνολα δεδομένων από τις συλλογές CrisisNLP και CrisisLex. Το μοντέλο αυτό το σύγκριναν με τους αλγορίθμους Logistic Regression, SVM και με το απλό CNN μοντέλο, ενώ το προτεινόμενο μοντέλο είχε τα καλύτερα αποτελέσματα σε όλα τα σύνολα δεδομένων [26]. Σε μία άλλη έρευνα, οι συγγραφείς (Abhinav Kumar et al., 2019) εξέτασαν τις μεθόδους SVM, Random Forest, Logistic Regression, KNN, Naive Bayes, Gradient Boosting, Decision Tree, CNN, LSTM, GRU, BiGRU και GRU-CNN σε έτοιμα σύνολα δεδομένων, όπου τα tweets είναι ταξινομημένα σε έξι κλάσεις ανάλογα με το περιεχόμενο τους. Στο κάθε σύνολο δεδομένων διαφορετικοί ταξινομητές είχαν το καλύτερο αποτέλεσμα, αλλά κυριώς ήταν τα μοντέλα LSTM, BiGRU, GRU και GRU-CNN [27]. Οι Dat Nguyen et al. (2016) ανέπτυξαν ένα CNN μοντέλο με δύο διαφορετικά embeddings και το εφάρμοσαν σε δυαδική ταξινόμηση, καθώς και σε ταξινόμηση πολλαπλών κλάσεων, σε δεδομένα από τις συλλογές CrisisNLP, CrisisLex και AIDR. Στη συνέχεια, σύγκριναν τα αποτελέσματα με τους αλγορίθμους Random Forest, Logistic Regression και SVM στις δυαδικές ταξινομήσεις για κάθε σύνολο δεδομένων ξεχωριστά, και με τα μοντέλα SVM και MLP-CNN στις ταξινομήσεις πολλαπλών κλάσεων για κάθε σύνολο δεδομένων ξεχωριστά. Στην πρώτη περίπτωση το CNN μοντέλο είχε τα καλύτερα αποτελέσματα, ενώ στη δεύτερη περίπτωση το μοντέλο MLP-CNN είχε την καλύτερη ακρίβεια [28]. Οι Dat Tien Nguyen et al. (2017) επίσης ανέπτυξαν ένα CNN μοντέλο το οποίο εφάρμοσαν με δύο διαφορετικά embeddings σε δεδομένα από τις συλλογές CrisisNLP, CrisisLex και AIDR, ενώ η χρήση των crisis embeddings έδωσε λίγο καλύτερα αποτελέσματα. Τα μοντέλα CNN συγκρίθηκαν επίσης με τα μοντέλα SVM, Logistic Regression και Random Forest, όπου τα CNN είχαν πολύ καλύτερα αποτελέσματα [29]. Οι Sreenivasulu Madichetty et al. (2019) πρότειναν ένα μοντέλο που αποτελεί συνδυασμός ενός CNN και ενός ANN και το εφάρμοσαν σε ένα σύνολο δεδομένων όπου τα tweets έχουν καταταχτεί σε κλάσεις ανάλογα με τη σχετικότητά τους με την καταστροφή. Το μοντέλο αυτό συγκρίθηκε με τα μοντέλα CNN, ANN και SVM και είχε τα καλύτερα αποτελέσματα, ενώ τα δεύτερα καλύτερα αποτελέσματα τα είχε το μοντέλο CNN [30]. Τέλος, σε μία άλλη έρευνα, οι συγγραφείς (Ashwin Devaraj et al., 2020) πήραν δεδομένα από το Twitter για μία καταστροφή και αφού τα ταξινόμησαν χειροκίνητα με βάση το πόσο επείγον είναι το περιεχόμενο των μηνυμάτων, χρησιμοποίησαν τα μοντέλα Naive Bayes, Decision Tree, AdaBoost, SVM, MLP, CNN, Logistic Regression και Ridge Regression για την αυτόματη ταξινόμηση τους. Οι αλγόριθμοι CNN και SVM έδωσαν τα καλύτερα αποτελέσματα [31].

Σε μερικές έρευνες χρησιμοποιήθηκαν προ-εκπαιδευμένα γλωσσικά μοντέλα, όπως είναι το BERT, που αποτελούν τις πιο σύγχρονες μεθόδους που χρησιμοποιούνται σε προβλήματα επεξεργασίας της φυσικής γλώσσας. Οι Hamada M. Zahera et al. (2019) εφάρμοσαν το μοντέλο BERT στο σύνολο δεδομένων TREC-IS, όπου το κάθε tweet ανήκει σε πολλαπλές κλάσεις ανάλογα με το περιεχόμενο του [32]. Σε μία άλλη έρευνα, ο συγγραφέας (Warih Maharani, 2020) συγκέντρωσε tweets από μία καταστροφή και τα ταξινόμησε σε δύο κλάσεις ανάλογα με τη σχετικότητα του κάθε tweet με τη συγκεκριμένη καταστροφή. Στα δεδομένα αυτά εφάρμοσε το προ-εκπαιδευμένο μοντέλο BERT, όπου πρόσθεσε στο τέλος ένα MLP [33]. Οι Pallavi Jain et al. (2019) χρησιμοποίησαν τα embeddings Word2Vec, GloVe, ELMo και BERT σε συνδυασμό με ένα MLP σε δεδομένα από τις συλλογές CrisisLex και CrisisNLP, τα οποία τα συνδύασαν με βάση τον τύπο της καταστροφής. Τα αποτελέσματα ήταν παρόμοια για όλα τα μοντέλα, ενώ σε διαφορετικούς τύπους καταστροφών υπερίσχυαν διαφορετικά μοντέλα [34]. Οι Matti Wiegmann et al. (2020), που σύνθεσαν και το Disaster Tweet Corpus 2020, χρησιμοποίησαν τα μοντέλα CNN, BERT σε συνδυασμό με MLP και USE σε συνδυασμό με MLP στα δεδομένα Disaster Tweet Corpus 2020, αφού τα χώρισαν ανάλογα με τον τύπο της καταστροφής, ενώ τα καλύτερα αποτελέσματα τα είχε το USE μοντέλο [2]. Σε μία άλλη έρευνα, οι συγγραφείς (Mohamed Barbouch et al., 2020) αρχικά σύλλεξαν και ταξινόμησαν tweets με βάση το περιεχόμενο

τους για οχτώ διαφορετικές καταστροφές. Στη συνέχεια, σύγκριναν το μοντέλο RoBERTa με το SVM, όπου το RoBERTa είχε καλύτερα αποτελέσματα, αλλά στις κλάσεις όπου δεν αντιστοιχούσαν πολλά δεδομένα, το SVM φάνηκε να τα πηγαίνει καλύτερα [35]. Οι Abdul Hameed Azeemi et al. (2021) σύλλεξαν δεδομένα από το Twitter σχετικά με τον covid-19 και τα ταξινόμησαν σε 7 κλάσεις με βάση το συναίσθημα. Ύστερα, χρησιμοποίησαν τα προ-εκπαιδευμένα μοντέλα BERT, RoBERTa, XLNet και ELECTRA για την ταξινόμηση των tweets με βάση το συναίσθημα, ενώ το μοντέλο RoBERTa είχε τα καλύτερα αποτελέσματα [36]. Τέλος, οι Hansi Hettiarachchi et al. (2020) χρησιμοποίησαν τα μοντέλα Bert, RoBERTa, AlBERT, XLNET, ELCTRA, BERTweet και covid-twitter-BERT σε ένα σύνολο δεδομένων, για την ανίχνευση των tweets που είναι σχετικά με τον covid-19, ενώ το covid-twitter-BERT είχε την καλύτερη απόδοση [37].

Από την παραπάνω βιβλιογραφική ανασκόπηση συμπεραίνουμε ότι σπάνια συγκρίνονται αρχιτεκτονικές νευρωνικών δικτύων με τα σύγχρονα προ-εκπαιδευμένα γλωσσικά μοντέλα, ενώ κυρίως συγκρίνονται παραδοσιακές τεχνικές μηχανικής μάθησης με τεχνικές βαθιάς μηχανικής μάθησης. Σε αυτή την εργασία λοιπόν, συγκρίνουμε μεθόδους βαθιάς μηχανικής μάθησης με προ-εκπαδευμένα γλωσσικά μοντέλα για την εύρεση δεδομένων από το Twitter που σχετίζονται με καταστροφές.

3 Θεωρητικό Υπόβαθρο

3.1 Συναρτήσεις Ενεργοποίησης

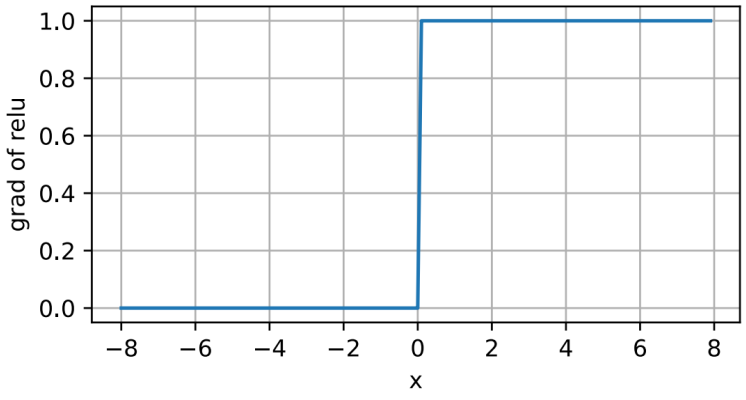
Τα νευρωνικά δίκτυα ονομάζονται νευρωνικά καθώς είναι εμπνευσμένα από τη νευροεπιστήμη. Κάθε ένα επίπεδο αποτελείται από κόμβους που μπορούν να θεωρηθούν ότι έχουν λειτουργία ανάλογη με αυτήν των νευρώνων του εγκεφάλου. Με βάση τις λειτουργίες που εκτελούν οι βιολογικοί νευρώνες χρησιμοποιούνται και στη βαθιά μηχανική μάθηση συναρτήσεις που εκτελούν παρόμοιους υπολογισμούς και που ονομάζονται συναρτήσεις ενεργοποίησης [38]. Οι συναρτήσεις αυτές αποφασίζουν εάν ένας νευρώνας θα ενεργοποιηθεί ή όχι υπολογίζοντας το σταθμισμένο άθροισμα και προσθέτοντας bias σε αυτό. Οι συναρτήσεις αυτές περιλαμβάνουν διαφοροποιήσιμες λειτουργίες που μετασχηματίζουν εισόδους σε εξόδους, ενώ οι περισσότερες προσθέτουν μία μη γραμμικότητα [39]. Σε αυτήν την ενότητα αναλύουμε τις συναρτήσεις ενεργοποίησης που αναφέρονται σε επόμενες ενότητες.

3.1.1 Ανορθωμένη Γραμμική Μονάδα

Η πιο δημοφιλής επιλογή συνάρτησης ενεργοποίησης, τόσο λόγω της απλότητας της υλοποίησης της όσο και της καλής της απόδοσης σε ένα μεγάλο εύρος προβλημάτων, είναι η ανορθωμένη γραμμική μονάδα (Rectified Linear Unit, ReLU), που θα τη λέμε ReLU για συντομία. Η ReLU παρέχει έναν πολύ απλό μη γραμμικό μετασχηματισμό. Δεδομένου ενός στοιχείου x, η συνάρτηση ορίζεται ως το μέγιστο αυτού του στοιχείου και του 0. Δηλαδή:

$$ReLU(x) = max(x, 0).$$

Πρακτικά, η συνάρτηση ReLU κρατάει μόνο τα θετικά στοιχεία και απορρίπτει τα αρνητικά, θέτοντας τις αντίστοιχες ενεργοποιήσεις σε μηδέν. Όταν η είσοδος είναι αρνητική το παράγωγο της συνάρτησης είναι 0, ενώ όταν η είσοδος είναι θετική το παράγωγο της συνάρτησης είναι 1. Σημειώνουμε ότι η συνάρτηση ReLU δεν είναι διαφοροποιήσιμη όταν η είσοδος παίρνει ακριβώς την τιμή 0. Σε αυτές τις περιπτώσεις θεωρούμε ότι το παράγωγο είναι 0 όταν η είσοδος είναι 0. Στην παρακάτω εικόνα φαίνεται το παράγωγο της συνάρτησης ReLU:



Εικόνα 1: Το παράγωγο της συνάρτησης ReLU [39].

Ο λόγος που χρησιμοποιείται η ReLU είναι ότι τα παράγωγα της συμπεριφέρονται ιδιαίτερα καλά. Δηλαδή, είτε εξαφανίζουν το στοιχείο είτε το αφήνουν να περάσει. Με αυτόν τον τρόπο η βελτιστοποίηση λειτουργεί καλύτερα καθώς μετριάζεται το πρόβλημα των vanishing gradients. Το πρόβλημα αυτό προκύπτει από τον πολλαπλασιασμό πολλών πιθανοτήτων, όπου το γινόμενο μπορεί να είναι είτε πολύ

μικρό, είτε πολύ μεγάλο. Όπως αναφέρουμε και σε επόμενη ενότητα, οι κλίσεις (gradients) είναι οι αλλαγές στα βάρη του δικτύου κατά την εκπαίδευση και προκύπτουν από τον πολλαπλασιασμό πολλών πινάκων. Καθώς υπάρχει περίπτωση οι κλίσεις να μην μπορούν να αναπαρασταθούν αριθμητικά, απειλείται η σταθερότητα των αλγορίθμων βελτιστοποίησης. Δηλαδή, μπορεί οι ενημερώσεις των παραμέτρων να είναι υπερβολικά μεγάλες, καταστρέφοντας με αυτόν τον τρόπο το μοντέλο (πρόβλημα των exploding gradients), ή να είναι υπερβολικά μικρές (πρόβλημα των vanishing gradients), καθιστώντας αδύνατη την εκπαίδευση καθώς οι παράμετροι δεν αλλάζουν σχεδόν καθόλου τιμή κατά την ενημέρωση τους. Το πρόβλημα των vanishing gradients προκαλείται συνήθως από την επιλογή της συνάρτησης ενεργοποίησης. Συνεπώς, καθώς η συνάρτηση ReLU μετριάζει αυτό το πρόβλημα είναι μία καλή επιλογή ως συνάρτηση ενεργοποίησης [39].

3.1.2 Σιγμοειδής Συνάρτηση

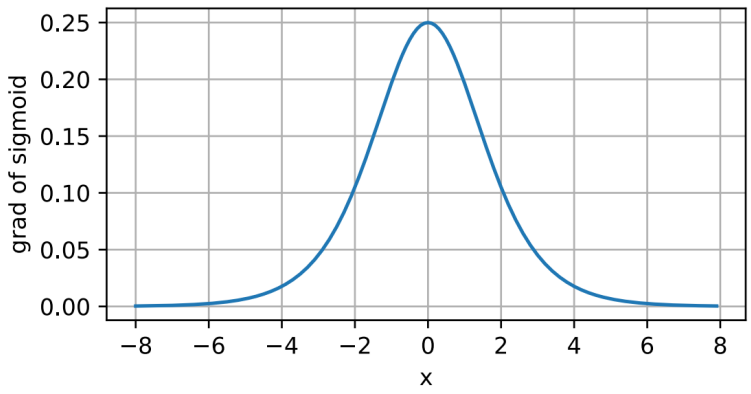
Η σιγμοειδής συνάρτηση (sigmoid) μετατρέπει της είσοδό της, που οι τιμές της ανήκουν στο σύνολο $\mathbb{R}$, σε έξοδο που ανήκει στο διάστημα $(0, 1)$:

$$sigmoid(x) = \frac{1}{1 + exp(-x)}.$$

Όταν η είσοδος πλησιάζει το 0, η σιγμοειδής συνάρτηση λειτουργεί παρόμοια με ένα γραμμικό μετασχηματισμό. Το παράγωγο αυτής της συνάρτησης δίνεται από την παρακάτω εξίσωση:

$$\frac{d}{dx} sigmoid(x) = \frac{exp(-x)}{(1 + exp(-x))^2} = sigmoid(x)(1 - sigmoid(x)).$$

Όταν η είσοδος είναι 0, το παράγωγο της συνάρτησης παίρνει τη μέγιστη τιμή 0.25, ενώ όταν η είσοδος αποκλίνει από το 0 προς οποιαδήποτε από τις δύο κατευθύνσεις, το παράγωγο τείνει να γίνει 0. Το παράγωγο της σιγμοειδούς συνάρτησης φαίνεται στην παρακάτω εικόνα:



Εικόνα 2: Το παράγωγο της σιγμοειδούς συνάρτησης [39].

Ιστορικά, η σιγμοειδής συνάρτηση ήταν δημοφιλής καθώς μοιάζει με τις συναρτήσεις κατωφλίου, στις οποίες η έξοδος παίρνει τιμή 0 όταν η είσοδος είναι μικρότερη από ένα κατώφλι και τιμή 1 όταν η είσοδος υπερβαίνει το κατώφλι. Όμως, η συνάρτηση αυτή μπορεί να προκαλέσει το πρόβλημα των vanishing gradients, καθώς οι κλίσεις εξαφανίζονται όταν η είσοδος είναι μεγάλη αλλά και όταν η είσοδος είναι μικρή. Συνεπώς, η σιγμοειδής συνάρτηση έχει αντικατασταθεί από την πιο απλή συνάρτηση ReLU στα κρυφά επίπεδα των νευρωνικών δικτύων. Ωστόσο, η σιγμοειδής συνάρτηση χρησιμοποιείται

εκτενώς ως συνάρτηση ενεργοποίησης στους κόμβους του επιπέδου εξόδου, όταν θέλουμε να ερμηνεύσουμε τις εξόδους ως πιθανότητες για προβλήματα δυαδικής ταξινόμησης. Δηλαδή, εάν η έξοδος της σιγμοειδούς συνάρτησης είναι στο διάστημα $(0, 0.5)$, τότε το μοντέλο προβλέπει την κλάση με ετικέτα 0, ενώ αν η έξοδος είναι στο διάστημα $(0.5, 1)$, τότε το μοντέλο προβλέπει την κλάση με ετικέτα 1 [39].

3.1.3 Υπερβολική Εφαπτομένη

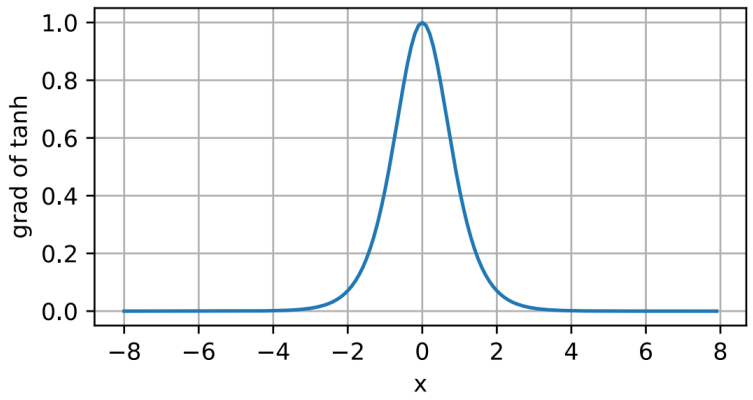
Σαν τη σιγμοειδή συνάρτηση, η υπερβολική εφαπτομένη (hyperbolic tangent, tanh) επίσης μετατρέπει την είσοδό της, που οι τιμές της ανήκουν στο σύνολο $\mathbb{R}$, σε έξοδο που ανήκει στο διάστημα $(-1, 1)$:

$$tanh(x) = \frac{1 - exp(-2x)}{1 + exp(-2x)}.$$

Όταν η είσοδος πλησιάζει το 0, η υπερβολική εφαπτομένη λειτουργεί παρόμοια με έναν γραμμικό μετασχηματισμό. Παρ'όλο που το σχήμα της μοιάζει με αυτό της σιγμοειδούς συνάρτησης, η υπερβολική εφαπτομένη παρουσιάζει συμμετρία ως προς την αρχή των αξόνων. Το παράγωγο αυτής της συνάρτησης είναι:

$$\frac{d}{dx}tanh(x) = 1 - tanh^2(x).$$

Όταν η είσοδος πλησιάζει το 0, το παράγωγο της υπερβολικής εφαπτομένης πλησιάζει τη μέγιστη τιμή 1. Ενώ, όπως και στη σιγμοειδή συνάρτηση, όταν η είσοδος αποκλίνει από το 0 προς οποιαδήποτε από τις δύο κατευθύνσεις, το παράγωγο τείνει να γίνει 0. Το παράγωγο της υπερβολικής εφαπτομένης φαίνεται στην ακόλουθη εικόνα [39]:



Εικόνα 3: Το παράγωγο της υπερβολικής εφαπτομένης [39].

3.1.4 Συνάρτηση Softmax

Η συνάρτηση softmax είναι μία γενίκευση της σιγμοειδούς συνάρτησης, που χρησιμοποιείται σε προβλήματα που υπάρχουν περισσότερες από δύο κλάσεις. Δηλαδή, χρησιμοποιείται για την ερμηνεία των εξόδων ενός μοντέλου με τη μορφή πιθανοτήτων. Στη συνέχεια, οι παράμετροι του μοντέλου βελτιστοποιούνται ώστε το μοντέλο να παράγει πιθανότητες που θα αυξάνουν την ακρίβεια των προβλέψεων. Συγκεκριμένα, κάθε έξοδος $\hat{y}_j$ ερμηνεύεται ως πιθανότητα ότι μία παρατήρηση ανήκει στην κλάση j. Στη συνέχεια, επιλέγουμε την κλάση με τη μεγαλύτερη τιμή εξόδου ως την πρόβλεψη, $argmax_j y_j$. Για παραδειγμα, εάν έχουμε $\hat{y}_1 = 0.1$, $\hat{y}_2 = 0.8$ και $\hat{y}_3 = 0/1$, τότε η κλάση που θα προβλέψει το μοντέλο είναι η 2. Για την ερμηνεία των εξόδων ως πιθανότητες, πρέπει αυτές να είναι

μη αρνητικές και να έχουν άθροισμα 1. Η συνάρτηση softmax κάνει ακριβώς αυτό:

$$\hat{y} = softmax(o) \quad \text{όπου} \quad \hat{y}_j = \frac{exp(o_j)}{\sum_k exp(o_k)}.$$

Από τον παραπάνω τύπο προκύπτει εύκολα ότι ισχύει $\hat{y}_1 + \hat{y}_2 + \hat{y}_3 = 1$ με $0 \leq \hat{y}_j \leq 1$ για όλα τα j. Επομένως, το $\hat{y}$ είναι μία κατανομή πιθανότητας που οι επιμέρους τιμές της μπορούν να ερμηνευτούν κατάλληλα. Οπότε, κατά την πρόβλεψη το μοντέλο επιλέγει την πιο πιθανή κλάση [39]:

$$argmax_j \hat{y}_j = argmax_j o_j.$$

3.2 Συναρτήσεις Κόστους

Οι περισσότεροι αλγόριθμοι βαθιάς μηχανικής μάθησης περιλαμβάνουν κάποιου είδους βελτιστοποίηση. Η βελτιστοποίηση αναφέρεται στην ελαχιστοποίηση ή μεγιστοποίηση μιας συνάρτησης $f(x)$ μεταβάλλοντας το x. Η συνάρτηση που θέλουμε να ελαχιστοποιήσουμε ή να μεγιστοποιήσουμε ονομάζεται αντικειμενική συνάρτηση (objective function) ή κριτήριο (criterion), ενώ στην περίπτωση που θέλουμε να την ελαχιστοποιήσουμε, ονομάζεται και συνάρτηση κόστους (loss function). Το σημείο που παράγει την ελάχιστη τιμή της $f(x)$ ονομάζεται ολικό ελάχιστο. Μπορεί να υπάρχει ένα μόνο ολικό ελάχιστο ή ολικό μέγιστο της συνάρτησης. Ωστόσο, είναι πιθανό να υπάρχουν τοπικά ελάχιστα που δεν είναι τα βέλτιστα. Στη βαθιά μάθηση, γίνεται βελτιστοποίηση συναρτήσεων που μπορεί να έχουν πολλά τοπικά ελάχιστα ή σημεία που περιβάλλονται από επίπεδες περιοχές, και συνεπώς η βελτιστοποίηση γίνεται πιο δύσκολη, ειδικά όταν η είσοδος της συνάρτησης είναι πολυδιάστατη. Επομένως, συνήθως συμβιβαζόμαστε με το να βρούμε μία τιμή της f που είναι αρκετά χαμηλή, αλλά που δεν είναι απαραίτητα το ολικό ελάχιστο. Η βελτιστοποίηση γίνεται με την παράγωγο της συνάρτησης, καθώς δείχνει πως πρέπει να αλλάξει το x ώστε να βελτιωθεί το y. Στις συναρτήσεις με πολλές εισόδους χρησιμοποιείται η μερική παράγωγος για την αντίστοιχη είσοδο [38]. Στη συνέχεια, αναλύουμε τις δύο συναρτήσεις κόστους που αναφέρονται σε επόμενες ενότητες.

3.2.1 Δυαδική Διασταυρούμενη Εντροπία

Η εντροπία είναι ένας τρόπος μέτρησης της αβεβαιότητας που συνδέεται με μία κατανομή $p(y)$. Η εντροπία μίας κατανομής υπολογίζεται από τον παρακάτω τύπο, όπου C είναι ο αριθμός των κλάσεων:

$$H(p) = -\sum_{c=1}^{C} p(y_c) log(p(y_c)).$$

Ωστόσο, κατά την εκπαίδευση ενός ταξινομητή δεν γνωρίζουμε την κατανομή των δεδομένων. Οπότε, υπολογίζουμε τη διασταυρούμενη εντροπία (cross-entropy) μεταξύ μιας υποθετικής κατανομής $q(y)$ και της πραγματικής κατανομής $p(x)$, την οποία δε γνωρίζουμε, με τον παρακάτω τύπο:

$$H_q(p) = -\sum_{c=1}^{C} p(y_c) log(q(y_c)).$$

Η χαμηλότερη δυνατή διασταυρούμενη εντροπία επιτυγχάνεται όταν $p = q$, διαφορετικά η διασταυρούμενη εντροπία θα έχει πάντα μεγαλύτερη τιμή από την εντροπία που υπολογίστηκε με βάση την πραγματική κατανομή. Κατά την εκπαίδευση ο ταξινομητής ψάχνει την καλύτερη δυνατή $q(y)$ που θα ελαχιστοποιήσει την διασταυρούμενη εντροπία.

Θεωρούμε ότι έχουμε ένα πρόβλημα δυαδικής ταξινόμησης με n παρατηρήσεις $x_1, ..., x_n$. Θεωρούμε, επίσης, τις τιμές 1 και 0 ως τη θετική και αρνητική ετικέτα της κλάσης y_i αντίστοιχα, και ότι το

νευρωνικό δίκτυο παραμετροποιείται από μία μεταβλητή θ. Σκοπός είναι να βρούμε το βέλτιστο θ ώστε $\hat{y}_i = p_\theta(y_i|x_i)$. Συγκεκριμένα, για πραγματικές ετικέτες y_i και προβλέψεις $\hat{y}_i = p_\theta(y_i|x_i)$, η πιθανότητα μία παρατήρηση να ταξινομηθεί ως θετική είναι $\pi_i = p_{theta}(y_i = 1|x_i)$. Συνεπώς, η συνάρτηση που πρέπει να ελαχιστοποιηθεί είναι η ακόλουθη:

$$l(\theta) = -\sum_{i=1}^{n} y_i log(\pi_i) + (1 - y_i) log(1 - \pi_i).$$

Η παραπάνω συνάρτηση ονομάζεται και κόστος διασταυρούμενης εντροπίας, $CE(y, \hat{y})$, όπου το y ακολουθεί την πραγματική κατανομή P και το $\hat{y}$ ακολουθεί την υποθετική κατανομή Q. Καθώς ο παραπάνω τύπος αφορά δυαδική ταξινόμηση, η συνάρτηση κόστους ονομάζεται και δυαδική διασταυρούμενη εντροπία (binary cross-entropy, BCE), που για συντομία θα τη λέμε BCE [39].

3.2.2 Συνάρτηση Τετραγωνικού Σφάλματος

Η πιο δημοφιλής συνάρτηση κόστους σε προβλήματα παλινδρόμησης (regression) είναι η συνάρτηση τετραγωνικού σφάλματος (squared loss function). Όταν η πρόβλεψη για μία παρατήρηση i είναι $\hat{y}^{(i)}$ και η αντίστοιχη πραγματική ετικέτα είναι $y^{(i)}$, τότε το τετραγωνικό σφάλμα δίνεται από τον παρακάτω τύπο:

$$l^{(i)}(w, b) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2,$$

όπου w και b είναι τα βάρη και η μεταβλητή μεροληψίας (bias) αντίστοιχα. Η σταθερά $1/2$ δεν έχει πραγματική σημασία αλλά είναι αρκετά βολική για τη διαγραφή του 2 όταν πάρουμε την παράγωγο της συνάρτησης. Για τη μέτρηση της ποιότητας του μοντέλου σε ολόκληρο το σύνολο δεδομένων, που αποτελείται από n παρατηρήσεις, απλά παίρνουμε τον μέσο όρο των σφαλμάτων στα δεδομένα εκπαίδευσης. Δηλαδή:

$$L(w, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(w, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(w^T x^{(i)} + b - y^{(i)})^2.$$

Κατά την εκπαίδευση του μοντέλου, ψάχνουμε να βρούμε τις παραμέτρους (w^*, b^*) που ελαχιστοποιούν το συνολικό κόστος σε όλα τα δεδομένα εκπαίδευσης [39]:

$$w^*, b^* = argmin_{w,b} L(w, b).$$

3.3 Εκπαίδευση Νευρωνικών Δικτύων

Η εκπαίδευση νευρωνικών δικτύων αποτελείται από δύο βήματα, την προς τα εμπρός διάδοση (forward propagation) και την οπισθοδρομική διάδοση (backward propagation, backpropagation). Η προς τα εμπρός διάδοση αναφέρεται στον υπολογισμό και αποθήκευση των ενδιάμεσων μεταβλητών (συμπεριλαμβανομένου της εξόδου) ενός νευρωνικού δικτύου από το επίπεδο εισόδου προς το επίπεδο εξόδου [39]. Κατά τη διάρκεια της εκπαίδευσης, η προς τα εμπρός διάδοση συνεχίζεται μέχρι την παραγωγή του κόστους. Έπειτα, ο αλγόριθμος για την οπισθοδρομική διάδοση επιτρέπει στην πληροφορία από το κόστος να ρεύσει οπισθοδρομικά στο δίκτυο με σκοπό να υπολογιστούν οι αλλαγές στα βάρη του δικτύου (κλίσεις, gradients).

Ο όρος backpropagation αναφέρεται μόνο στη μέθοδο υπολογισμού της κλίσης, ενώ κάποιος άλλος αλγόριθμος χρησιμοποιείται για να πραγματοποιήσει την εκπαίδευση με τη χρήση της κλίσης. Για την περιγραφή αυτού του αλγορίθμου είναι χρήσιμη η χρήση ενός υπολογιστικού γράφου (computational graph) [38]. Ο σχεδιασμός υπολογιστικών γράφων βοηθά στην απεικόνιση των εξαρτήσεων

των μεταβλητών και των πράξεων (operations) [39]. Χωρίς απώλεια της γενικότητας, θεωρούμε για ευκολία ότι μία λειτουργία (operation) επιστρέφει μία μόνο μεταβλητή εξόδου. Η γενικότητα με αυτόν τον τρόπο δε χάνεται καθώς η μεταβλητή εξόδου μπορεί να έχει παραπάνω τιμές, όπως είναι ένα διάνυσμα, ενώ στην πράξη οι υλοποιήσεις του backpropagation υποστηρίζουν λειτουργίες με πολλαπλές εξόδους. Εάν μία μεταβλητή y υπολογίζεται με την εφαρμογή μιας λειτουργίας σε μία μεταβλητή x, τότε αυτό στο γράφο απεικονίζεται με μία κατευθυνόμενη ακμή από το x στο y.

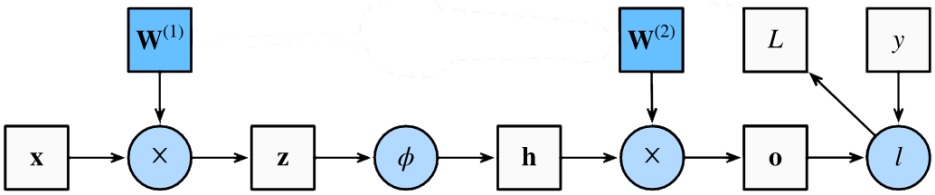
Ο αλυσιδωτός κανόνας του διαφορικού λογισμού (chain rule of calculus) χρησιμοποιείται για τον υπολογισμό των παραγώγων συναρτήσεων που προκύπτουν από τη σύνθεση άλλων συναρτήσεων, που τα παράγωγά τους είναι γνωστά. Ο αλγόριθμος backpropagation είναι ένας αλγόριθμος που υπολογίζει αυτόν τον αλυσιδωτό κανόνα, με μία συγκεκριμένη σειρά υπολογισμών που είναι εξαιρετικά αποδοτική. Έστω ότι x είναι ένα πραγματικός αριθμός και f και g είναι συναρτήσεις που αντιστοιχίζουν έναν πραγματικό αριθμό σε έναν άλλον πραγματικό αριθμό. Θεωρούμε επίσης ότι $y = g(x)$ και $z = f(g(x)) = f(y)$. Τότε με βάση τον αλυσιδωτό κανόνα έχουμε:

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}.$$

Ο παραπάνω τύπος μπορεί να γενικευτεί. Θεωρούμε ότι $x \in \mathbb{R}^m, y \in \mathbb{R}^n$, η συνάρτηση g αντιστοιχεί από $\mathbb{R}^m$ σε $\mathbb{R}^n$, και η συνάρτηση f αντιστοιχεί από $\mathbb{R}^n$ σε $\mathbb{R}$. Εάν $y = g(x)$ και $z = f(y)$, τότε [38]:

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i}.$$

Για παράδειγμα, θεωρούμε τον ακόλουθο υπολογιστικό γράφο:



Εικόνα 4: Υπολογιστικός γράφος μιας προς τα εμπρός διάδοσης (forward propagation) [39].

Οπότε, για την είσοδο $x \in \mathbb{R}^d$ και άμα θεωρήσουμε ότι δεν έχουμε μεταβλητή μεροληψίας (bias), τότε με βάση τον παραπάνω γράφο έχουμε:

$$z = W^{(1)}x,$$

$$h = \phi(z),$$

όπου $W^{(1)} \in \mathbb{R}^{hxd}$ είναι το βάρος του κρυφού επιπέδου και ϕ είναι η συνάρτηση ενεργοποίησης. Η κρυφή μεταβλητή h είναι μία ενδιάμεση μεταβλητή, που είναι ένα δυάνυσμα μήκους h. Θεωρούμε ως $W(2) \in \mathbb{R}^{hxd}$ το βάρος του επιπέδου εξόδου, ως l τη loss function και ως y την πραγματική ετικέτα του παραδείγματος. Οπότε έχουμε:

$$o = W^{(2)}h,$$

$$L = l(o, y).$$

Ο στόχος του backpropagation είναι ο υπολογισμός των κλίσεων $\partial L/\partial W^{(1)}$ και $\partial L/\partial W^{(2)}$. Για το σκοπό αυτό, εφαρμόζουμε τον αλυσιδωτό κανόνα και υπολογίζουμε την κλίση κάθε ενδιάμεσης μεταβλητής και παραμέτρου, δηλαδή την αλλαγή που πρέπει να υποστεί η αντίστοιχη μεταβλητή ώστε η έξοδος του νευρωνικού δικτύου να πλησιάσει την επιθυμητή. Η σειρά των υπολογισμών είναι η

ανάστροφη από τους υπολογισμούς που έγιναν κατά την προς τα εμπρός διάδοση, καθώς αρχίζουμε από το αποτέλεσμα του υπολογιστικού γράφου και κατευθυνόμαστε προς τις παραμέτρους. Το πρώτο βήμα είναι ο υπολογισμός της κλίσης της συνάρτησης κόστους ως προς την μεταβλητή εξόδου o του επιπέδου εξόδου, $\frac{\partial L}{\partial o}$. Ύστερα, ακολουθεί ο υπολογισμός των παραμέτρων που είναι πιο κοντά στο επίπεδο εξόδου. Δηλαδή:

$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial W^{(2)}} = \frac{\partial L}{\partial o}h^T.$$

Για να πάρουμε την κλίση ως προς τη μεταβλητή $W^{(1)}$, πρέπει να συνεχιστεί το backpropagation από το επίπεδο εξόδου προς το κρυφό επίπεδο. Η κλίση ως προς την έξοδο του κρυφού επιπέδου δίνεται από τον ακόλουθο τύπο:

$$\frac{\partial L}{\partial h} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial h} = W^{(2)^T}\frac{\partial L}{\partial o}.$$

Εφόσον η συνάρτηση ενεργοποίησης ϕ εφαρμόζεται ανά στοιχείο, δηλαδή κάθε στοιχείο (i, j) του νέου πίνακα είναι το αποτέλεσμα του γινομένου των στοιχείων (i, j) των δύο αρχικών πινάκων, χρησιμοποιείται το σύμβολο $\odot$:

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial h}\frac{\partial h}{\partial z} = \frac{\partial L}{\partial h} \odot \phi'(z).$$

Τελικά, υπολογίζουμε την κλίση $\frac{\partial L}{\partial W^{(1)}}$. Σύμφωνα με τον αλυσιδωτό κανόνα έχουμε:

$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial W^{(1)}} = \frac{\partial L}{\partial z}x^T.$$

Κατά την εκπαίδευση νευρωνικών δικτύων, η προς τα εμπρός και η οπισθοδρομική διάδοση εξαρτώνται η μία από την άλλη. Συγκεκριμένα, στην προς τα εμπρός διάδοση διασχίζουμε τον υπολογιστικό γράφο στην κατεύθυνση των εξαρτήσεων και υπολογίζουμε όλες τις μεταβλητές από τις οποίες περνάμε. Στη συνέχεια, αυτές χρησιμοποιούνται στην οπισθοδρομική διάδοση, όπου η κατεύθυνση των υπολογισμών στον γράφο αντιστρέφεται. Κατά τη διάρκεια της εκπαίδευσης, λοιπόν, γίνεται μία εναλλαγή σε αυτές τις δύο διαδόσεις, ενημερώνοντας τελικά τις παραμέτρους του μοντέλου με βάση τις κλίσεις που υπολογίστηκαν στην οπισθοδρομική διάδοση. Το backpropagation επαναχρησιμοποιεί τις αποθηκευμένες ενδιάμεσες τιμές από την προς τα εμπρός διάδοση για την αποφυγή του να γίνουν οι υπολογισμοί δυο φορές. Συνεπώς, οι ενδιάμεσες τιμές πρέπει να να μείνουν στη μνήμη μέχρι να ολοκληρωθεί το backpropagation, ενώ αυτός είναι ένας από τους λόγους που η εκπαίδευση απαιτεί περισσότερη μνήμη από την απλή πρόβλεψη τιμών. Επιπλέον, το μέγεθος αυτών των ενδιάμεσων τιμών είναι ανάλογο του αριθμού των επιπέδων του νευρωνικού δικτύου και του μεγέθους του batch, που είναι ο αριθμός των παρατηρήσεων για τις οποίες γίνεται η προς τα εμπρός διάδοση, πριν γίνει η ενημέρωση των παραμέτρων του μοντέλου. Επομένως, η εκπαίδευση βαθιών δικτύων που συνοδεύεται με τη χρήση μεγάλου μεγέθους του batch είναι πιθανο να οδηγήσει σε σφάλματα μνήμης [39].

3.4 Τεχνικές Ομαλοποίησης

Ένα κεντρικό πρόβλημα στη μηχανική μάθηση είναι ένας αλγόριθμος να μπορεί να έχει καλά αποτελέσματα όχι μόνο στα δεδομένα εκπαίδευσης, αλλά και σε νέα δεδομένα, δηλαδή να μπορεί να γενικεύει. Η ικανότητα γενίκευσης ενός μοντέλου εκτιμάται από τις επιδόσεις του στα δεδομένα επαλήθευσης. Πριν την εκπαίδευση τα δεδομένα διαχωρίζονται σε δεδομένα εκπαίδευσης και δεδομένα επαλήθευσης με τυχαίο τρόπο. Κατά την εκπαίδευση χρησιμοποιούνται τα δεδομένα εκπαίδευσης με σκοπό να ενημερωθούν οι παράμετροι ώστε να μειωθεί το σφάλμα στα δεδομένα εκπαίδευσης. Στη συνέχεια, εξετάζεται η απόδοση του μοντέλου στα δεδομένα επαλήθευσης. Το σφάλμα στα δεδομένα επαλήθευσης αναμένεται να είναι μεγαλύτερο ή ίσο με το σφάλμα στα δεδομένα εκπαίδευσης. Οι παράγοντες που καθορίζουν το πόσο καλά ένας αλγόριθμος μηχανικής μάθησης γενικεύει είναι η ικανότητά του

να έχει μικρό σφάλμα στα δεδομένα εκπαίδευσης και η ικανότητά του να έχει μικρή διαφορά μεταξύ των σφαλμάτων των δύο ομάδων δεδομένων. Αυτοί οι δύο παράγοντες αντιστοιχούν σε δύο κεντρικές προκλήσεις στη μηχανική μάθηση, η υποπροσαρμογή (underfitting) και η υπερπροσαρμογή (overfitting). Η υποπροσαρμογή συμβαίνει όταν το μοντέλο δεν μπορεί να έχει ένα αρκετά χαμηλό σφάλμα στα δεδομένα εκπαίδευσης, ενώ η υπερπροσαρμογή συμβαίνει όταν η διαφορά μεταξύ του σφάλματος στα δεδομένα εκπαίδευσης και του σφάλματος στα δεδομένα επαλήθευσης είναι πολύ μεγάλη.

Πολλές στρατηγικές που χρησιμοποιούνται στη μηχανική μάθηση έχουν σχεδιαστεί αποκλειστικά για τη μείωση του σφάλματος στα δεδομένα επαλήθευσης, πιθανώς σε βάρος του σφάλματος στα δεδομένα εκπαίδευσης. Αυτές οι στρατηγικές είναι γνωστές ως τεχνικές ομαλοποίησης (regularization). Μία αποδοτική τεχνική ομαλοποίησης μειώνει σημαντικά τη διακύμανση (variance) χωρίς να αυξήσει πολύ τη μεροληψία (bias). Η μεροληψία είναι η διαφορά ανάμεσα στην πρόβλεψη του μοντέλου και τη σωστή τιμή που προσπαθεί να προβλέψει. Ένα μοντέλο με πολύ υψηλή μεροληψία δίνει λίγη προσοχή στα δεδομένα εκπαίδευσης και υπερ-απλουστεύει το μοντέλο, κάτι που οδηγεί πάντα σε υψηλό σφάλμα στα δεδομένα εκπαίδευσης και επαλήθευσης. Η διακύμανση από την άλλη πλευρά είναι όταν το μοντέλο λαμβάνει υπόψιν τις διακυμάνσεις στα δεδομένα εκπαίδευσης. Εάν το μοντέλο έχει υψηλό variance τότε το μοντέλο μαθαίνει πολύ καλά τα δεδομένα εκπαιδευσης και συνεπώς δεν μπορεί να γενικεύσει σε δεδομένα που δεν έχει ξαναδεί. Σε αυτήν την ενότητα αναλύουμε τους δύο μηχανισμούς ομαλοποίησης που χρησιμοποιήσαμε στα μοντέλα μας [38].

3.4.1 Πρόωρος Τερματισμός

Πολύ συχνά κατά την εκπαίδευση μοντέλων, παρατηρείται ότι το σφάλμα στα δεδομένα εκπαίδευσης μειώνεται σταθερά, ενώ το σφάλμα στα δεδομένα επαλήθευσης αρχίζει να αυξάνεται μετά από κάποια εποχή. Αυτό σημαίνει ότι μπορούμε να αποκτήσουμε ένα μοντέλο με καλύτερο σφάλμα στα δεδομένα επαλήθευσης επιστρέφοντας στις παραμέτρους που είχε το μοντέλο τη στιγμή που παρουσίασε το χαμηλότερο σφάλμα στα δεδομένα επαλήθευσης. Κάθε φορά που το σφάλμα στα δεδομένα επαλήθευσης βελτιώνεται, αποθηκεύουμε ένα αντίγραφο των παραμέτρων του μοντέλου. Όταν η εκπαίδευση ολοκληρωθεί επιστρέφουμε το μοντέλο σε αυτές τις παραμέτρους που αποθηκεύσαμε. Ο αλγόριθμος τερματίζει όταν το μοντέλο δεν έχει βελτιώσει το σφάλμα στα δεδομένα επαλήθευσης για έναν προκαθορισμένο αριθμό εποχών. Σημειώνουμε ότι μία εποχή αντιστοιχεί σε ένα πέρασμα όλων των δεδομένων κατά τη διάρκεια της εκπαίδευσης. Η τεχνική που περιγράψαμε ονομάζεται πρόωρος τερματισμός (early stopping) και είναι μία από τις πιο διαδεδομένες τεχνικές ομαλοποίησης, καθώς είναι αποτελεσματική και εύκολη στην υλοποίηση.

Με τον πρόωρο τερματισμό ο αριθμός των εποχών λειτουργεί ως μία υπερ-παράμετρος. Το μόνο σημαντικό κόστος επιλογής αυτής της υπερ-παραμέτρου είναι ότι πρέπει να γίνεται εκτίμηση του σφάλματος στα δεδομένα επαλήθευσης αρκετά συχνά κατά τη διάρκεια της εκπαίδευσης. Ιδανικά, αυτός ο υπολογισμός γίνεται παράλληλα με τη διαδικασία της εκπαίδευσης σε μία ξεχωριστή CPU ή GPU. Εάν αυτοί οι πόροι δεν είναι διαθέσιμοι τότε το κόστος αυτών των περιοδικών υπολογισμών μπορεί να μειωθεί με τη χρήση ενός συνόλου δεδομένων επαλήθευσης που το μέγεθος του είναι μικρό σε σχέση με το σύνολο δεδομένων εκπαίδευσης, ή με το να γίνεται πιο αραιά η εκτίμηση του σφάλματος στα δεδομένα επαλήθευσης. Επίσης, ένα άλλο κόστος αυτής της τεχνικής είναι η αποθήκευση των καλύτερων παραμέτρων. Το κόστος αυτό είναι συνήθως αμελητέο, καθώς αυτές οι παράμετροι μπορούν να αποθηκευτούν σε μία μεγαλύτερη μνήμη. Καθώς αυτές οι παράμετροι δεν διαβάζονται ποτέ κατά τη διάρκεια της εκπαίδευσης, οι περιστασιακές αποθηκεύσεις αυτών των μεταβλητών έχουν μικρή επίδραση στον συνολικό χρόνο εκπαίδευσης.

Ο πρόωρος τερματισμός είναι μία τεχνική που δεν απαιτεί σχεδόν καμία αλλαγή στη διαδικασία της εκπαίδευσης, στη συνάρτηση κόστους ή στο σύνολο των επιτρεπτών τιμών των παραμέτρων. Αυτό σημαίνει ότι είναι εύκολη η χρήση αυτής της τεχνικής χωρίς να αλλάξουν οι δυναμικές της εκπαίδευσης.

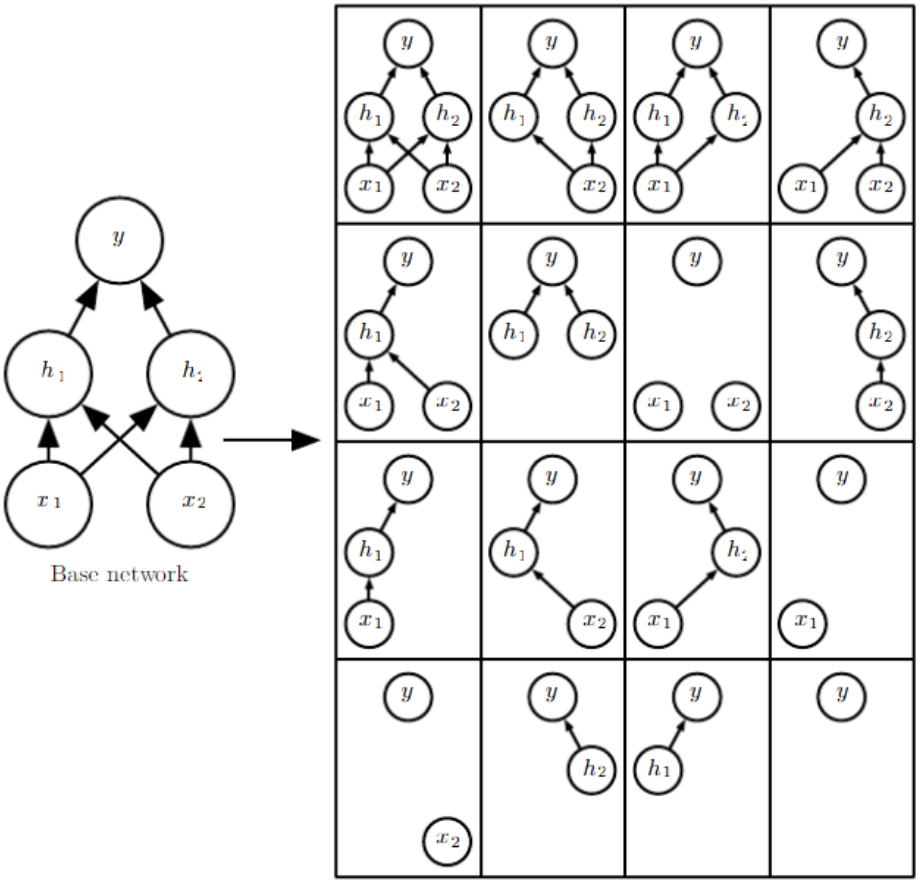
Επιπλέον, η τεχνική αυτή μπορεί να χρησιμοποιηθεί είτε μόνη της είτε σε συνδυασμό με άλλες τεχνικές ομαλαποίησης. Επίσης, για αυτήν την τεχνική είναι απαραίτητη η ύπαρξη δεδομένων επαλήθευσης. Τέλος, ο πρόωρος τερματισμός έχει το πλεονέκτημα σε σχέση με άλλες τεχνικές ομαλοποίησης ότι καθορίζει αυτόματα τη σωστή ποσότητα ομαλοποίησης χωρίς να χρειαστούν πολλά πειράματα εκπαίδευσης για τον καθορισμό της υπερ-παραμέτρου [38].

3.4.2 Dropout

Το dropout αποτελεί μία υπολογιστικά φθηνή αλλά ισχυρή μέθοδο ομαλοποίησης. Εκ πρώτης όψεως η τεχνική αυτή μοιάζει με την τεχνική bagging. Η τεχνική αυτή περιλαμβάνει την εκπαίδευση πολλών μοντέλων και την αξιολόγηση πολλών μοντέλων σε κάθε δεδομένο επαλήθευσης. Αυτός όμως ο τρόπος ομαλοποίησης δεν είναι πρακτικός όταν το κάθε μοντέλο είναι ένα μεγάλο νευρωνικό δίκτυο, καθώς η εκπαίδευση και η αξιολόγηση τέτοιων δικτύων είναι δαπανηρές σε χρόνο και μνήμη. Το dropout παρέχει μία φθηνή προσέγγιση στην εκπαίδευση και στην αξιολόγηση πολλών νευρωνικών δικτύων.

Συγκεκριμένα, το dropout εκπαιδεύει ένα σύνολο από υπο-δίκτυα που δημιουργούνται από την αφαίρεση κόμβων, που δεν είναι κόμβοι εξόδου, από το αρχικό δίκτυο, όπως φαίνεται στην εικόνα 5. Στα μοντέρνα νευρωνικά δίκτυα, ένας κόμβος μπορεί να αφαιρεθεί αποτελεσματικά από ένα δίκτυο πολλαπλασιάζοντας την έξοδό του με μηδέν. Κατά την εκπαίδευση, κάθε φορά που φορτώνεται μία παρατήρηση γίνεται μία τυχαία επιλογή δυαδικής μάσκας (mask) που εφαρμόζεται σε όλους του κόμβους εισόδου και τους κρυφούς κόμβους. Η δυαδική μάσκα δηλώνει ποιοι κόμβοι θα αφαιρεθούν. Η μάσκα για κάθε νευρώνα επιλέγεται ανεξάρτητητα από τις υπόλοιπες. Η πιθανότητα επιλογής τιμής ένα για τη μάσκα, που σημαίνει ότι ο αντίστοιχος κόμβος θα συμπεριληφθεί, είναι μία υπερ-παράμετρος που καθορίζεται πριν την αρχή της εκπαίδευσης. Στη συνέχεια, γίνονται κανονικά η προς τα εμπρός και η οπισθοδρομική διάδοση.

Η τεχνική dropout έχει πολλές ομοιότητες με τον αλγόριθμο bagging. Το bagging (bootstrap aggregating) είναι μία τεχνική ομαλοποίησης που συνδυάζει πολλά διαφορετικά μοντέλα. Η ιδέα είναι ότι εκπαιδεύονται πολλά διαφορετικά μοντέλα ξεχωριστά και ύστερα όλα τα μοντέλα ψηφίζουν για την έξοδο των δεδομένων επαλήθευσης. Ο λόγος που αυτή η στρατηγική δουλεύει είναι επειδή τα διαφορετικά μοντέλα συνήθως δεν κάνουν τα ίδια λάθη στα δεδομένα επαλήθευσης και επομένως το συνδυασμένο αποτέλεσμα θα είναι καλύτερο από το καθένα μοντέλο ξεχωριστά. Το bagging είναι μία μέθοδος που επιτρέπει το ίδιο μοντέλο να χρησιμοποιηθεί αρκετές φορές. Συγκεκριμένα, το bagging περιλαμβάνει την κατασκευή k διαφορετικών συνόλων δεδομένων. Το κάθε σύνολο δεδομένων θα έχει τον ίδιο αριθμό παρατηρήσεων με το αρχικό σύνολο δεδομένων, αλλά το κάθε νέο σύνολο δεδομένων κατασκευάζεται με δειγματοληψία με επανατοποθέτηση. Αυτό σημαίνει ότι σε κάθε νέο σύνολο δεδομένων απουσιάζουν μερικές παρατηρήσεις από το αρχικό σύνολο δεδομένων, ενώ μερικές παρατηρήσεις υπάρχουν παραπάνω από μία φορές. Στη συνέχεια, το μοντέλο i εκπαιδεύεται στο i νέο σύνολο δεδομένων. Οι διαφορές στις παρατηρήσεις που συμπεριλαμβάνονται σε κάθε σύνολο δεδομένων έχουν ως αποτέλεσμα να υπάρχουν διαφορές στα εκπαιδευόμενα μοντέλα.



Εικόνα 5: Τα πιθανά υπο-δίκτυα ενός αρχικού δικτύου που δημιουργούνται από την αφαίρεση κόμβων στο dropout [38].

Η εκπαίδευση με το dropout διαφέρει από την εκπαίδευση με την τεχνική bagging, καθώς σε αυτή την περίπτωση όλα τα μοντέλα είναι ανεξάρτητα. Στην περίπτωσση του dropout, τα μοντέλα μοιράζονται παραμέτρους, με το κάθε μοντέλο να κληρονομεί ένα διαφορετικό υποσύνολο παραμέτρων από το αρχικό νευρωνικό δίκτυο. Το γεγονός ότι μοιράζονται οι παράμετροι καθιστά δυνατή την αναπαράσταση ενός εκθετικού αριθμού μοντέλων χωρίς υπερβολικό κόστος μνήμης. Τυπικά, τα περισσότερα μοντέλα δεν εκπαιδεύονται καν, ενώ συνήθως το μοντέλο είναι αρκετά μεγάλο που είναι ακατόρθωτο να εκπαιδευτούν όλα τα υπο-δίκτυα μέσα στη διάρκεια ζωής του σύμπαντος. Αντιθέτως, ένα ελάχιστο ποσοστό όλων των πιθανών υπο-δικτύων εκπαιδεύονται σε κάθε βήμα. Πέρα από αυτές τις διαφορές, η τεχνική dropout ακολουθεί τον αλγόριθμο bagging. Δηλαδή, τα δεδομένα εκπαίδευσης που χρησιμοποιούνται για κάθε υπο-δίκτυο είναι υποσύνολα των αρχικών δεδομένων εκπαίδευσης που έχουν προκύψει με δειγματοληψία με επανατοποθέτηση.

Ένα πλεονέκτημα του dropout είναι ότι είναι υπολογιστικά φθηνό. Η χρήση του dropout κατά την εκπαίδευση απαιτεί $O(n)$ υπολογισμούς ανά παρατήρηση, ανά ενημέρωση παραμέτρων και για να παραγάγει n τυχαίους δυαδικούς αριθμούς και να τους πολλαπλασιάσει με τον κάθε νευρώνα. Αναλόγως την εφαρμογή, μπορεί να χρειάζεται και $O(n)$ μνήμη για την αποθήκευση του διανύσματος της μάσκας μέχρι το στάδιο του backpropagation. Ένα άλλο σημαντικό πλεονέκτημα του dropout είναι ότι δεν θέτει κάποιον περιορισμό για τον τύπο του μοντέλου ή τη διαδικασία εκπαίδευσης που μπορεί να χρησιμοποιηθεί. Ωστόσο, για πολύ μεγάλα σύνολα δεδομένων, η ομαλοποίηση προσφέρει λίγη μείωση στο σφάλμα γενίκευσης. Σε αυτές τις περιπτώσεις, το υπολογιστικό κόστος χρήσης dropout μπορεί

να υπερβαίνει τα πλεονεκτήματα. Τέλος, στη μη επιβλεπόμενη μάθηση, δηλαδή όπου τα δεδομένα εκπαίδευσης δεν ανήκουν σε κάποια κλάση, το dropout δεν είναι πολύ αποτελεσματικό [38].

3.5 Αλγόριθμοι Βελτιστοποίησης

Οι αλγόριθμοι βελτιστοποίησης είναι εργαλεία που επιτρέπουν την ενημέρωση των παραμέτρων ενός μοντέλου και την ελαχιστοποίηση της τιμής της συνάρτησης κόστους. Οι αλγόριθμοι αυτοί είναι σημαντικοί για τη βαθιά μηχανική μάθηση. Από τη μία πλευρά, η εκπαίδευση ενός πολύπλοκου μοντέλου βαθιάς μάθησης μπορεί να διαρκέσει ώρες, μέρες ή ακόμα και εβδομάδες. Η επίδοση του αλγορίθμου βελτιστοποίησης μπορεί άμεσα να επηρεάσει την αποτελεσματικότητα της εκπαίδευσης του μοντέλου. Από την άλλη πλευρά, η κατανόηση των αρχών των διαφορετικών αλγορίθμων βελτιστοποίησης και των υπερ-παραμέτρων τους μας δίνει τη δυνατότητα να ρυθμίσουμε κατάλληλα τις υπερ-παραμέτρους ώστε να βελτιωθεί η επίδοση των μοντέλων βαθιάς μηχανικής μάθησης.

Παρ'όλο που η βελτιστοποίηση παρέχει έναν τρόπο ελαχιστοποίησης της συνάρτησης κόστους για τη βαθιά μηχανική μάθηση, οι στόχοι της βελτιστοποίησης και της βαθιάς μάθησης είναι διαφορετικοί. Η πρώτη ασχολείται με την ελαχιστοποίηση του κόστους ενώ η δεύτερη με την εύρεση ενός κατάλληλου μοντέλου. Δηλαδή, ο στόχος της βελτιστοποίησης είναι η μείωση του σφάλματος εκπαίδευσης, ενώ ο στόχος της βαθιάς μάθησης είναι η μείωση του σφάλματος γενίκευσης. Συνεπώς, για να επιτευχθούν και οι δύο στόχοι πρέπει πέρα από τη χρήση ενός καλού αλγορίθμου βελτιστοποίησης, να μετριάζεται η υπερπροσαρμογή (overfitting). Η ελαχιστοποίηση του σφάλματος εκπαίδευσης δεν εγγυάται ότι θα βρεθεί το καλύτερο σύνολο παραμέτρων για την ελαχιστοποίηση του σφάλματος γενίκευσης. Επίσης, όπως είδαμε και στην ενότητα 3.2, μία συνάρτηση κόστους μπορεί να έχει πολλά τοπικά ελάχιστα, ενώ το πρόβλημα των vanishing gradients, που εξηγήσαμε στην ενότητα 3.1, μπορεί να εμποδίσει τη βελτιστοποίηση. Επομένως, δεν είναι απαραίτητη η εύρεση της βέλτιστης λύσης, καθώς και οι κατά προσέγγιση λύσεις είναι πολύ χρήσιμες [39].

Η βασική υπερ-παράμετρος των αλγορίθμων βελτιστοποίησης είναι ο ρυθμός μάθησης (learning rate). Η παράμετρος αυτή ελέγχει το πόσο αλλάζει το μοντέλο με βάση το εκτιμώμενο σφάλμα, κάθε φορά που ενημερώνονται τα βάρη του μοντέλου. Η επιλογή του ρυθμού μάθησης είναι δύσκολη, καθώς μία μικρή τιμή μπορεί να καταλήξει σε μία μακροπρόθεσμη διαδικασία εκπαίδευσης που δεν είναι απαραίτητο ότι θα έχει καλό αποτέλεσμα, ενώ μία μεγάλη τιμή μπορεί να έχει ως αποτέλεσμα το μοντέλο να μάθει πολύ γρήγορα ένα σύνολο βαρών που να μην είναι το βέλτιστο ή να καταλήξει σε μία ασταθή διαδικασία εκπαίδευσης [1].

Μερικές φορές ο ρυθμός μάθησης η αντικαθιστάται από έναν ρυθμό μάθησης που εξαρτάται από τον χρόνο $\eta(t)$. Σε αυτήν την περίπτωση, πρέπει να καθορίσουμε πόσο γρήγορα το η πρέπει να μειώνεται. Αν η μείωση είναι πολύ γρήγορη τότε η βελτιστοποίηση τερματίζεται πρόωρα, ενώ αν η μείωση είναι πολύ αργή τότε σπαταλάται πολύς χρόνος στη βελτιστοποίηση [39].

3.5.1 Adagrad

Για να έχουμε καλή ακρίβεια από την εκπαίδευση ενός μοντέλου, τυπικά πρέπει ο ρυθμός μάθησης να μειώνεται όπως προχωράει η εκπαίδευση συνήθως με ένα ποσοστό $O(t^{-\frac{1}{2}})$ ή μικρότερο. Η ενημέρωση των παραμέτρων του μοντέλου που συνδέονται με χαρακτηριστικά των δεδομένων που δεν συναντώνται συχνά έχει νόημα να συμβαίνει μόνο όταν συναντώνται αυτά τα χαρακτηριστικά. Δεδομένου ενός ρυθμού μάθησης που μειώνεται, είναι πιθανό να καταλήξουμε σε μία κατάσταση όπου οι παράμετροι για τα συχνά χαρακτηριστικά συγκλίνουν πολύ γρήγορα στις βέλτιστες τιμές τους, ενώ οι παράμετροι για τα μη συχνά χαρακτηριστικά δεν έχουν παρατηρηθεί επαρκώς για τον καθορισμό των βέλτιστων

[1] https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

τιμών τους. Με λίγα λόγια, ο ρυθμός μάθησης μειώνεται είτε πολύ αργά για τα συχνά χαρακτηριστικά είτε πολύ γρήγορα για τα μη συχνά χαρακτηριστικά. Ένας τρόπος διόρθωσης αυτού του προβλήματος είναι η μέτρηση των φορών που συναντάμε ένα χαρακτηριστικό και η χρήση αυτής της μέτρησης για την προσαρμογή του ρυθμού μάθησης. Δηλαδή, μπορούμε να χρησιμοποιήσουμε:

$$\eta_i = \frac{\eta_0}{\sqrt{s(i,t) + \epsilon}},$$

όπου $s(i,t)$ είναι οι φορές που έχει παρατηρηθεί το χαρακτηριστικό i μέχρι τη χρονική στιγμή t. Ωστόσο, αυτή η λύση αποτυγχάνει όταν οι κλίσεις (gradients) έχουν συνήθως πολύ μικρές τιμές και σπάνια μεγάλες τιμές.

Ο αλγόριθμος Adagrad απευθύνται σε αυτό το πρόβλημα αντικαθιστώντας τον όρο $s(i,t)$ με μία συνάθροιση των τετραγώνων από κλίσεις που έχουν παρατηρηθεί προηγουμένως. Συγκεκριμένα, χρησιμοποιεί:

$$s(i, t+1) = s(i,t) + (\partial_i f(x))^2,$$

ως έναν τρόπο για να προσαρμόσει τον ρυθμό μάθησης. Αυτή η τακτική έχει δύο πλεονεκτήματα: α) δεν χρειάζεται πλέον να αποφασίσουμε εάν μία κλίση είναι αρκετά μεγάλη και β) κλιμακώνεται αυτόματα με το μέγεθος των κλίσεων. Έτσι, γίνονται μικρότερες αλλαγές στις παραμέτρους για τα συχνά χαρακτηριστικά και μεγαλύτερες αλλαγές στις παραμέτρους για τα μη συχνά χαρακτηριστικά. Ο αλγόριθμος αυτός τροποποιεί τον ρυθμό μάθησης η σε κάθε χρονικό βήμα t για κάθε παράμετρο w_i με βάση τις προηγούμενες κλίσεις g_t της w_t:

$$g_t = \partial_w l(y_t, f(x_t, w)),$$

$$s_t = s_{t-1} + g_t^2,$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{s_t + \epsilon}} g_t,$$

όπου ϵ είναι μία σταθερά που διασφαλίζει ότι δε θα γίνει διαίρεση με το 0, ενώ η αρχική αρχικοποίηση είναι $s_0 = 0$. Ωστόσο, σε προβλήματα βαθιάς μηχανικής μάθησης, ο αλγόριθμος Adagrad μερικές φορές μπορεί να είναι αρκετά επιθετικός στη μείωση του ρυθμού μάθησης, ενώ μπορεί να χρειάζεται η μείωση αυτή να γίνεται με πιο αργούς ρυθμούς [39].

3.5.2 RMSProp

Ο αλγόριθμος RMSProp είναι παρόμοιος με τον Adagrad, καθώς και οι δύο χρησιμοποιούν το τετράγωνο της κλίσης για την ενημέρωση των παραμέτρων. Ωστόσο, ο Adagrad συγκεντρώνει τα τετράγωνα των κλίσεων g_t σε ένα διάνυσμα s_t, με αποτέλεσμα το s_t να μεγαλώνει χωρίς όριο, λόγω έλλειψης κανονικοποίησης, όπως συγκλίνει ο αλγόριθμος. Ένας τρόπος διόρθωσης αυτού του προβλήματος είναι η χρήση του leaky average:

$$v_t = \beta v_{t-1} + g_{t,t-1} = \sum_{\tau=0}^{t-1} \beta^\tau g_{t-\tau,t-\tau-1},$$

για κάποιο $\beta \in (0,1)$. Σε αυτή την τεχνική η κλίση αντικαθιστάται από το v, που ονομάζεται momentum, που είναι ο μέσος όρος πολλών παρελθοντικών κλίσεων. Ένα μεγάλο β σημαίνει ότι η κλίση επηρεάζεται από πολλές παρελθοντικές κλίσεις, ενώ ένα μικρό β κάνει μία ελάχιστη διόρθωση στην κλίση. Ο τρόπος που λειτουργεί με το leaky average ο αλγόριθμος RMSProp είναι ο ακόλουθος:

$$s_t = \gamma s_{t-1} + (1-\gamma) g_t^2 = (1-\gamma)(g_t^2 + \gamma g_{t-1}^2 + \gamma^2 g_{t-2} + ...),$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{s_t + \epsilon}} \odot g_t.$$

Η σταθερά $\epsilon > 0$ τυπικά έχει τιμή 10^{-6} και διασφαλίζει όπως και πριν ότι δε θα γίνει διαίρεση με το 0, ενώ ισχύει $\gamma > 0$. Ο όρος γ καθορίζει κατά πόσο τα βάρη επηρεάζονται από παρελθοντικές τιμές [39].

3.5.3 Adam

Ο αλγόριθμος Adam συνδυάζει χαρακτηριστικά από πολλούς αλγορίθμους βελτιστοποίησης, με αποτέλεσμα να έχει γίνει από τους πιο δημοφιλείς και αποτελεσματικούς αλγορίθμους βελτιστοποίησης. Ο Adam χρησιμοποιεί επίσης το leaky average. Άρα έχουμε:

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t,$$

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) g_t^2.$$

Τα β_1 και β_2 είναι μη αρνητικές παράμετροι. Συνηθισμένες τιμές για αυτές τις παραμέτρους είναι $\beta_1 = 0.9$ και $\beta_2 = 0.999$. Οι κανονικοποιημένες μεταβλητές δίνονται από τους παρακάτω τύπους:

$$\hat{v}_t = \frac{v_t}{1 - \beta_1^t} \quad \kappa\alpha\iota \quad \hat{s}_t = \frac{s_t}{1 - \beta_2^t}.$$

Ο τύπος για την κλίση είναι ο ακόλουθος:

$$g_t = \frac{\eta \hat{v}_t}{\sqrt{\hat{s}_t} + \epsilon}.$$

Αντίθετα από τον RMSProp, ο Adam χρησιμοποιεί το momentum $\hat{v}_t$ αντί της κλίσης για τις ενημερώσεις των παραμέτρων. Οι ενημερώσεις των βαρών γίνονται με τον ακόλουθο απλό τρόπο [39]:

$$w_t = w_{t-1} - g_t.$$

3.6 Βελτιστοποίηση Υπερ-παραμέτρων

Οι παράμετροι που ρυθμίζονται αλλά δεν ενημερώνονται κατά τη διάρκεια της εκπαίδευσης ονομάζονται υπερ-παράμετροι. Η βελτιστοποίηση υπερ-παραμέτρων είναι η διαδικασία κατά την οποία οι υπερ-παράμετροι επιλέγονται με βάση τα αποτελέσματα της εκπαίδευσης, αφού γίνει εκτίμηση στα δεδομένα επαλήθευσης [39]. Για τη βελτιστοποίηση των υπερ-παραμέτρων χρησιμοποιήσαμε τη βιβλιοθήκη Optuna, που εκτελεί αυτόματα την αναζήτηση τιμών στις υπερ-παραμέτρους σαν μαύρο κουτί [2]. Η βιβλιοθήκη Optuna χρησιμοποιεί Tree-structured Parzen Estimator (TPE), που είναι μία μορφή Μπεϋζιανής Βελτιστοποίησης (Bayesian Optimization), για να αναζητήσει πιο αποτελεσματικά τιμές για τις υπερ-παραμέτρους.

Η Μπεϋζιανή βελτιστοποίηση κρατάει τα προηγούμενα αποτελέσματα, τα οποία τα χρησιμοποιεί για να σχηματίσει ένα πιθανοτικό μοντέλο που αντιστοιχεί τις υπερ-παραμέτρους σε μία πιθανότητα ακρίβειας στη συνάρτηση κόστους. Στη βιβλιογραφία το μοντέλο αυτό αναπαρίσταται ως $p(y|x)$. Οι Μπεϋζιανές μέθοδοι λειτουργούν βρίσκοντας το επόμενο σύνολο υπερ-παραμέτρων για την εκτίμηση της συνάρτησης κόστους, επιλέγοντας τιμές για τις υπερ-παραμέτρους που δίνουν καλύτερα αποτελέσματα στη συνάρτηση $p(y|x)$. Δηλαδή τα βήματα αυτής της μεθόδου είναι τα ακόλουθα:

1. Δημιουργία ενός πιθανοτικού μοντέλου της συνάρτησης κόστους.

2. Εύρεση των υπερ-παραμέτρων που αποδίδουν καλύτερα σε αυτό το μοντέλο.

[2] https://optuna.org/

3. Εφαρμογή αυτών των υπερ-παραμέτρων στη συνάρτηση κόστους.

4. Ενημέρωση του πιθανοτικού μοντέλου ενσωματώνοντας τα νέα αποτελέσματα.

5. Επανάληψη των βημάτων 2-4 μέχρι το μέγιστο αριθμό επαναλήψεων ή το μέγιστο χρόνο εκτέλεσης.

Ο στόχος της Μπεϋζιανής λογικής είναι να βρίσκει "λιγότερο λάθος" τιμές για τις υπερ-παραμέτρους, ενημερώνοντας συνεχώς το πιθανοτικό μοντέλο μετά από κάθε εκτίμηση της συνάρτησης κόστους για ένα σύνολο υπερ-παραμέτρων. Εκτιμώντας υπερ-παραμέτρους που φαίνονται πολλά υποσχόμενες από προηγούμενα αποτελέσματα, η Μπεϋζιανή μέθοδος μπορεί να βρει καλύτερες τιμές από την τυχαία αναζήτηση (random search) και την αναζήτηση πλέγματος (grid search) σε λιγότερες επαναλήψεις, καθώς οι δύο αυτές μέθοδοι ψάχνουν σε όλο το εύρος των τιμών των υπερ-παραμέτρων ακόμα και αν είναι ξεκάθαρο ότι η βέλτιστη επιλογή βρίσκεται πιθανώς σε ένα μικρό εύρος τιμών.

Υπάρχουν διαφορετικές μέθοδοι ανάλογα με την κατασκευή του πιθανοτικού μοντέλου $p(y|x)$. Η βιβλιοθήκη Optuna χρησιμοποιεί τη μέθοδο TPE, η οποία κατασκευάζει ένα μοντέλο εφαρμόζοντας τον Μπεϋζιανό κανόνα:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)},$$

όπου $p(x|y)$ είναι η πιθανότητα των υπερ-παραμέτρων δεδομένου του κόστους της συνάρτησης κόστους, που με τη σειρά της εκφράζεται ως:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^*, \end{cases}$$

όπου $y < y^*$ αναπαριστά μία μικρότερη τιμή της συνάρτησης κόστους από το κατώφλι. Η εξίσωση αυτή εξηγείται ως ότι φτιάχνουμε δύο διαφορετικές κατανομές για τις υπερ-παραμέτρους: μία όπου η τιμή της συνάρτησης κόστους είναι μικρότερη από το κατώφλι, την $l(x)$, και μία όπου η τιμή της συνάρτησης κόστους είναι μεγαλύτερη από το κατώφλι, την $g(x)$. Ωστόσο, είναι φανερό ότι θέλουμε να παίρνουμε τιμές x από την $l(x)$, και όχι από την $g(x)$, καθώς αυτή η κατανομή έχει βασιστεί μόνο στις τιμές του x που παράγουν χαμηλότερο κόστος από το κατώφλι. Η $g(x)$ είναι απαραίτητη όμως καθώς η αναμενόμενη βελτίωση (Expected Improvement) είναι ανάλογη του $l(x)/g(x)$ και επομένως για τη μεγιστοποίηση της αναμενόμενης βελτίωσης πρέπει να μεγιστοποιήσουμε αυτόν τον όρο. Η αναμενόμενη βελτίωση είναι το κριτήριο μέσα από τη μεγιστοποίηση του οποίου επιλέγονται οι καλύτερες υπερ-παράμετροι. Η μέθοδος TPE λειτουργεί παίρνοντας τιμές υπερ-παραμέτρων από την κατανομή l(x), αξιολογώντας τις τιμές αυτές στον όρο $l(x)/g(x)$ και επιστρέφοντας το σύνολο τιμών που παράγει τη μεγαλύτερη τιμή στον όρο $l(x)/g(x)$. Ύστερα, αυτές οι υπερ-παράμετροι αξιολογούνται στη συνάρτηση κόστους και το πιθανοτικό μοντέλο ενημερώνεται κατάλληλα [3].

3.7 Έννοιες Επεξεργασίας Φυσικής Γλώσσας

Η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing, NLP) είναι ένας τομέας της τεχνητής νοημοσύνης που δίνει στις μηχανές την ικανότητα να διαβάζουν, να καταλαβαίνουν και να αντλούν νόημα από τις ανθρώπινες γλώσσες. Με απλά λόγια, η Επεξεργασία Φυσικής Γλώσσας αντιπροσωπεύει την αυτόματη ανάλυση της ανθρώπινης γλώσσας, όπως είναι η ομιλία ή η γραφή, και μπορεί να χρησιμοποιηθεί σε πολλούς τομείς, όπως είναι η ανάλυση συναισθημάτων, που παρέχει πολλές πληροφορίες για τις επιλογές των καταναλωτών, ή η αναγνώριση ανεπιθύμητης αλληλογραφίας.

[3] https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f

Στην περίπτωση μας, χρησιμοποιούμε την Επεξεργασία Φυσικής Γλώσσας για την αυτόματη ανάλυση tweets, για τον εντοπισμό περιεχομένου που είναι σχετικό με καταστροφές [4]. Σε αυτήν την ενότητα αναλύουμε μερικές έννοιες Επεξεργασίας Φυσικής Γλώσσας που χρησιμοποιούμε σε επόμενες ενότητες.

3.7.1 Tokenization

Tokenization είναι η διαδικασία διαχωρισμού κειμένου σε λέξεις (tokens). Ως tokens μπορούν να θεωρηθούν λέξεις που αποτελούνται απο γράμματα, σημεία στίξης ή και αριθμοί. Αναλόγως το πρόβλημα όμως κάποια από αυτά τα tokens μπορεί να μην είναι χρήσιμα. Για παράδειγμα, στα tweets που συνήθως υπάρχει μία μικρή πρόταση, η ύπαρξη των σημείων στίξης δεν προσφέρει κάποια χρησιμότητα. Επίσης, ο διαχωρισμός των λέξεων γίνεται ανάλογα με τη γλώσσα. Σε πολλές ευρωπαϊκές γλώσσες, όπως στα αγγλικά που χρησιμοποιούμε σε αυτήν την εργασία, ο διαχωρισμός γίνεται με βάση τα κενά διαστήματα, ενώ σε άλλες γλώσσες, όπως είναι τα κινέζικα, πρέπει να ακολουθηθεί διαφορετική τεχνική.

Πριν το tokenization πολλές φορές αφαιρούνται τα stop words. Οι λέξεις αυτές είναι λέξεις που εμφανίζονται πολύ συχνά στο κείμενο και θεωρούνται ότι περιέχουν ελάχιστη σημασιολογική χρησιμότητα. Παραδείγματα τέτοιων λέξεων στα αγγλικά είναι: the, a, and, to κ.τ.λ. Ωστόσο, η αρνητική πλευρά αφαίρεσης αυτών των λέξεων είναι ότι μπορεί να τροποποιηθεί το νόημα μιας πρότασης. Για παράδειγμα, σε μία ανάλυση συναισθημάτων θα ήταν λάθος να αφαιρεθεί η λέξη "not". Παρ' όλα αυτά, η αφαίρεση των stop words είναι χρήσιμη σε πολλές εφαρμογές της Επεξεργασίας Φυσικής Γλώσσας [40].

3.7.2 N-Grams

Τα μοντέλα που αναθέτουν πιθανότητες σε ακολουθίες λέξεων ονομάζονται γλωσσικά μοντέλα (language models). Το πιο απλό μοντέλο που αναθέτει πιθανότητες σε προτάσεις και σε ακολουθίες λέξεων είναι το n-gram. Ένα n-gram είναι μία ακολουθία από n λέξεις. Για παράδειγμα, ένα 2-gram είναι μία ακολουθία από δύο λέξεις και ονομάζεται bigram, ενώ ένα 3-gram είναι μία ακολουθία από τρεις λέξεις και ονομάζεται trigram. Η πιθανότητα μιας λέξης δεδομένου μιας ακολουθίας λέξεων υπολογίζεται με βάση την αλυσίδα Markov. Το μοντέλο αυτό είναι ένα στοχαστικό μοντέλο που περιγράφει μία ακολουθία από πιθανά γεγονότα, όπου η πιθανότητα για το κάθε γεγονός εξαρτάται μόνο από το προηγούμενο γεγονός. Στο n-gram μοντέλο, η πιθανότητα μιας λέξης δεδομένου όλων των προηγούμενων λέξεων υπολογίζεται μόνο από τις τελευταίες n λέξεις. Δηλαδή για το unigram, που περιλαμβάνει ακολουθία μιας λέξης, το bigram και το trigram μοντέλο θα έχουμε αντίστοιχα [39]:

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2)P(x_3)P(x_4),$$

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2|x_1)P(x_3|x_2)P(x_4|x_3),$$

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)P(x_4|x_1, x_2, x_3).$$

3.7.3 Word Embedding

Μία φυσική γλώσσα είναι ένα πολύπλοκο σύστημα που χρησιμοποιείται για την έκφραση νοημάτων. Σε αυτό το σύστημα οι λέξεις είναι η βασική μονάδα των γλωσσικών νοημάτων. Ένα διάνυσμα λέξης είναι ένα διάνυσμα που αντιπροσωπεύει μία λέξη. Η τεχνική λοιπόν που αντιστοιχεί λέξεις σε διανύσματα πραγματικών αριθμών ονομάζεται word embedding. Σε αυτήν την εργασία χρησιμοποιήσαμε τα GloVe Embeddings, που είναι μία παραλλαγή του skip-gram μοντέλου. Το μοντέλο αυτό θεωρεί ότι μία λέξη μπορεί να χρησιμοποιηθεί για την παραγωγή των λέξεων που την περιβάλλουν σε μία ακολουθία κειμένου. Για παράδειγμα, άμα θεωρήσουμε την ακολουθία κειμένου "ο", "άντρας", "αγαπά", "τον",

[4] https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1

"γιο", "του" και θεωρήσουμε τη λέξη "αγαπά" ως την κεντρική λέξη και ότι το παράθυρο συμφραζόμενων έχει μέγεθος 2, τότε το skip-gram μοντέλο υπολογίζει τη δεσμευμένη πιθανότητα παραγωγής των λέξεων που απέχουν το πολύ δύο λέξεις από την κεντρική λέξη. Επίσης, το μοντέλο αντιμετωπίζει ως ανεξάρτητες τις λέξεις που πρέπει να παραγάγει. Δηλαδή ισχύει:

$$P(\text{"ο"}, \text{"άντρας"}, \text{"τον"}, \text{"γιο"} | \text{"αγαπά"}) =$$

$$P(\text{"ο"} | \text{"αγαπά"}) P(\text{"άντρας"} | \text{"αγαπά"}) P(\text{"τον"} | \text{"αγαπά"}) P(\text{"γιο"} | \text{"αγαπά"})$$

Στο skip-gram μοντέλο, κάθε λέξη αντιπροσωπεύεται από δύο διανύσματα που χρησιμοποιούνται για τον υπολογισμό της δεσμευμένης πιθανότητας. Άμα θεωρήσουμε ότι μία λέξη έχει δείκτη i στο λεξικό, τότε το διάνυσμα της αναπαριστάται ως v_i όταν λειτουργεί ως κεντρική λέξη και ως u_i όταν είναι λέξη συμφραζομένων. Θεωρούμε ότι έχουμε την κεντρική λέξη w_c και τη λέξη συμφραζομένων w_o, με τους δείκτες c και o αντίστοιχα στο λεξικό. Η δεσμευμένη πιθανότητα παραγωγής της λέξης συμφραζομένων δεδομένου της κεντρικής λέξης δίνεται από τον παρακάτω τύπο:

$$P(w_o|w_c) = \frac{exp(u_o v_c)}{\sum_{i \in V} exp(u_i^T v_c)},$$

όπου ο δείκτης του λεξικού που ανήκει στο σύνολο $V = \{0, 1, ..., |V| - 1\}$. Θεωρούμε μία ακολουθία κειμένου μήκους T, όπου η λέξη στο χρονικό βήμα t συμβολίζεται ως $w^{(t)}$, και ότι οι λέξεις συμφραζομένων παραγάγονται ανεξάρτητα δεδομένου των κεντρικών λέξεων. Όταν το παράθυρο συμφραζομένων είναι m, η συνάρτηση πιθανότητας του skip-gram μοντέλου είναι η συλλογική πιθανότητα παραγωγής όλων των λέξεων συμφραζομένων δεδομένου οποιασδήποτε κεντρικής λέξης:

$$\prod_{t=1}^{T} \prod_{-m \le j \le m, j \ne 0} P(w^{(t+j)} | w(t)).$$

Οι παράμετροι του skip-gram μοντέλου είναι το διάνυσμα κεντρικής λέξης και το διάνυσμα λέξης συμφραζόμενων για κάθε ξεχωριστή λέξη. Η εκπαίδευση των παραμέτρων γίνεται με τη μέθοδο της μέγιστης πιθανοφάνειας, η οποία ισοδυναμεί με την ελαχιστοποίηση της παρακάτω συνάρτησης κόστους:

$$-\sum_{t=1}^{T} \sum_{-m \le j \le m, j \ne 0} log P(w^{(t+j)} | w(t)).$$

Μετά την εκπαίδευση, για κάθε λέξη στο λεξικό με δείκτη i, έχουμε τα δύο διανύσματα v_i και u_i. Η παραπάνω συνάρτηση μπορεί να γραφτεί και με άλλο τρόπο. Αρχικά, θεωρούμε ως q_{ij} τη δεσμευμένη πιθανότητα $P(w_j|w_i)$ που ορίσαμε παραπάνω. Κάθε λέξη w_i μπορεί να εμφανιστεί στο σύνολο δεδομένων πολλές φορές. Οπότε, εάν συγκεντρώσουμε όλες τις λέξεις συμφραζομένων κάθε φορά που η λέξη w_i είναι κεντρική λέξη, δημιουργούμε ένα σύνολο C_i, όπου μπορούν τα στοιχεία του συνόλου να εμφανίζονται πάνω από μία φορά (multiset). Ο αριθμός των φορών που εμφανίζεται μία λέξη σε αυτό το σύνολο ονομάζεται πολλαπλότητα της λέξης. Έστω x_{ij} η πολλαπλότητα της λέξης με δείκτη j στο σύνολο C_i. Δηλαδή, x_{ij} είναι οι φορές που εμφανίζεται η λέξη w_j ως λέξη συμφραζομένων για την κεντρική λέξη w_i σε όλο το σύνολο δεδομένων. Συνεπώς, η παραπάνω συνάρτηση κόστους μπορεί να γραφτεί ως:

$$-\sum_{i \in V} \sum_{j \in V} x_{ij} log q_{ij}.$$

Προσθέτουμε τους αριθμούς από όλες τις λέξεις συμφραζομένων για την κεντρική λέξη w_i και παίρνουμε το x_i. Στη συνέχεια, θεωρούμε ως p_{ij} τη δεσμευμένη πιθανότητα x_{ij}/x_i για την παραγωγή της λέξης w_j με βάση την κεντρική λέξη w_i. Οπότε, μπορούμε να γράψουμε τη συνάρτηση κόστους ως εξης:

$$-\sum_{i \in V} x_i \sum_{j \in V} p_{ij} log q_{ij}.$$

Το εσωτερικό άθροισμα υπολογίζει την κατανομή της δεσμευμένης πιθανότητας p_{ij} για την παραγωγή των λέξεων συμφραζομένων δεδομένου της κεντρικής λέξης w_i και τη διαστραυρούμενη εντροπία (cross-entropy) της κατανομής της δεσμευμένης πιθανότητας q_{ij}. Η συνάρτηση κόστους έχει ως βάρος το άθροισμα των πολλαπλοτήτων των λέξεων συμφραζομένων δεδομένου της κεντρικής λέξης w_i. Ωστόσο, η αποθήκευση όλων των αθροισμάτων όλων των στοιχείων όλου του λεξικού μπορεί εύκολα να οδηγήσει σε υπερβολικό υπολογιστικό κόστος. Επίσης, συχνά υπάρχουν πολλές ασυνήθιστες λέξεις στο λεξικό, που εμφανίζονται σπάνια στο σύνολο δεδομένων. Στη διαστραυρούμενη εντροπία ως συνάρτηση κόστους, η τελική πρόβλεψη της κατανομής της δεσμευμένης πιθανότητας σε αυτές τις λέξεις είναι πιθανό να είναι ανακριβής. Για αυτούς τους λόγους το GloVe έκανε κάποιες αλλαγές στο skip-gram μοντέλο.

Το GloVe υιοθέτησε το τετραγωνικό σφάλμα (squared loss) και έκανε τρεις αλλαγές στο skip-gram μοντέλο. Πρώτα, θεωρεί τις μεταβλητές $p'_{ij} = x_{ij}$ και $q'_{ij} = exp(u_j^T v_i)$ και τις λογαριθμίζει. Οπότε παίρνουμε το τετραγωνικό σφάλμα $(logp'_{ij} - logq'_{ij})^2 = (u_j^T v_i - log x_{ij})^2$. Ύστερα, προσθέτει δύο βαθμωτές παραμέτρους για κάθε λέξη w_i, τους όρους μεροληψίας (bias) b_i (για τις κεντρικές λέξεις) και c_i (για τις λέξεις συμφραζομένων). Τέλος, αντικατέστησε το βάρος της συνάρτησης κόστους με τη συνάρτηση $h(x_{ij})$, που είναι μία γνησίως αύξουσα συνάρτηση στο διάστημα $[0, 1]$, δηλαδή είναι μία συνάρτηση που συνεχώς αυξάνεται, δεν μένει ποτέ σταθερή, ούτε μειώνεται. Συνεπώς, στόχος του GloVe είναι η ελαχιστοποίηση της παρακάτω συνάρτησης κόστους:

$$\sum_{i \in V} \sum_{j \in V} h(x_{ij})(u_j^T v_i + b_i + c_j - log x_{ij})^2.$$

Τα x_{ij} υπολογίζονται πριν από την εκπαίδευση με βάση όλο το σύνολο δεδομένων και περιέχουν όλα τα στατιστικά (global) των δεδομένων. Επομένως, το όνομα του GloVe μοντέλου προέρχεται από τα "Global Vectors". Σημειώνουμε ότι αν η λέξη w_i εμφανίζεται στο παράθυρο συμφραζομένων της λέξης w_j, τότε η λέξη w_j θα εμφανιστεί επίσης στο παράθυρο συμφραζομένων της λέξης w_i και ως αποτέλεσμα θα ισχύει $x_{ij} = x_{ji}$. Σε αντίθεση με το skip-gram μοντέλο, το GloVe προσαρμόζει τον συμμετρικό όρο $log x_{ij}$ αντί της ασυμμετρικής δεσμευμένης πιθανότητας p_{ij}. Ως εκ τούτου, το διάνυσμα κεντρικής λέξης και το διάνυσμα λέξης συμφραζόμενων κάθε λέξης είναι ισοδύναμα στο GloVe. Ωστόσο, οι τιμές των δύο διανυσμάτων μπορεί να διαφέρουν μετά την εκπαίδευση λόγω της διαφορετικής αρχικοποίησης τους. Αφού εκπαιδευτούν όλα τα διανύσματα λέξεων, το GloVe χρησιμοποιεί το άθροισμα του διανύσματος κεντρικής λέξης και του διανύσματος λέξης συμφραζόμενων ως το τελικό διάνυσμα λέξης για την κάθε λέξη [39].

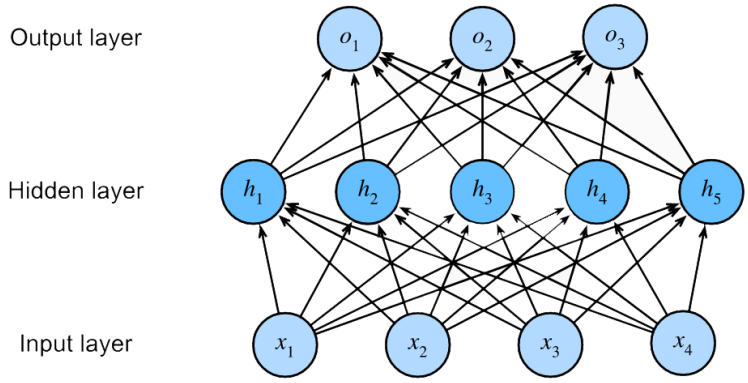
3.8 Αρχιτεκτονικές Νευρωνικών Δικτύων

Σε αυτήν την ενότητα περιγράφουμε την αρχιτεκτονική των νευρωνικών δικτύων που χρησιμοποίησαμε.

3.8.1 DFNN

Τα Deep FeedForward Neural Networks (DFNNs), ή αλλιώς Multilayer Perceptrons (MLPs) είναι βασικά μοντέλα βαθιάς μηχανικής μάθησης. Ονομάζονται δίκτυα πρόσθιας τροφοδότησης καθώς

η πληροφορία ρέει σε μία ευθύγραμμη κατεύθυνση, δηλαδή ρέει μέσω της εισόδου x μιας συνάρτησης f, μέσω των ενδιάμεσων υπολογισμών που καθορίζουν τη συνάρτηση f και τελικά καταλήγει στον υπολογισμό της εξόδου y. Δεν υπάρχουν συνδέσεις όπου έξοδοι του μοντέλου συνδέονται με εισόδους του μοντέλου. Στην περίπτωση που υπήρχαν τέτοιες συνδέσεις το μοντέλο θα ονομαζόταν Recurrent Neural Network (RNN). Το DFNN αποτελείται από το επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα και το επίπεδο εξόδου [38]. Κάθε επίπεδο είναι πλήρως συνδεδεμένο, δηλαδή κάθε νευρώνας συνδέεται με κάθε νευρώνα του επόμενου επιπέδου με ένα αντίστοιχο βάρος w_{ij}, ενώ δεν υπάρχουν συνδέσεις μεταξύ νευρώνων του ίδιου επιπέδου ή που δεν ανήκουν σε διαδοχικά επίπεδα, όπως φαίνεται στην παρακάτω εικόνα. Σε κάθε επίπεδο, εκτός του επιπέδου εισόδου, χρησιμοποιείται μία συνάρτηση ενεργοποίησης που προσθέτει συνήθως μία μη γραμμικότητα στους υπολογισμούς. Αυτές που χρησιμοποιούνται συνήθως είναι η ReLU, η σιγμοειδής και η υπερβολική εφαπτομένη. Επιπλέον, τα DFNN ακολουθούν το παγκόσμιο θεώρημα προσέγγισης. Δηλαδή, ένα DFNN με ένα κρυφό επίπεδο, με τον κατάλληλο αριθμό νευρώνων και τα κατάλληλα βάρη, μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση. Ωστόσο, συνήθως ο προσδιορισμός αυτής της συνάρτησης είναι αρκετά δύσκολος, οπότε η προσέγγιση μιας συνάρτησης μπορεί να γίνει πιο εύκολα με τη χρήση περισσότερων κρυφών επιπέδων [39].



Εικόνα 6: Ένα DFNN με ένα κρυφό επίπεδο και 5 κρυφούς νευρώνες [39].

3.8.2 CNN

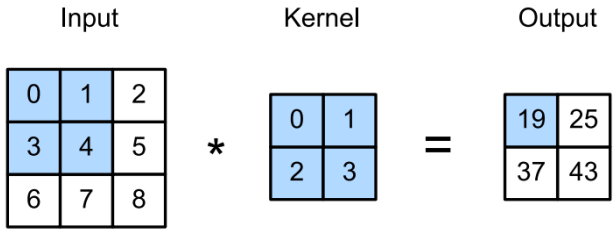
Τα Convolutional Neural Networks (CNNs) είναι μία κατηγορία νευρωνικών δικτύων που περιλαμβάνουν επίπεδα συνέλιξης. Στα μαθηματικά η συνέλιξη μεταξύ δύο συναρτήσεων, $f, g : \mathbb{R}^d \to \mathbb{R}$ ορίζεται ως:

$$(f * g)(x) = \int f(z)g(x - z)dz.$$

Με τη συνέλιξη μετράμε την επικάλυψη μεταξύ των f και g όταν η μία συνάρτηση έχει υποστεί ανάκλαση ως προς τον κατακόρυφο άξονα και μετατόπιση κατά x. Όταν έχουμε διαχωρίσιμα στοιχεία, το ολοκλήρωμα μετατρέπεται σε άθροισμα. Τα μοντέλα αυτά είναι κυρίως σχεδιασμένα για προβλήματα που έχουν να κάνουν με εικόνες, αλλά έχουν εφαρμογές και σε προβλήματα Επεξεργασίας Φυσικής Γλώσσας.

Σε ένα επίπεδο συνέλιξης ο τανυστής εισόδου (tensor), που είναι ένας πίνακας n-διάστασης, και ο τανυστής πυρήνα (kernel) συνδυάζονται για να παράγουν έναν τανυστή εξόδου μέσω μιας λειτουργίας αλληλοσυσχέτισης (cross-correlation). Στο παράδειγμα της παρακάτω εικόνας, η είσοδος είναι ένας τανυστής με διαστάσεις 3×3, ενώ ο πυρήνας είναι ένας τανυστής με διαστάσεις 2×2. Οι διαστάσεις του πυρήνα καθορίζουν και το παράθυρο της συνέλιξης. Σε μία λειτουργία αλληλοσυσχέτισης, το παράθυρο της συνέλιξης τοποθετείται στην πάνω αριστερή γωνία του τανυστή εισόδου και στη συνέχεια

ολισθαίνει πάνω στον τανυστή εισόδου από τα αριστερά προς τα δεξια και από πάνω προς τα κάτω. Όταν τα παράθυρο τη συνέλιξης βρίσκεται σε μία θέση, τότε ο υπο-τανυστής εισόδου που βρίσκεται σε αυτό το παράθυρο πολλαπλασιάζεται στοιχείο προς στοιχείο με τον πυρήνα, ενώ τα στοιχεία του τανυστή που προκύπτουν αθροίζονται σε μία βαθμωτή τιμή. Η τιμή αυτή μπαίνει στην αντίστοιχη θέση στον τανυστή εξόδου. Σημειώνουμε ότι ο τανυστής εξόδου έχει μέγεθος μικρότερο από τον τανυστή εισόδου. Αν ο τανυστής εισόδου έχει διαστάσεις $n_h \times n_w$ και ο πυρήνας έχει διαστάσεις $k_h \times k_w$, τότε ο τανυστής εξόδου θα έχει διαστάσεις $(n_h - k_h + 1) \times (n_w - k_w + 1)$. Σημειώνουμε επίσης ότι το αποτέλεσμα θα είναι το ίδιο είτε εκτελέσουμε την ακριβή λειτουργία της συνέλιξης είτε εκτελέσουμε τη λειτουργία αλληλοσυσχέτισης. Κατά την εκπαίδευση ο πυρήνας αρχικοποιείται αρχικά με τυχαίες τιμές. Ύστερα, σε κάθε επανάληψη, υπολογίζεται το κόστος και η κλίση (gradient) από τη συνάρτηση κόστους και τελικά ενημερώνονται κατάλληλα οι τιμές του πυρήνα. Ένα CNN, επιπλέον, μπορεί να έχει παραπάνω από έναν πυρήνα, που αλλιώς ονομάζονται φίλτρα. Η λειτουργία αλληλοσυσχέτισης φαίνεται και στο παρακάτω παράδειγμα:



Εικόνα 7: Δισδιάστατη λειτουργία αλληλοσυσχέτισης (cross-correlation), όπου τα σκιασμένα μέρη είναι τα στοιχεία εισόδου και τα στοιχεία του πυρήνα που χρησιμοποιούνται για τον υπολογισμό της πρώτης εξόδου [39].

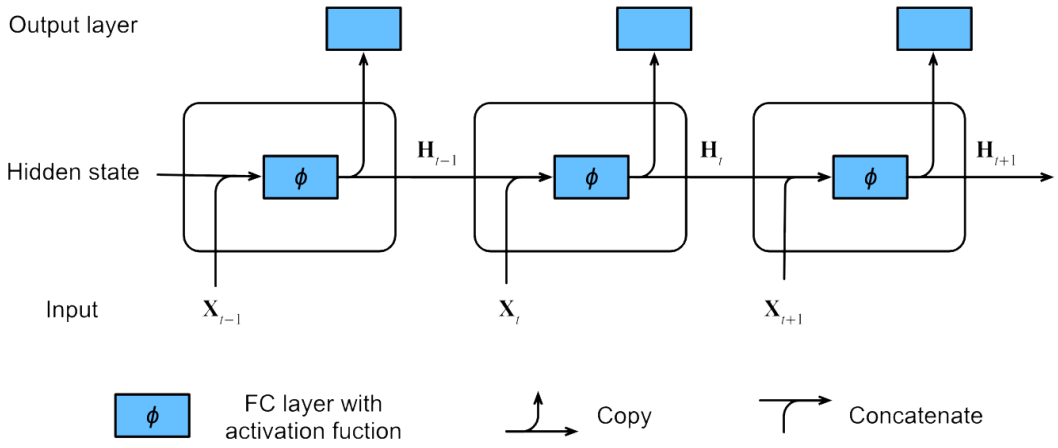
Το μέγεθος του τανυστή εξόδου μπορεί να επηρεαστεί από δύο τεχνικές, το padding και το stride. Στην πρώτη τεχνική προσθέτονται επιπλέον τιμές με μηδενικά γύρω από τα όρια του τανυστή εισόδου. Με αυτόν τον τρόπο, εάν προσθέσουμε p_h σειρές και p_w στήλες από padding συνολικά στον τανυστή εισόδου τότε το ύψος και το πλάτος του τανυστή εξόδου θα αυξηθεί επίσης κατά p_h και p_w αντίστοιχα. Σε πολλές περιπτώσεις προτιμάται ο τανυστής εξόδου να έχει τις ίδιες διαστάσεις με τον τανυστή εισόδου, καθώς αυτό διευκολύνει τον καθορισμό των υπόλοιπων επιπέδων στην κατασκευή ενός δικτύου. Η δεύτερη τεχνική έχει να κάνει με το πόσες θέσεις (stride) μετακινείται το παράθυρο της συνέλιξης, ενώ χρησιμοποιείται όταν θέλουμε να μειωθούν οι διαστάσεις του τανυστή εξόδου. Συνολικά, άμα το stride για το ύψος είναι s_h και το stride για το πλάτος είναι s_w, τότε ο τανυστής εξόδου θα έχει διαστάσεις $[(n_h - k_h + p_h + s_h)/s_h] \times [(n_w - k_w + p_w + s_w)/s_w]$.

Ένα άλλο επίπεδο που συναντάται συνήθως στα CNNs είναι το επίπεδο ομαδοποίησης (pooling). Το επίπεδο αυτό έχει ως στόχο να μετριάσει την ευαισθησία των επιπέδων συνέλιξης και να συναθροίσει πληροφορίες, έτσι ώστε να δημιουργηθεί μία αναπαράσταση που θα έχει κρατήσει τις πιο σημαντικές πληροφορίες που χρειάζονται για την ταξινόμηση. Όπως και στα επίπεδα συνέλιξης, οι λειτουργίες ομαδοποίησης αποτελούνται από ένα καθορισμένου μεγέθους παράθυρο που ολισθαίνει πάνω στον τανυστή εισόδου από τα αριστερά προς τα δεξια και από πάνω προς τα κάτω. Για κάθε μία τοποθεσία που διασχίζεται από το παράθυρο υπολογίζεται μία έξοδος. Αντίθετα από τη λειτουργία αλληλοσυσχέτισης το επίπεδο ομαδοποίησης δεν περιέχει πυρήνα, αλλά γίνεται μία πράξη στα στοιχεία που ανήκουν στο παράθυρο ομαδοποίησης. Οι δύο πιο τυπικές λειτουργίες είναι ο υπολογισμός του μέγιστου ή του μέσου όρου των στοιχείων που ανήκουν στο παράθυρο και ονομάζονται μέγιστη ομαδοποίηση (maximum ή max pooling) και ομαδοποίηση μέσου όρου (average pooling) αντίστοιχα. Η έξοδος των επιπέδων ομαδοποίησης μπορεί αντίστοιχα να ρυθμιστεί με τις τεχνικές padding και stride [39].

Ένα CNN σε προβλήματα με κείμενο λειτουργεί σαν n-gram. Δηλαδή, συγκεντρώνει πληροφορίες από n λέξεις στη σειρά. Για παράδειγμα, ένα 1×2 φίλτρο βλέπει δύο συνεχόμενες λέξεις σε ένα κείμενο κάθε φορά, οπότε λειτουργεί σαν ένα bigram. Με τη μετατροπή του κειμένου σε word embeddings, το κείμενο γίνεται σαν ένας δισδιάστατος τανυστής, όπου μία γραμμή είναι η αναπαράσταση μιας λέξης σε διάνυσμα. Άρα, εάν χρησιμοποιήσουμε φίλτρο διαστάσεων $n \times emb\_dim$, όπου emb_dim είναι η διάσταση των word embeddings, τότε το φίλτρο καλύπτει εντελώς n συνεχόμενες λέξεις. Συνεπώς, η έξοδος για το συγκεκριμένο φίλτρο είναι ένας τανυστής πλάτους 1 και ύψους το μήκος της πρότασης, μείον το ύψος του φίλτρου n, συν 1. Η διαδικασία αυτή συνήθως επαναλαμβάνεται για N διαφορετικά φίλτρα. Η λογική είναι ότι το κάθε φίλτρο θα μάθει να εξάγει ένα διαφορετικό χαρακτηριστικό και στη συνέχεια μέσω του επιπέδου ομαδοποίησης επιλέγεται η πιο σημαντική πληροφορία του κάθε χαρακτηριστικού. Καθώς η λειτουργίες της συνέλιξης και της ομαδοποίησης εκτελούνται ανεξάρτητα, τα χαρακτηριστικά που εξάγονται δεν περιέχουν πληροφορίες για την τοποθεσία των λέξεων στην πρόταση. Ωστόσο, οι αλληλεπιδράσεις μεταξύ των διαφορετικών χαρακτηριστικών μπορούν να μοντελοποιηθούν με τη χρήση ενός πλήρους συνδεδεμένου επιπέδου μετά το επίπεδο ομαδοποίησης [41].

3.8.3 LSTM

Τα Long Short-Term Memory (LSTMs) είναι μία εξέλιξη των Recurrent Neural Networks (RNNs). Τα RNNs είναι σχεδιασμένα για να χειρίζονται ακολουθητική πληροφορία, ενώ χρησιμοποιούν μεταβλητές για να αποθηκεύσουν παρελθοντική πληροφορία, ώστε μαζί με την τρέχουσα είσοδο να καθορίσουν την τρέχουσα έξοδο. Δηλαδή, περιλαμβάνουν κρυφές καταστάσεις, που διαφέρουν απο τα κρυφά επίπεδα. Αντίθετως από τα DFNN, αποθηκεύεται η κρυφή κατάσταση του προηγούμενου χρονικού βήματος και περιλαμβάνονται επιπλέον βάρη που περιγράφουν το πως η κρυφή κατάσταση του προηγούμενου χρονικού βήματος επηρεάζει το τρέχων χρονικό βήμα. Συνεπώς, μία κρυφή καταστάση αποτελεί τη μνήμη του νευρωνικού δικτύου. Τα δίκτυα αυτά ονομάζονται επαναλαμβανόμενα, καθώς ο υπολογισμός της κρυφής κατάστασης επαναλαμβάνεται σε κάθε χρονικό βήμα. Συγκεκριμένα, σε κάθε χρονικό βήμα t, ο υπολογισμός της κρυφής κατάστασης μπορεί να θεωρηθεί ως εξής: πρώτα συνενώνεται η είσοδος X_t στο τρέχων χρονικό βήμα t με την κρυφή κατάσταση H_{t-1} του προηγούμενου χρονικού βήματος $t-1$ και μετά το αποτέλεσμα τροφοδοτείται σε ένα πλήρως συνδεδεμένο επίπεδο με μία συνάρτηση ενεργοποίησης. Το αποτέλεσμα αυτού του επιπέδου είναι η κρυφή κατάσταση H_t του τρέχοντος χρονικού βήματος t. Στη συνέχεια, η H_t θα συμμετέχει στον υπολογισμό της κρυφής κατάστασης H_{t+1}. Τα βήματα αυτά φαίνονται και στην παρακάτω εικόνα. Ωστόσο, τα RNNs έχουν το πρόβλημα ότι δεν μπορούν να διατηρήσουν μακροπρόθεσμα πληροφορίες, κάτι που καλύπτει το LSTM.



Εικόνα 8: Ένα RNN με μία κρυφή κατάσταση [39].

Τα LSTM χρησιμοποιούν ένα κελί μνήμης το οποίο έχει παρόμοια λογική με την κρυφή κατάσταση, το οποίο όμως μπορεί να αποθηκεύσει περισσότερη πληροφορία. Για τον έλεγχο αυτού του κελιού είναι απαραίτητες κάποιες πύλες. Η πρώτη πύλη χρησιμοποιείται για την έξοδο του κελιού, που ονομάζεται πύλη εξόδου. Η δεύτερη πύλη, που ονομάζεται πύλη εισόδου, καθορίζει το κατά πόσο θα ληφθεί υπόψιν η νέα είσοδος. Τέλος, η τρίτη πύλη ονομάζεται πύλη λήθης και καθορίζει πόση από την παλιά μνήμη θα διατηρηθεί. Όπως και στα RNNs, η είσοδος στις πύλες του LSTM είναι η είσοδος στο τρέχων χρονικό βήμα και η κρυφή κατάσταση του προηγούμενου χρονικού βήματος. Αφού συνενώνονται, περνάνε από τρία πλήρως συνδεδεμένα επίπεδα με μία σιγμοειδή συνάρτηση ενεργοποίησης το καθένα για τον υπολογισμό των τιμών των τριών πυλών. Ως αποτέλεσμα, οι τιμές των τριών πυλών είναι στο εύρος $(0, 1)$. Το κάθε πλήρως συνδεδεμένο επίπεδο περιλαμβάνει αντίστοιχα βάρη για την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση. Άμα θεωρήσουμε ως X_t την είσοδο, ως H_{t-1} την κρυφή κατάσταση του προηγούμενου χρονικού βήματος, ως I_t, F_t και O_t τις πυλες εισόδου, λήθης και και εξόδου αντίστοιχα για το χρονικό βήμα t, τότε έχουμε:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i),$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f),$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o),$$

όπου W_{xi}, W_{xf}, W_{xo} και W_{hi}, W_{hf}, W_{ho} είναι τα βάρη και b_i, b_f, b_o είναι οι παράμετροι μεροληψίας (bias).

Το κελί μνήμης του LSTM, εκτός από την τρέχουσα κρυφή κατάσταση, έχει ως έξοδο και τη μνήμη του κελιού, C_t. Η έξοδος αυτή καθορίζεται από την πύλη λήθης, την πύλη εισόδου, την υποψήφια μνήμη κελιού $\tilde{C}_t$, καθώς και τη μνήμη C_{t-1} του προηγούμενου χρονικού βήματος. Η υποψήφια μνήμη κελιού υπολογίζεται με παρόμοιο τρόπο με αυτόν που υπολογίζονται οι τρεις πύλες, με τη διαφορά ότι χρησιμοποιείται η υπερβολική εφαπτομένη ως συνάρτηση ενεργοποίησης αντί της σιγμοειδούς. Επόμενως, η υποψήφια μνήμη κελιού θα έχει τιμές στο εύρος $(-1, 1)$. Δηλαδή έχουμε:

$$\tilde{C}_t = tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c),$$

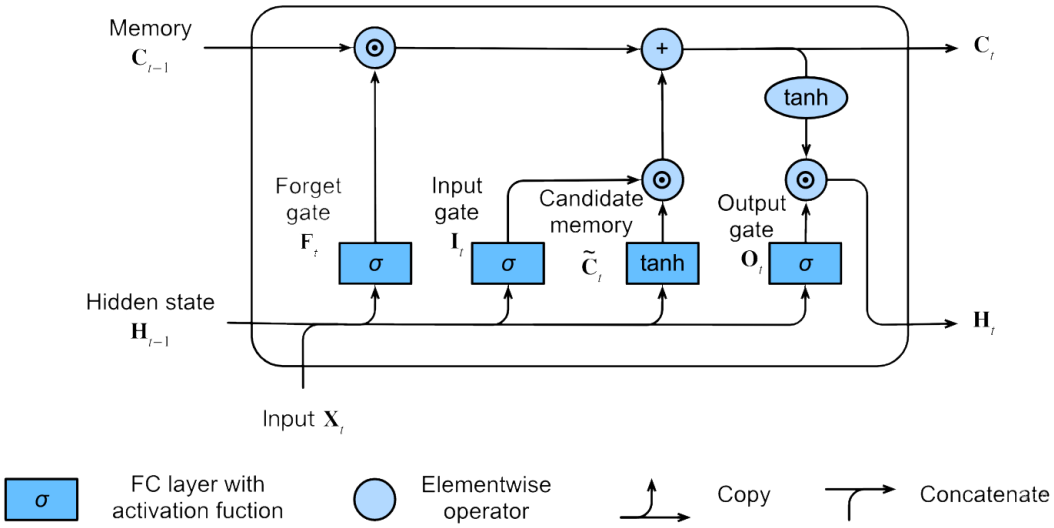
όπου W_{xc} και W_{hc} είναι τα βάρη και b_c είναι η παράμετρος μεροληψίας. Οπότε η μνήμη του κελιού δίνεται από την παρακάτω εξίσωση:

$$C_t = F_t * C_{t-1} + I_t * \tilde{C}_t.$$

Αν η πύλη λήθης είναι πάντα κοντά στο 1 και η πύλη εισόδου είναι πάντα κοντά στο 0, η προηγούμενη μνήμη κελιού C_{t-1} θα αποθηκεύεται και θα περνάει στο τρέχων χρονικό βήμα. Τέλος, η τρέχουσα κρυφή κατάσταση H_t καθορίζεται από την πύλη εξόδου και την τρέχουσα μνήμη κελιού. Δηλαδή:

$$H_t = O_t * tahn(C_t).$$

Με αυτόν τον τρόπο οι τιμές της τρέχουσας κρυφής κατάστασης είναι πάντα στο διάστημα $(-1, 1)$. Όταν η πύλη εξόδου είναι κοντά στο 1, μεταφέρεται όλη η πληροφορία της μνήμης, ενώ άμα είναι κοντά στο 0, τότε η πληροφορία της μνήμης μένει μόνο στο τρέχων κελί μνήμης και δεν προχωράει παρακάτω. Στο τελικό επίπεδο εξόδου περνάει μόνο η τελευταία κρυφή κατάσταση. Η μνήμη του κελιού είναι πλήρως εσωτερική του LSTM. Στην παρακάτω εικόνα φαίνεται ένα ολοκληρωμένο κελί μνήμης του μοντέλου LSTM [39].



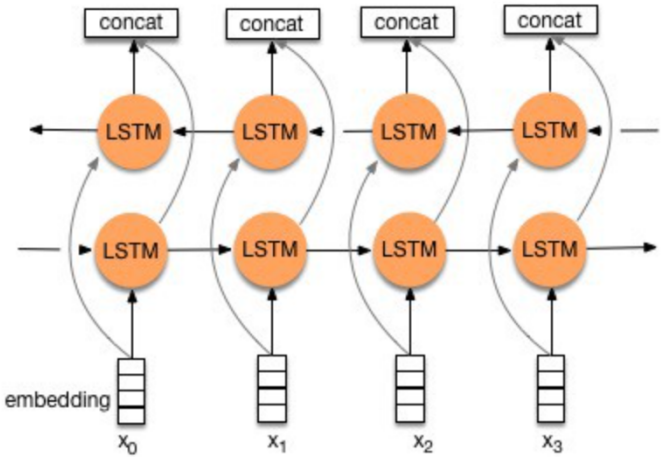
Εικόνα 9: Το κελί μνήμης ενός LSTM μοντέλου [39].

3.8.4 BiLSTM

Ένα Bidirectional LSTM (BiLSTM) αποτελείται από δύο ανεξάρτητα LSTMs. Η ακολουθία της εισόδου τροφοδοτείται στο ένα δίκτυο με την κανονική της κατεύθυνση και με την ανάποδη κατεύθυνση στο δεύτερο δίκτυο. Οι έξοδοι των δύο δικτύων συνενώνονται σε κάθε χρονικό βήμα. Συνεπώς, ένα από τα βασικά χαρακτηριστικά των BiLSTM, και των BiRNNs, είναι ότι η πληροφορία και από τις δύο πλευρές της ακολουθίας εισόδου χρησιμοποιούνται για την εκτίμηση της εξόδου. Με αυτόν τον τρόπο τα BiLSTMs αυξάνουν τον όγκο της πληροφορίας που δέχονται. Για παράδειγμα, τα μοντέλα αυτά γνωρίζουν ποιες λέξεις προηγούνται από μία λέξη ή ακολουθούν μία λέξη. Ωστόσο, τα μοντέλα αυτά είναι πολύ αργά [39]. Όπως βλέπουμε στην παρακάτω εικόνα [5], το BiLSTM επιστρέφει τις δύο εξόδους από τα δύο LSTM συνενωμένες για κάθε λέξη. Συνεπώς, στην λέξη x_0 το πρώτο μισό κομμάτι της εξόδου περιλαμβάνει την αντίστοιχη έξοδο από το ευθύ LSTM. Ωστόσο, σε αυτό το στάδιο το ευθύ LSTM έχει μόνο εξέτασει τη λέξη x_0. Το δεύτερο μισό κομμάτι της εξόδου της λέξης x_0 περιλαμβάνει την αντίστοιχη έξοδο από το ανάποδο LSTM για αυτή τη λέξη. Το ανάποδο LSTM όμως έχει δει σε αυτό το σημείο όλες τις λέξεις της πρότασης με ανάποδη σειρά. Επομένως, η τελική έξοδος του ανάποδου LSTM είναι το δεύτερο μισό κομμάτι της εξόδου της λέξης x_0. Αντίστοιχα, η τελική έξοδος

[5] https://towardsdatascience.com/understanding-bidirectional-rnn-in-pytorch-5bd25a5dd66

του ευθέος LSTM είναι το πρώτο μισό κομμάτι της εξόδου της τελευταίας λέξης της πρότασης, καθώς έτσι το LSTM θα έχει δει όλες τις λέξεις. Συνεπώς, στο τελικό επίπεδο εξόδου πρέπει να περάσουν συνενωμένες οι δύο σωστές έξοδοι του κάθε LSTM (στη βιβλιοθήκη PyTorch).

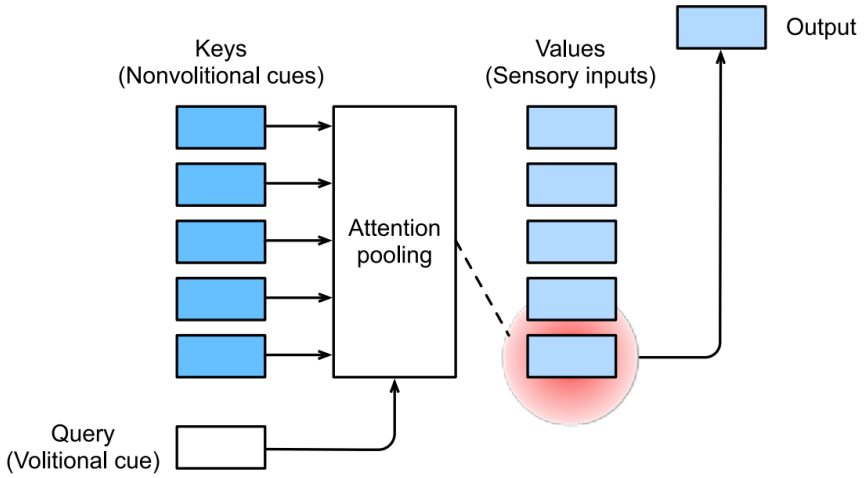


Εικόνα 10: Το BiLSTM μοντέλο.

3.8.5 Μηχανισμός Προσοχής

Ένα νευρωνικό δίκτυο θεωρείται μία προσπάθεια μίμησης των λειτουργιών του ανθρώπινου εγκεφάλου με έναν απλοποιημένο τρόπο. Έτσι και ο μηχανισμός προσοχής (Attention Mechanism) είναι μία προσπάθεια αντιγραφής της ανθρώπινης προσοχής. Η ικανότητα καθοδήγησης της προσοχής με μόνο ένα μικρό κομμάτι πληροφορίας, επιτρέπει στον ανθρώπινο εγκέφαλο να διανέμει πιο έξυπνα τους πόρους ώστε να ανταπεξέλθει στην αντίστοιχη δραστηριότητα. Με παρόμοιο τρόπο, ο μηχανισμός προσοχής συγκεντρώνει μερικά σχετικά στοιχεία, ενώ αγνοεί τα υπόλοιπα. Ο μηχανισμός αυτός μπορεί να αντιμετωπίσει το πρόβλημα των RNN/LSTMs, δηλαδή που τείνουν να είναι μυωπικά. Παρόλο που το LSTM θεωρείται ότι μπορεί να συλλάβει καλύτερα τις εξαρτήσεις των λέξεων μακροπρόθεσμα, δεν μπορεί να δώσει μεγαλύτερη βαρύτητα σε κάποιες από τις λέξεις εισόδου, συγκριτικά με τις υπόλοιπες, όπως διαβάζει μία ακολουθία.

Στη βιολογία, η προσοχή καθορίζεται με τη χρήση των μη-θεληματικών (nonvolitional) και των θεληματικών (volitional) σημάτων (cue). Το μη-θεληματικό σήμα βασίζεται στην ευκρίνεια των αντικειμένων στο χώρο, ενώ το θεληματικό σήμα βασίζεται στην πράξη που θέλουμε να κάνουμε. Δηλαδή, παρόλο που το μη-θεληματικό σήμα μας προδιαθέτει να στρέψουμε την προσοχή μας στο αντίστοιχο αντικείμενο, το θεληματικό σήμα είναι πιο ισχυρό. Κατασκευαστικά, αυτό που διαχωρίζει τους μηχανισμούς προσοχής από τα πλήρως συνδεδεμένα επίπεδα είναι τα "ερωτήματα" (queries), που λειτουργούν ως τα θεληματικά σήματα. Δεδομένου ενός ερωτήματος, ο μηχανισμός προσοχής προδιαθέτει το αποτέλεσμα μέσω ενός επιπέδου ομαδοποίησης προσοχής (attention pooling). Δηλαδή, εκφράζει τη συσχέτιση του ερωτήματος με τις εισόδους (values). Κάθε είσοδος (value) συνδέεται με ένα κλειδί (key), το οποίο μπορεί να θεωρηθεί ως το μη-θεληματικό σήμα. Όπως φαίνεται στην παρακάτω εικόνα, το επίπεδο ομαδοποίησης προσοχής μπορεί να σχεδιαστεί ώστε το ερώτημα να αλληλεπιδρά με τα κλειδιά, ώστε να οδηγούμαστε τελικά στη μεροληπτική επιλογή εισόδων.



Εικόνα 11: Ο μηχανισμός προσοχής προδιαθέτει την επιλογή εισόδων (values) μέσω ενός επιπέδου ομαδοποίησης προσοχής (attention pooling), όπου τα ερωτήματα (queries) αλληλεπιδρούν με τα κλειδιά (keys) [39].

Το επίπεδο ομαδοποίησης προσοχής συναθροίζει τιμές χρησιμοποιώντας βάρη, τα οποία μπορούν να εκπαιδεύονται μαζί με το υπόλοιπο δίκτυο. Στην περίπτωση που τα ερωτήματα είναι τα ίδια με την είσοδο (κλειδιά και είσοδοι), ο μηχανισμός ονομάζεται Self-Attention, όπου συσχετίζονται διαφορετικές θέσεις της ίδιας εισόδου [39]. Σε αυτήν την εργασία χρησιμοποιήσαμε τον Ιεραρχικό Μηχανισμό Προσοχής (Hierarchical Attention Network, HAN). Αν θεωρήσουμε ως h_i, την αντίστοιχη ενδιάμεση έξοδο ενός LSTM μοντέλου, ως W και b τα βάρη και την παράμετρο μεροληψίας (bias) του μηχανισμού προσοχής αντίστοιχα, τότε έχουμε:

$$u_i = tanh(Wh_i + b)$$

$$a_i = softmax(u_s^T u_i)$$

$$v = \sum_i a_i h_i,$$

όπου u_s είναι το διάνυσμα με τιμές 0 και 1 που εκφράζει το πραγματικό μήκος της κάθε πρότασης [42].

3.9 Μεταφορά Μάθησης

Η μεταφορά μάθησης (transfer learning) είναι μία τεχνική κατά την οποία οι γνώσεις που έχουν παρθεί από ένα πρόβλημα χρησιμοποιούνται για τη βελτίωση της γενίκευσης σε ένα άλλο πρόβλημα. Συνήθως, η τεχνική αυτή χρησιμοποιείται για τη μεταφορά γνώσης από προβλήματα όπου που είναι διαθέσιμα πολλά δεδομένα σε προβλήματα όπου λίγα δεδομένα είναι διαθέσιμα. Με αυτόν τον τρόπο, το μοντέλο που θα χρησιμοποιήσει τις γνώσεις του πρώτου μοντέλου θα μπορεί να γενικεύσει πιο γρήγορα με λίγα μόνο δεδομένα [38]. Μία σύνηθης τεχνική της μεταφοράς μάθησης είναι το fine-tuning, που φαίνεται και στην ακόλουθη εικόνα. Τα βήματα αυτής της τεχνικής είναι τα ακόλουθα:

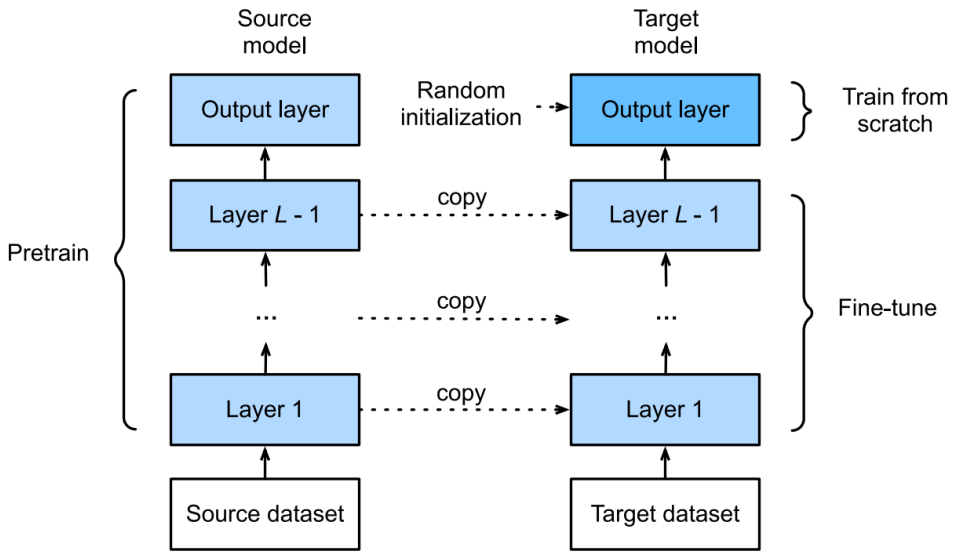
1. Εκπαίδευση ενός νευρωνικού δίκτυου, που αποτελεί το μοντέλο-πηγή, σε κάποιο σύνολο δεδομένων.

2. Δημιουργία ενός νέου νευρωνικού μοντέλου, που αποτελεί το μοντέλο-στόχος. Αυτό το μοντέλο αντιγράφει όλη την αρχιτεκτονική και τις παραμέτρους του μοντέλου-πηγή εκτός από το επίπεδο εξόδου. Υποθέτουμε ότι αυτές οι παράμετροι περιέχουν γνώση από τα δεδομένα στα οποία

εκπαιδεύτηκε το μοντέλο-πηγή και ότι αυτή η γνώση είναι εφαρμόσιμη στα νέα δεδομένα. Υποθέτουμε, επίσης, ότι το επίπεδο εξόδου του μοντέλου-πηγή σχετίζεται με τις κλάσεις του συνόλου δεδομένου στο οποίο εκπαιδεύτηκε, και επομένως δεν χρησιμοποιείται στο μοντέλο-στόχος.

3. Πρόσθεση ενός επιπέδου εξόδου στο μοντέλο-στόχος, του οποίου οι έξοδοι είναι ο αριθμός των κλάσεων στο νέο σύνολο δεδομένων. Στη συνέχεια, οι παράμετροι του μοντέλου αυτού του επιπέδου αρχικοποιούνται τυχαία.

4. Εκπαίδευση του μοντέλου-στόχου στο νέο σύνολο δεδομένων. Το επίπεδο εξόδου θα εκπαιδευτεί από την αρχή, ενώ οι παράμετροι των υπόλοιπων επιπέδων θα γίνουν fine-tuned με βάση τις παραμέτρους του μοντέλου-πηγή. Δηλαδή, θα αλλάξουν λίγο οι τιμές των παραμέτρων με βάση τα νέα δεδομένα.



Εικόνα 12: Η τεχνική fine-tuning [39].

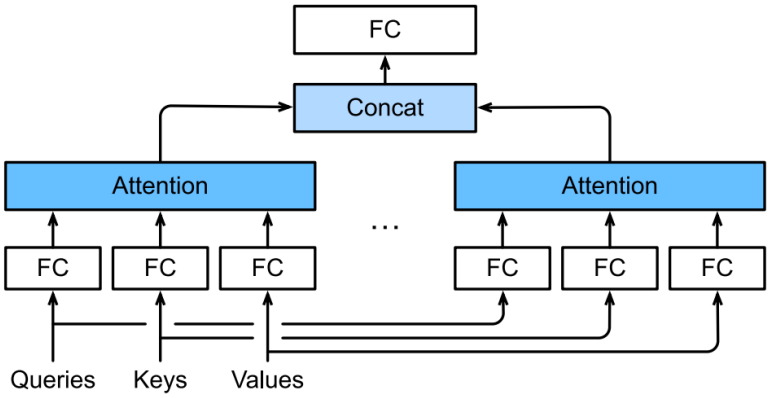
Σε αυτήν την ενότητα αναλύουμε το BERT, ένα μοντέλο που προτάθηκε πρόσφατα και πετυχαίνει υψηλή ακρίβεια σε προβλήματα κατανόησης της φυσικής γλώσσας, και κάποιες παραλλαγές του, που κάναμε fine-tuned και προσαρμόσαμε στα πειράματά μας.

3.9.1 BERT

Το BERT (Bidirectional Encoder Representations from Transformers) [43], σε αντίθεση με τα μοντέλα word embedding που αναθέτουν το ίδιο προ-εκπαιδευμένο διάνυσμα σε μία λέξη ασχέτως από τα συμφραζόμενα της λέξης, κωδικοποιεί τα συμφραζόμενα αμφίδρομα (bidirectionally), ενώ απαιτεί ελάχιστες αλλαγές στην αρχιτεκτονική για ένα μεγάλο εύρος προβλημάτων της Επεξεργασίας Φυσικής Γλώσσας. Υπάρχουν δύο βήματα σε αυτό το μοντέλο: η προ-εκπαίδευση (pretraining) και το fine-tuning. Κατά την προ-εκπαίδευση, το μοντέλο εκπαιδεύεται σε δεδομένα που δεν ανήκουν σε κάποια κλάση, για διάφορα προβλήματα Επεξεργασίας Φυσικής Γλώσσας. Στο fine-tuning, το μοντέλο BERT αρχικοποιείται αρχικά με τις προ-εκπαιδευμένες παραμέτρους, ενώ προστίθεται ένα επίπεδο εξόδου όπως στην τεχνική μεταφοράς γνώσης. Στη συνέχεια, οι παράμετροι του BERT γίνονται fine-tuned και το επίπεδο εξόδου εκπαιδεύεται από την αρχή. Το BERT όταν προτάθηκε πρώτη φορά βελτίωσε τα

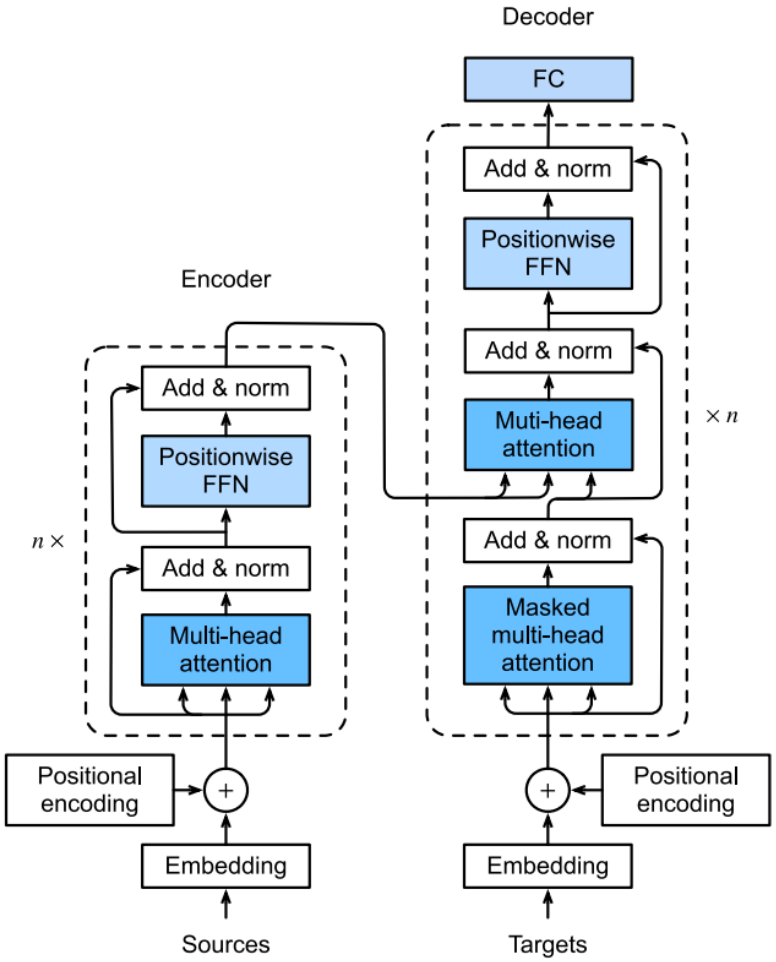
αποτελέσματα σε έντεκα προβλήματα Επεξεργασίας Φυσικής Γλώσσας που χρησιμοποιούνται για την αξιολόγηση των μοντέλων.

Το BERT χρησιμοποιεί τον transformer encoder ως την αμφίδρομη (bidirectional) αρχιτεκτονική. Το μοντέλο transformer [44] βασίζεται αποκλειστικά σε μηχανισμούς προσοχής και αποτελείται από έναν κωδικοποιητή (encoder) και έναν αποκωδικοποιητή (decoder). Η είσοδος στον transformer είναι embeddings αθροισμένα με την κωδικοποίηση θέσης (positional encoding), που έχει να κάνει με τη θέση που βρίσκεται μία λέξη στην πρόταση. Η είσοδος αυτή τροφοδοτείται στη συνέχεια στον κωδικοποιητή και στον αποκωδικοποιητή που αποτελούνται από στοίβες από επίπεδα βασισμένα στο self-attention. Συγκεκριμένα, ο κωδικοποιητής αποτελείται από μία στοίβα από πολλαπλά πανομοιότυπα επίπεδα, όπου το κάθε επίπεδο έχει δύο υπο-επίπεδα. Το πρώτο είναι ένα multi-head self-attention pooling και το δεύτερο είναι ένα positionwise feedforward network. Το multi-head attention συνδυάζει διαφορετικές συμπεριφορές από τον ίδιο μηχανισμό προσοχής. Δηλαδή, τα ερωτήματα, τα κλειδιά και οι είσοδοι μετασχηματίζονται από h πλήρως συνδεδεμένα επίπεδα, ενώ στη συνέχεια τα μετασχηματισμένα ερωτήματα, κλειδιά και είσοδοι τροφοδοτούνται σε παράλληλα επίπεδα ομαδοποίησης προσοχής (attention pooling). Στο τέλος, οι h έξοδοι συνενώνονται και μετασχηματίζονται με ένα άλλο πλήρως συνδεδεμένο επίπεδο για την παραγωγή της τελικής εξόδου, όπως φαίνεται στην παρακάτω εικόνα:



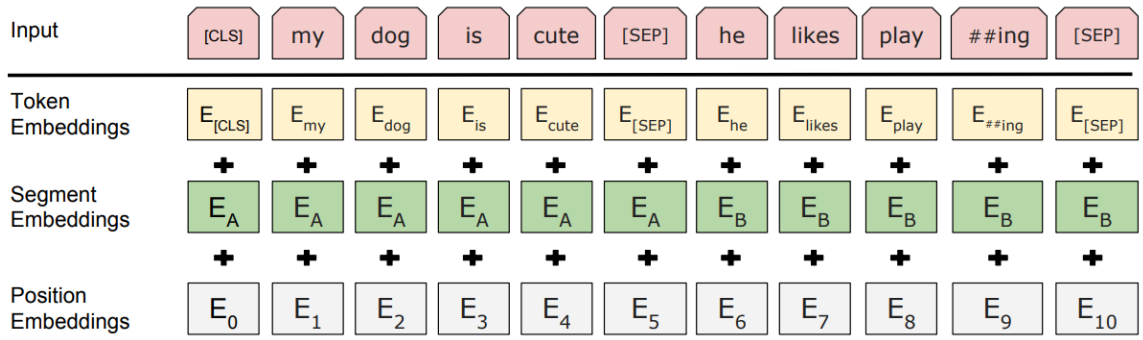
Εικόνα 13: Multi-head attention [39].

Το positionwise feedforward network αποτελείται από ένα MLP το οποίο εφαρμόζεται σε κάθε λέξη της πρότασης ανεξάρτητα και πανομοιότυπα. Στο self-attention του κωδικοποιητή, τα ερωτήματα, τα κλειδιά και οι είσοδοι είναι από την έξοδο του προηγούμενου επιπέδου κωδικοποιητή. Η είσοδος κάθε υπο-επιπέδου αθροίζεται με την έξοδο του αντίστοιχου υπο-επιπέδου, ενώ ακολουθεί μία κανονικοποίηση επιπέδου (layer normalization). Η λειτουργία αυτή αντιστοιχεί στο "κουτί" "add & norm" της παρακάτω εικόνας, όπου φαίνεται η αρχιτεκτονική του transformer. Η κανονικοποίηση επιπέδου μοιάζει με την κανονικοποίηση για το batch (batch normalization), όπου συνεχώς προσαρμόζεται η ενδιάμεση έξοδος ενός νευρωνικού δικτύου με τη χρήση του μέσου όρου και της τυπικής απόκλισης του batch, έτσι ώστε οι τιμές της ενδιάμεσης εξόδου του κάθε επιπέδου να είναι πιο σταθερές. Η κανονικοποίηση επιπέδου διαφέρει με την παραπάνω στο ότι ο μέσος όρος και η τυπική απόκλιση υπολογίζονται στην τελευταία διάσταση των τανυστών (axis=-1), και όχι στην διάσταση του batch (axis=0).



Εικόνα 14: Η αρχιτεκτονική του transformer [39].

Για να μπορεί το BERT να χειριστεί μία ποικιλία προβλημάτων, η αναπαράσταση της εισόδου μπορεί να αναπαραστήσει τόσο μία πρόταση όσο και ένα ζευγάρι προτάσεων σε μία ακολουθία. Μία ακολουθία εισόδου του BERT μπορεί να είναι, λοιπόν, μία πρόταση ή δύο προτάσεις μαζί. Για την αναπαράσταση των λέξεων χρησιμοποιήθηκαν τα WordPiece embeddings. Μία ακολουθία εισόδου του BERT είναι η συνένωση του ειδικού token "[CLS]", των tokens της πρώτης πρότασης, του ειδικού token "[SEP]", και στην περίπτωση που υπάρχει και δεύτερη πρόταση, των tokens της δεύτερης πρότασης και του ειδικού token "[SEP]". Η τελική κρυφή κατάσταση που αντιστοιχεί στο token "[CLS]" χρησιμοποιείται ως η συναθροισμένη αναπαράσταση της ακολουθίας σε προβλήματα ταξινόμησης. Για τον διαχωρισμό των προτάσεων, προσθέτονται τα εκπαιδευμένα segment embeddings E_A και E_B στα token embeddings της πρώτης και της δεύτερης πρότασης αντίστοιχα. Για τις ακολουθίες που περιλαμβάνουν μόνο μία πρόταση χρησιμοποιούνται μόνο τα E_A. Τελικά, για ένα token η αναπαράσταση της εισόδου κατασκευάζεται από την άθροιση των αντίστοιχων token, segment και position embeddings όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα 15: Η αναπαράσταση εισόδου του BERT. Τα embeddings εισόδου αποτελούν το άθροισμα των token embeddings, των segment embeddings και των position embeddings [43].

Τα δεδομένα που χρησιμοποιήθηκαν στην προ-εκπαίδευση του BERT είναι το BooksCorpus (800M λέξεις) και η αγγλική Wikipedia (2,500M λέξεις). Η προ-εκπαίδευση αποτελείται από δύο μέρη: το masked language modeling και την πρόβλεψη επόμενης πρότασης (next sentence prediction). Ένα γλωσσικό μοντέλο προβλέπει ένα token χρησιμοποιώντας τα συμφραζόμενα από τα αριστερά. Για την κωδικοποίηση των συμφραζομένων από τις δύο κατευθύνσεις (bidirectionally), το BERT τυχαία καλύπτει (masks) tokens και στη συνέχεια προσπαθεί να προβλέψει τα masked tokens. Αυτή η διαδικασία ονομάζεται masked language model (MLM). Σε αυτήν τη διαδικασία, το 15% των tokens θα επιλεχθούν τυχαία για να λειτουργήσουν ως τα masked tokens που θα προβλεφθούν. Για το σκοπό αυτό τα tokens που επιλέγονται αντικαθιστούνται από ένα ειδικό token: "[MASK]". Ωστόσο, το τεχνητό αυτό token δε θα εμφανιστεί ποτέ στη διαδικασία του fine-tuning. Για τη μετρίαση αυτού του προβλήματος, εάν ένα token έχει καλυφθεί για πρόβλεψη, τότε στην είσοδο αυτό το token θα αντικατασταθεί από:

- το ειδικό "[MASK]" token κατά 80% των περιπτώσεων.
- ένα τυχαίο token κατά 10% των περιπτώσεων.
- το ίδιο token, δηλαδή θα παραμείνει το ίδιο, κατά 10% των περιπτώσεων.

Ο περιστασιακός θόρυβος του να εισαχθεί ένα τυχαίο token ενθαρρύνει το BERT να είναι λιγότερο μεροληπτικό ως προς το masked token. Παρ'όλο όμως που το MLM μπορεί να κωδικοποιήσει τα συμφραζόμενα από τις δύο κατευθύνσεις για την αναπαράσταση λέξεων, δεν μπορεί να μοντελοποιήσει τη λογική σχέση μεταξύ δύο προτάσεων. Για να μπορέσει να καταλάβει τη σχέση μεταξύ δύο προτάσεων, το BERT εκτελεί μία δυαδική ταξινόμηση κατά την προ-εκπαίδευση, την πρόβλεψη επόμενης πρότασης. Όταν παράγονται ζευγάρια προτάσεων για την προ-εκπαίδευση, τις μισές φορές οι προτάσεις αυτές είναι όντως διαδοχικές προτάσεις και έχουν την ετικέτα "True", ενώ τις άλλες μισές φορές η δεύτερη πρόταση έχει επιλεχθεί τυχαία από το κείμενο και οι δύο προτάσεις έχουν την ετικέτα "False" [43] [39].

3.9.2 DistilBERT

Το DistilBERT [45] είναι ένα μικρό, γρήγορο και φθηνό μοντέλο βασισμένο στην αρχιτεκτονική του BERT, όπου χρησιμοποιήθηκε η τεχνική απόσταξης (distillation) κατά την προ-εκπαίδευση για τη μείωση του μεγέθους του μοντέλου BERT κατά 40%. Η τεχνική knowledge distillation είναι μία τεχνική συμπίεσης όπου το μοντέλο-μαθητής εκπαιδεύεται με σκοπό να αναπαραγάγει τη συμπεριφορά ενός μεγαλύτερου μοντέλου, του μοντέλου-δασκάλου. Κατά την εκπαίδευση δασκάλου-μαθητή, το μοντέλο-μαθητής εκπαιδεύεται με σκοπό να μιμηθεί την κατανομή εξόδου του μοντέλου-δασκάλου. Το

DistilBERT έχει την ίδια σχεδόν αρχιτεκτονική του BERT. Τα segment embeddings αφαιρέθηκαν, ενώ ο αριθμός των επιπέδων μειώθηκε κατά ένα συντελεστή 2. Επίσης, το DistilBERT εκπαιδεύτηκε με πολύ μεγάλα batches, με δυναμικό masking και χωρίς την πρόβλεψη επόμενης πρότασης, ενώ εκπαιδεύτηκε στα ίδια δεδομένα με το αυθεντικό BERT μοντέλο. Κατά το MLM, το BERT καλύπτει tokens μόνο κατά τη διάρκεια της προεπεξεργασίας των δεδομένων, που σημαίνει ότι οι ίδιες μάσκες εισόδου τροφοδοτούνται στο μοντέλο σε κάθε εποχή. Η διαδικασία αυτή ονομάζεται στατικό (static) masking. Στο DistilBERT κατά το MLM χρησιμοποιήθηκε δυναμικό (dynamic) masking, όπου το masking πραγματοποιείται κάθε φορά που μία ακολουθία τροφοδοτείται στο μοντέλο. Με αυτόν τον τρόπο το μοντέλο βλέπει διαφορετικές εκδόσεις της ίδιας πρότασης με μάσκες σε διαφορετικές τοποθεσίες. Συνολικά, το DistilBERT είναι 40% μικρότερο, 60% πιο γρήγορο και διατηρεί το 97% της ικανότητας της γλωσσικής κατανόησης του BERT [45].

3.9.3 DeBERTa

Το DeBERTa (Decoding-enhanced BERT with disentangled attention) [46] βελτιώνει το BERT με τη χρήση δύο νέων τεχνικών. Η πρώτη είναι ο ξεμπλεγμένος (disentangled) μηχανισμός προσοχής, όπου η κάθε λέξη αναπαριστάται με τη χρήση δύο διανυσμάτων που κωδικοποιούν το περιεχόμενο και τη θέση αντίστοιχα, ενώ τα βάρη του μηχανισμού προσοχής μεταξύ των λέξεων υπολογίζονται χρησιμοποιώντας ξεμπλεγμένους (disentangled) πίνακες στα περιεχόμενα και στις σχετικές θέσεις των λέξεων αντίστοιχα. Αντιθέτως από το BERT όπου η κάθε λέξη στο επίπεδο εισόδου αναπαριστάται με τη χρήση ενός διανύσματος, που είναι το άθροισμα του token embedding και του position embedding της λέξης, κάθε λέξη στο DeBERTa αναπαριστάται με τη χρήση δύο διανυσμάτων που κωδικοποιούν το περιεχόμενο και τη θέση ξεχωριστά. Καθώς πλέον η κάθε λέξη αναπαριστάται από δύο διανύσματα, η προσοχή ανάμεσα σε δύο λέξεις υπολογίζεται ως το άθροισμα τεσσάρων υπολογισμών προσοχής με τη χρήση ξεμπλεξμένων πινάκων στα περιεχόμενα και στις σχετικές θέσεις των λέξεων. Δηλαδή, προκύπτουν τέσσερις όροι: περιεχόμενο-προς-περιεχόμενο, περιεχόμενο-προς-θέση, θέση-προς-περιεχόμενο και θέση-προς-θέση. Ο όρος θέση-προς-θέση δεν προσφέρει κάποια επιπρόσθετη πληροφορία, καθώς το σχετικό position embedding μίας λέξης παραμένει το ίδιο, αφού περιέχει μόνο πληροφορία για τη θέση και όχι το περιεχόμενο, και επομένως αφαιρείται. Οπότε, τα βάρη προσοχής για ένα ζευγάρι λέξεων προκύπτουν από το άθροισμα των τριών αυτών διαφορετικών πινάκων. Η σχετική θέση των λέξεων τροφοδοτείται στο δίκτυο σε κάθε έξοδο ενός transformer επιπέδου και κάθε φορά γίνονται οι υπολογισμοί που αναφέραμε παραπάνω για όλες τις λέξεις. Με αυτήν τη τεχνική, αυξάνεται η πληροφορία που επεξεργάζεται το μοντέλο.

Η δεύτερη τεχνική που χρησιμοποιεί το DeBERTa είναι η ενσωμάτωση των απόλυτων position embeddings στο τελευταίο επίπεδο για την πρόβλεψη των masked tokens στην προ-εκπαίδευση του μοντέλου. Αυτό συμβαίνει ώστε να ενσωματωθούν στην πρόβλεψη και τα απόλυτα position embeddings των λέξεων που μερικές φορές είναι απαραίτητα. Συγκεκριμένα, ενσωματώνονται μετά από όλα τα transformer επίπεδα, αλλά πριν το τελευταίο επίπεδο που κάνει την πρόβλεψη των masked tokens. Με αυτόν τον τρόπο, το DeBERTa συλλαμβάνει τις σχετικές θέσεις σε όλα τα transformer επίπεδα και χρησιμοποιεί τις απόλυτες θέσεις ως συμπληρωματική πληροφορία κατά την πρόβλεψη των masked tokens. Αυτό το μέρος του DeBERTa ονομάζεται Enhanced Mask Decoder (EMD).

Το DeBERTa όπως και το DistilBERT, εκπαιδεύτηκε με πολύ μεγάλα batches, με δυναμικό masking και χωρίς την πρόβλεψη επόμενης πρότασης, ενώ σημειώνουμε ότι αυτές οι τεχνικές προτάθηκαν από μία άλλη παραλλαγή του BERT, το RoBERTa [47]. Όπως είναι προφανές από τις παραπάνω αλλαγές, το DeBERTa έχει περισσότερες παραμέτρους από το BERT. Συγκεκριμένα, υπάρχει μία άυξηση κατά 13% στις παραμέτρους, ενώ αυξάνει το υπολογιστικό κόστος κατά 30%. Επιπλέον, το DeBERTa εκπαιδεύτηκε σε περισσότερα δεδομένα. Τέλος, είχε καλύτερα αποτελέσματα σε αρκετά προβλήματα Επεξεργασίας Φυσικής Γλώσσας που χρησιμοποιούνται για την αξιολόγηση των μοντέλων

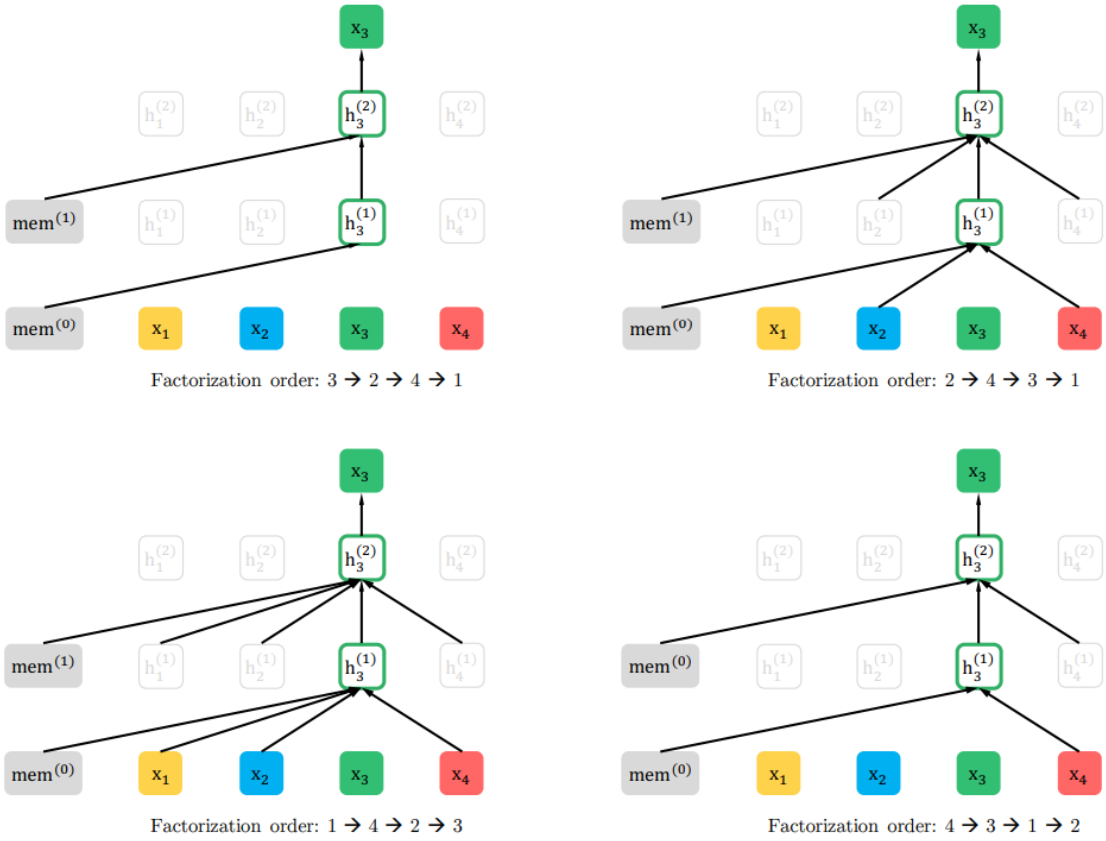
από το BERT ή άλλες παραλλαγές του [46].

3.9.4 XLNet

Το XLNet [48] είναι ένα γενικευμένο αυτοπαλίνδρομο (autoregressive) μοντέλο όπου το κάθε token εξαρτάται από όλα τα προηγούμενα tokens. Το μοντέλο αυτό είναι γενικευμένο καθώς συλλαμβάνει τα συμφραζόμενα με αμφίδρομο τρόπο (bidirectionally) με τη χρήση ενός μηχανισμού που ονομάζεται μεταθετική γλωσσική μοντελοποίηση (permutation language modeling). Ενσωματώνει την ιδέα των αυτοπαλίνδρομων μοντέλων και της αμφίδρομης μοντελοποίησης συμφραζομένων (bidirectional context modeling), ξεπερνώντας κάποια μειονεκτήματα του BERT. Έχει καλύτερα αποτελέσματα από το BERT σε πολλά προβλήματα Επεξεργασίας Φυσικής Γλώσσας που χρησιμοποιούνται για την αξιολόγηση των μοντέλων.

Ένα αυτοπαλίνδρομο (AR) μοντέλο προσπαθεί να εκτιμήσει την κατανομή πιθανότητας των επόμενων λέξεων με βάση τις προηγούμενες λέξεις, ενώ αυτό μπορεί να γίνει και προς τις δύο κατευθύνσεις ενός κειμένου. Ωστόσο, το μοντέλο αυτό μπορεί να κωδικοποιήσει τα συμφραζόμενα μόνο προς τη μία κατεύθυνση ξεχωριστά κάθε φορά. Το μοντέλο BERT, από την άλλη, χρησιμοποιεί autoencoding (AE), το οποίο δεν κάνει κάποια εκτίμηση πιθανότητας όπως το AR, αλλά προσπαθεί να κατασκευάσει τα αρχικά δεδομένα από την "κατεστραμμένη" είσοδο, δηλαδή προσπαθεί να βρει τα masked tokens. Ωστόσο, τα masked tokens απουσιάζουν από τα δεδομένα κατά το fine-tuning, κάτι που οδηγεί σε μία ασυμφωνία μεταξύ της προ-εκπαίδευσης και του fine-tuning. Επιπλέον, το BERT αντιμετωπίζει τα masked tokens ως ανεξάρτητα, οπότε δεν μαθαίνει πως μπορεί το ένα masked token να επηρεάσει ένα άλλο. Το XLNet είναι μία προσπάθεια χρήσης και των δύο τεχνικών με σκοπό να αποφευχθούν οι παραπάνω περιορισμοί.

Η μεταθετική γλωσσική μοντελοποίηση (permutation language modeling, PLM) συλλαμβάνει τα συμφραζόμενα με αμφίδρομο τρόπο εκπαιδεύοντας ένα αυτοπαλίνδρομο μοντέλο σε όλες τις πιθανές μεταθέσεις των λέξεων σε μία πρόταση. Συγκεκριμένα, το XLNet μεγιστοποιεί την αναμενόμενη πιθανότητα από όλες τις πιθανές μεταθέσεις λέξεων της πρότασης. Με αυτόν τον τρόπο, κάθε θέση μαθαίνει πληροφορίες από όλες τις άλλες θέσεις, συλλαμβάνοντας συνεπώς τα συμφραζόμενα με αμφίδρομο τρόπο, ενώ το ειδικό token "[MASK]" δεν χρειάζεται πλέον, καθώς τα δεδομένα εισόδου δεν αλλοιώνονται. Ένα παράδειγμα του PLM φαίνεται στην παρακάτω εικόνα όπου προσπαθούμε να μάθουμε το token x_3. Το PLM εκπαιδεύει ένα αυτοπαλίνδρομο μοντέλο για διαφορετικές μεταθέσεις των tokens στην πρόταση, ώστε στο τέλος να έχει μάθει το token x_3, δεδομένου όλων των υπόλοιπων tokens στην πρόταση. Επίσης, από την ακόλουθη εικόνα φαίνεται ότι το επόμενο επίπεδο παίρνει ως είσοδο μόνο τα tokens που προηγούνται του x_3 σε κάθε ακολουθία μετάθεσης, καθώς με αυτό τον τρόπο επιτυγχάνεται η αυτοπαλινδρόμηση.



Εικόνα 16: Παράδειγμα της μεταθετικής γλωσσικής μοντελοποίησης (permutation language modeling) για την πρόβλεψη του x_3, δεδομένου της ίδιας ακολουθίας εισόδου x αλλά με διαφορετικές ακολουθίες μετάθεσης [48].

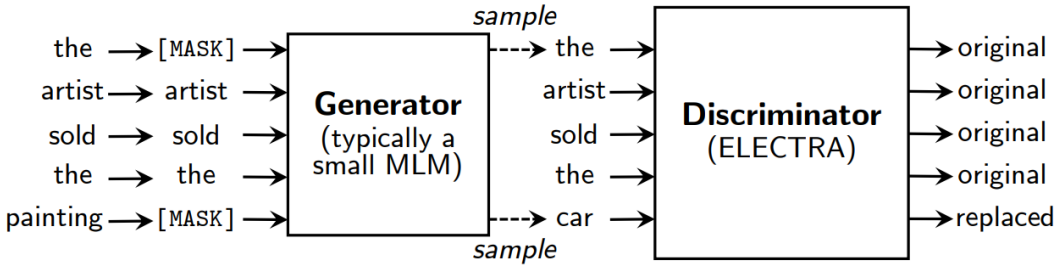
Το XLNet χρησιμοποιεί το μοντέλο transformer που χρησιμοποιεί και το BERT, αλλά με κάποιες αλλαγές, ώστε να επιτυγχάνεται η αυτοπαλινδρόμηση. Δηλαδή, να μπορεί να δει μόνο τις αναπαραστάσεις των tokens που προηγούνται του token που θέλουμε να προβλέψουμε κάθε φορά. Επιπλέον, όπως και στο BERT, στα token embeddings προστίθενται τα position embeddings. Ωστόσο, η πρόσθεση της πληροφορίας για τη θέση των tokens δεν θα είχε νόημα εάν ανακατευόντουσαν όντως τα tokens της πρότασης. Το πρόβλημα αυτό λύνεται με τη χρήση ενός πίνακα με μάσκες προσοχής (attention mask), που απλά δείχνει ποιες λέξεις κάθε φορά περιλαμβάνονται στα συμφραζόμενα που εξετάζονται. Δηλαδή, οι λέξεις της ακολουθίας μετάθεσης διαβάζονται με τη σειρά που εμφανίζονται στην κανονική πρόταση. Επιπρόσθετα, το XLNet χρησιμοποιεί ένα μηχανισμό επανάληψης (recurrency), με αποτέλεσμα να μπορεί να χρησιμοποιήσει μνήμη από προηγούμενους υπολογισμούς του PLM. Το XLNet εκπαιδεύτηκε σε περισσότερα δεδομένα από το BERT, ενώ το μέγεθός του είναι παρόμοιο με αυτό του BERT. Τέλος, το XLNet δεν χρησιμοποιεί την πρόβλεψη επόμενης πρότασης κατά την προ-εκπαίδευση, ενώ η διαδικασία fine-tuning ακολουθεί την αντίστοιχη διαδικασία του BERT [48].

3.9.5 ELECTRA

Το ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) [49] προτείνει μία νέα μέθοδο προ-εκπαίδευσης που έχει ως σκόπο να ξεπεράσει την απόδοση ενός μοντέλου που έχει εκπαιδευτεί με τη μέθοδο MLM χρησιμοποιώντας παράλληλα λιγότερους υπ-

ολογιστικούς πόρους για το στάδιο της προ-εκπαίδευσης. Συγκεκριμένα, οι συγγραφείς προτείνουν τη μέθοδο ανίχνευσης ενός token που έχει αντικατασταθεί (replaced token detection), κατά την οποία το μοντέλο μαθαίνει να διαχωρίζει τα tokens της εισόδου από πιθανά "ψεύτικα" tokens. Αντί να καλύπτει κάποια tokens, η μέθοδος αυτή αντικαθιστά κάποια tokens της εισόδου με άλλα tokens, όπου τα ψεύτικα tokens είναι η έξοδος ενός μικρού MLM μοντέλου. Η διαδικασία αυτή έχει το μειονέκτημα ότι υπάρχει μία ασυμφωνία μεταξύ της προ-εκπαίδευσης και του fine-tuning, όπως συμβαίνει στο BERT. Το μοντέλο προ-εκπαιδεύεται ως ένας διευκρινιστής (discriminator), που προβλέπει για κάθε token εάν είναι το αυθεντικό ή το ψεύτικο. Αντίθετα, το μικρό MLM μοντέλο εκπαιδεύεται ως ένας γεννήτορας (generator) που προσπαθεί να μαντέψει τις πραγματικές ταυτότητες των "κατεστραμμένων" tokens. Ένα πλεονέκτημα αυτής της μεθόδου είναι ότι το μοντέλο μαθαίνει από όλα τα tokens της εισόδου αντί από το μικρό υποσύνολο των masked tokens που μαθαίνει το BERT. Για την εκπαίδευση, το ELECTRA χρησιμοποιεί transformer encoders που μπορούν να γίνουν fine-tuned για πολλά διαφορετικά προβλήματα.

Η μέθοδος που ακολουθεί το ELECTRA εκπαιδεύει δύο νευρωνικά δίκτυα, έναν γεννήτορα και έναν διευκρινιστή. Το καθένα από αυτά τα δίκτυα αποτελείται από transformer encoders που αντιστοιχούν μία ακολουθία από tokens εισόδου x σε μία ακολουθία αναπαραστάσεων των συμφραζομένων $h(x)$. Για μία θέση t, ο γεννήτορας παράγει την πιθανότητα παραγωγής του token x_t. Για μία θέση t, ο διευκρινιστής προβλέπει εάν το token x_t είναι ψεύτικο. Ο γεννήτορας είναι εκπαιδευμένος για να εκτελεί τη διαδιακασία MLM. Δηλαδή, επιλέγονται τυχαία κάποια tokens για να καλυφθούν με το ειδικό "[MASK]" token και ο γεννήτορας μαθαίνει να μεγιστοποιεί την πιθανότητα πρόβλεψης των masked tokens. Ο διευκρινιστής εκπαιδεύεται ώστε να διαχωρίζει τα tokens της εισόδου από τα tokens που έχουν τοποθετηθεί από τον γεννήτορα. Παρόλο που η μέθοδος ανίχνευσης ενός token που έχει αντικατασταθεί θυμίζει τον τρόπο εκπαίδευσης ενός GAN (Generative Adversarial Network), υπάρχουν κάποιες βασικές διαφορές. Αρχικά, άμα ο γεννήτορας τυγχάνει το σωστό token, τότε αυτό το token θεωρείται πραγματικό και όχι ψεύτικο. Επίσης, ο γεννήτορας εκπαιδεύεται με σκοπό να βελτιώσει την πιθανότητα της πρόβλεψης και όχι για να ξεγελάσει τον διευκρινιστή. Μετά την προ-εκπαίδευση, ο γεννήτορας απομακρύνεται και χρησιμοποιείται η τεχνική fine-tuning μόνο στον διευκρινιστή. Η μέθοδος που περιγράψαμε φαίνεται στην παρακάτω εικόνα:



Εικόνα 17: Η μέθοδος ανίχνευσης ενός token που έχει αντικατασταθεί (replaced token detection) [49].

Το ELECTRA είναι υπολογιστικά πιο αποδοτικό από το BERT και το XLNET, καθώς πετυχαίνει παρόμοια ή καλύτερα αποτελέσματα σε πολλά προβλήματα Επεξεργασίας Φυσικής Γλώσσας που χρησιμοποιούνται για την αξιολόγηση των μοντέλων με λιγότερο χρόνο εκπαίδευσης, από ότι χρειάζονται τα άλλα δύο μοντέλα. Επιπλέον, το ELECTRA εκπαιδεύτηκε με δυναμικό masking και χωρίς την πρόβλεψη επόμενης πρότασης. Τέλος, εκπαιδεύτηκε στα δεδομένα που εκπαιδεύτηκε και το XLNet, ενώ ο διευκρινιστής έχει το ίδιο μέγεθος με το BERT [49].

3.10 Μέθοδοι Αξιολόγησης

Σε αυτήν την ενότητα αναφέρουμε τις μεθόδους αξιολόγησης που χρησιμοποιήσαμε για τα μοντέλα. Συγκεκριμένα, αξιολογήσαμε τα μοντέλα με τις μετρικές accuracy, ανάκληση (recall), ακρίβεια (precision), F1 score και πίνακα σύγχυσης (confusion matrix). Στους ακόλουθους ορισμούς ορίζουμε ως TP (true positive) τον αριθμό των σωστών προβλέψεων μιας κλάσης, της "κλάσης 1", ως TN (true negative) το πλήθος των σωστών προβλέψεων της άλλης κλάσης, της "κλάσης 0", ως FP (false positive) τον αριθμό των λανθασμένων προβλέψεων της "κλάσης 0" που κατηγοριοποιήθηκαν ως "κλάση 1" και ως FN (false negative) το πλήθος των λανθασμένων προβλέψεων της "κλάσης 1" που κατηγοριοποιήθηκαν ως "κλάση 0".

Το μέγεθος accuracy ισούται με το πηλίκο του πλήθους των συνολικών σωστών προβλέψεων του ταξινομητή προς το συνολικό πλήθος των προβλέψεων. Δηλαδή:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Η ανάκληση (recall) ορίζεται ως τον λόγο του πλήθους των σωστών προβλέψεων της "κλάσης 1" προς τον αριθμό των πραγματικών παρατηρήσεων που ανήκουν στην "κλάση 1":

$$Recall = \frac{TP}{TP + FN}.$$

Η ακρίβεια (precision) ισούται με τον λόγο του πλήθους των σωστών προβλέψεων της "κλάσης 1" προς τον αριθμό των παρατηρήσεων που ταξινομήθηκαν στην "κλάση 1":

$$Precision = \frac{TP}{TP + FP}.$$

Η ανάκληση ευνοεί τα μοντέλα που επιτυχώς ταξινομούν τις περισσότερες παρατηρήσεις που ανήκουν στην "κλάση 1", ασχέτως του αριθμού των παρατηρήσεων της "κλάσης 0" που ταξινομήθηκαν ως "κλάση 1". Αντιθέτως, η ακρίβεια ευνοεί τους ταξινομητές που επιλεκτικά ταξινομούν παρατηρήσεις ως "κλάση 1". Σαφώς ένας καλός ταξινομητής ανιχνεύει με ακρίβεια ένα μεγάλο ποσοστό των παρατηρήσεων που ανήκουν στην "κλάση 1", ενώ η μετρική που το δείχνει αυτό είναι η F1 score, που ορίζεται ως τον αρμονικό μέσο των μετρικών precision και recall [31]:

$$F1\ score = 2 \cdot \frac{precision \times recall}{precision + recall}.$$

Ο πίνακας σύγχυσης (confusion matrix) C είναι ένας $k \times k$ πίνακας όπου η κάθε γραμμή αντικατοπτρίζει τις πραγματικές κλάσεις και η κάθε στήλη αντιστοιχεί στις κλάσεις που προβλέπει το μοντέλο. Κάθε στοιχείο c_{ij} του πίνακα είναι το κλάσμα των συνολικών προβλέψεων στα δεδομένα, όπου η πραγματική κλάση ήταν η i και η κλάση που πρόβλεψε το μοντέλο ήταν η j. Τα στοιχεία της διαγωνίου έχουν ταξινομηθεί σωστά για την κάθε κλάση, ενώ αυτός ο τρόπος αξιολόγησης είναι ένας τρόπος εντοπισμού του ενδεχομένου το μοντέλο να μπερδεύει κάποιες κλάσεις μεταξύ τους [39].

4 Μεθοδολογία

4.1 Δεδομένα

Χρησιμοποιούμε δεδομένα από πολλαπλές πηγές: CrisisLex, CrisisNLP και Disaster Tweet Corpus 2020. Από τη συλλογή CrisisLex, χρησιμοποιούμε το σύνολο δεδομένων CrisisLexT6, το οποίο περιλαμβάνει tweets στα αγγλικά από έξι μεγάλες καταστροφές που έγιναν το 2012 και το 2013. Περιλαμβάνονται περίπου 10,000 tweets για το κάθε γεγονός, ενώ τα tweets είναι ταξινομημένα από ανθρώπους με βάση τη σχετικότητά τους με την αντίστοιχη καταστροφή ("on-topic" ή "off-topic"). Συγκεκριμένα περιλαμβάνει τις καταστροφές: τυφώνας Σάντι (Sandy) του 2012, πλημμύρες στην Αλμπέρτα (Alberta) του 2013, βομβιστική επίθεση στη Βοστώνη (Boston) του 2013 (κοινωνική καταστροφή), σίφωνας στην Οκλαχόμα (Oklahoma) του 2013, πλημμύρες στο Κουίνσλαντ (Queensland) του 2013 και έκρηξη σε εργοστάσιο στο δυτικό Τέξας (Texas) του 2013 (βιομηχανική καταστροφή) [4].

Από τη συλλογή CrisisNLP χρησιμοποιούμε το σύνολο δεδομένων #7, που αποτελείται από tweets που συγκεντρώθηκαν από τον σεισμό στο Νεπάλ (Nepal) του 2015 και τις πλημμύρες στο Κουίνσλαντ του 2013. Τα tweets ταξινομήθηκαν από ανθρώπους με βάση τη σχετικότητα του κάθε tweet ως προς την αντίστοιχη καταστροφή. Σημειώνουμε ότι από αυτό το σύνολο δεδομένων χρησιμοποιούμε μόνο τα δεδομένα από τον σεισμό στο Νεπάλ [50].

Η συλλογή Disaster Tweet Corpus 2020 περιλαμβάνει tweets από 48 καταστροφές για 10 τύπους καταστροφών, που ταξινομήθηκαν από ανθρώπους με βάση τη σχετικότητά τους με την αντίστοιχη καταστροφή. Από αυτή τη συλλογή επιλέγουμε τα σύνολα δεδομένων που αφορούν τους τύπους καταστροφών: σεισμό, πλημμύρα, τυφώνα, σίφωνα, βιομηχανικές καταστροφές και κοινωνικές καταστροφές. Δηλαδή επιλέγουμε τα σύνολα δεδομένων: σεισμός στο νησί Μποχόλ (Bohol) στις Φιλιππίνες του 2013, σεισμός στην Καλιφόρνια (California) του 2013, σεισμός στη Χιλή (Chile) του 2013, σεισμός στην Κόστα Ρίκα (Costa rica) του 2012, σεισμός στη Γουατεμάλα (Guatemala) του 2012, σεισμός στο Ιράν (Iran) και Ιράκ (Iraq) του 2017, σεισμός στην Ιταλία (Italy) του 2012, σεισμός στο Μεξικό (Mexico) του 2017, σεισμός στο Νεπάλ του 2018, σεισμός στο Πακιστάν (Pakistan) του 2013, πλημμύρες στο Κολοράντο (Colorado) του 2013, πλημμύρες στην Ινδία (India) του 2014, πλημμύρες στη Μανίλα (Manila), πλημμύρες στο Πακιστάν το 2014, πλημμύρες στις Φιλιππίνες (Philipinnes) του 2012, πλημμύρες στη Σαρδηνία (Sardinia) του 2012, πλημμύρες στη Σρι Λάνκα (Sri Lanka) του 2017, τυφώνας Χαγκοπίτ (Hagupit) του 2014, τυφώνας Χάρβεϊ (Harvey) του 2017, τυφώνας Ίρμα (Irma) του 2017, τυφώνας Μαρία (Maria) του 2017, τυφώνας Οντίλ (Odile) του 2014, τυφώνας Pablo του 2012, τυφώνας Παμ (Pam) του 2015, τυφώνας Γιολάντα (Yolanda) του 2013, κατάρρευση κτιρίου στη Σάβαρ (Savar) του 2013 (βιομηχανική καταστροφή), φωτιά σε διυλιστήριο στη Βενεζουέλα (Venezuela) του 2012 (βιομηχανική καταστροφή), φωτιά σε κλαμπ στη Βραζιλία (Brazil) του 2013 (κοινωνική καταστροφή), πυροβολισμοί στο αεροδρόμιο του Λος Άντζελες (Los Angeles) του 2013 (κοινωνική καταστροφή) και σίφωνας Τζόπλιν (Joplin) του 2011 [2].

Συνδυάζουμε τους παρόμοιους τύπους καταστροφών σε μεγαλύτερα σύνολα δεδομένων. Συγκεκριμένα, συνδυάζουμε όλους τους τυφώνες με όλους τους σίφωνες, όλες τις πλημμύρες, όλους τους σεισμούς, όλες τις βιομηχανικές καταστροφές και όλες τις κοινωνικές καταστροφές. Συνεπώς, προκύπτουν πέντε σύνολα δεδομένων: οι καταιγίδες, οι πλημμύρες, οι σεισμοί, οι βιομηχανικές καταστροφές και οι κοινωνικές καταστροφές αντίστοιχα. Από αυτά τα σύνολα δεδομένων αφαιρούμε τα tweets που εμφανίζονται δύο ή παραπάνω φορές. Τέλος, το κάθε σύνολο δεδομένων το διαχωρίζουμε με τυχαίο τρόπο σε δεδομένα εκπαίδευσης, που αποτελούν το 80% των δεδομένων, και σε δεδομένα επαλήθευσης, που αποτελούν το 20% των δεδομένων. Στον παρακάτω πίνακα βλέπουμε τον αριθμό των tweets που περιλαμβάνονται στο κάθε σύνολο δεδομένων και στην κάθε κλάση, πριν την προ-επεξεργασία:

Τύπος Καταστροφής	Αριθμός σχετικών tweets	Αριθμός μη-σχετικών tweets	Σύνολο
Καταιγίδες	29794	28673	58467
Πλημμύρες	15534	15849	31383
Σεισμοί	19036	19419	38455
Βιομηχανικές Καταστροφές	5029	5537	10566
Κοινωνικές Καταστροφές	5937	5396	11333

Πίνακας 1: Ο αριθμός των tweets που περιλαμβάνονται στο κάθε σύνολο δεδομένων και στην κάθε κλάση, πριν την προ-επεξεργασία.

4.1.1 Προ-επεξεργασία των Δεδομένων

Προ-επεξεργαζόμαστε τα tweets, έτσι ώστε ο αλγόριθμος βαθιάς μηχανικής μάθησης στο επόμενο στάδιο να μπορεί να κατανοήσει τα δεδομένα. Το κείμενο από τα tweets είναι εκ φύσεως θορυβώδες, οπότε ο καθαρισμός τους είναι απαραίτητος. Εκτελούμε τα ακόλουθα βήματα προ-επεξεργασίας:

- Μέσω της βιβλιοθήκης tweet-preprocessor αφαιρούμε τα urls, τα hashtags, τις αναφορές σε άλλους χρήστες, τα emojis και τα smileys.
- Μετατρέπουμε όλα τα γράμματα σε πεζά.
- Αφαιρούμε σύμβολα, αριθμούς και σημεία στίξης.
- Αφαιρούμε τα stop words μέσω της βιβλιοθήκης nltk.

Μετά την προ-επεξεργασία αφαιρούμε τα κενά tweets που προκύπτουν, καθώς και τα tweets που περιλαμβάνουν μόνο μία λέξη, ώστε να μειωθεί ο θόρυβος στα δεδομένα. Ο αριθμός των tweets που περιλαμβάνονται τελικά στο κάθε σύνολο δεδομένων και στην κάθε κλάση μετά την προ-επεξεργασία φαίνεται στον παρακάτω πίνακα. Όπως ήταν λογικό, οι αριθμοί έχουν μειωθεί λίγο, αλλά παρατηρούμε ότι όπως και πριν έχουμε ισόνομη κατανομή των tweets στις δύο κλάσεις.

Τύπος Καταστροφής	Αριθμός σχετικών tweets	Αριθμός μη-σχετικών tweets	Σύνολο
Καταιγίδες	29756	27493	57249
Πλημμύρες	15495	15378	30873
Σεισμοί	18813	18339	37152
Βιομηχανικές Καταστροφές	5018	5452	10470
Κοινωνικές Καταστροφές	5920	5287	11207

Πίνακας 2: Ο αριθμός των tweets που περιλαμβάνονται στο κάθε σύνολο δεδομένων και στην κάθε κλάση, μετά την προ-επεξεργασία.

Σημειώνουμε ότι την παραπάνω προ-επεξεργασία την πραγματοποιούμε για όλα τα μοντέλα, δηλαδή και για τα νευρωνικά δίκτυα και για τα προ-εκπαιδευμένα γλωσσικά μοντέλα.

4.1.2 Επιλογή Μήκους Προτάσεων

Μετά την προ-επεξεργασία, επιλέγουμε το μήκος που θα έχουν οι προτάσεις στα μοντέλα, καθώς πρέπει όλες να έχουν το ίδιο μήκος. Αυτό απαιτείται για να μπορούν να εκτελεστούν πράξεις γραμμικής άλγεβρας, όπως πολλαπλασιασμός πινάκων. Για τον λόγο αυτό επιλέγουμε ένα μέγιστο μήκος προτάσεων και συμπληρώνουμε με μηδενικά στοιχεία τις προτάσεις μικρότερου μήκους (zero-padding) ή αφαιρούμε τις λέξεις που ξεπερνούν τα επιλεγμένο μήκος. Για το κάθε σύνολο δεδομένων τυπώνουμε

την κατανομή και επιλέγουμε το μήκος που καλύπτει την πλειοψηφία των προτάσεων. Σε όλα τα σύνολα δεδομένων, ο μέσος όρος μήκους των προτάσεων είναι γύρω στις 8 λέξεις. Αιθαίρετα επιλέγουμε ότι το ποσοστό πλειοψηφίας των προτάσεων που θα καλυφθούν ολόκληρες είναι το 85%. Θέτοντας αυτό το κάτω όριο, παρατηρούμε ότι τουλάχιστον το 85% των tweets θα είναι ολόκληρα εάν κρατήσουμε 12 λέξεις, ενώ αυτό το αποτέλεσμα ήταν το ίδιο για όλα τα σύνολα δεδομένων. Σημειώνουμε ότι αυτό το μήκος λέξεων το χρησιμοποιούμε μόνο για τα νευρωνικά δίκτυα, καθώς με το tokenization στα μοντέλα τύπου BERT προκύπτουν περισσότερες λέξεις, οπότε χρειάζεται να θέσουμε μεγαλύτερο μέγιστο μήκος των προτάσεων. Σε αυτές τις περιπτώσεις ως μέγιστο μήκος των προτάσεων θέτουμε τις 20 λέξεις, από τις οποίες οι 2 λέξεις είναι τα ειδικά tokens που προσθέτει ο κάθε tokenizer.

4.1.3 Tokenization

Το επόμενο βήμα μετά την προ-επεξεργασία είναι το tokenization. Κάνουμε tokenization με τη χρήση του TweetTokenizer της βιβλιοθήκης nltk για τα νευρωνικά δίκτυα, ενώ στα μοντέλα τύπου BERT, το κάθε μοντέλο εχει τον δικό του tokenizer, από τη βιβλιοθήκη transformers [6]. Οι tokenizers αυτοί προσθέτουν και τα ειδικά tokens στην αρχή και στο τέλος της κάθε πρότασης. Στον παρακάτω πίνακα βλέπουμε ένα παράδειγμα tokenization του BertTokenizer όπου οι αριθμοί 101 και 102 αντιστοιχούν στα ειδικά tokens "[CLS]" και "[SEP]". Στην πρώτη γραμμή του πίνακα φαίνεται ένα προ-επεξεργασμένο tweet από το σύνολο δεδομένων με τις κοινωνικές καταστροφές. Στη δεύτερη γραμμή φαίνεται το tokenization του συγκεκριμένου tweet, ενώ στην τελευταία γραμμή φαίνεται η μετατροπή των tokens σε αριθμούς, με τα embeddings του BERT. Συγκεκριμένα, βλέπουμε τον τανυστή που θα χρησιμοποιηθεί ως είσοδος στο μοντέλο BERT για το συγκεκριμένο tweet.

boston bombing suspect loose police say
['boston', 'bombing', 'suspect', 'loose', 'police', 'say']
[101, 3731, 8647, 8343, 6065, 2610, 2360, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Πίνακας 3: Ένα παράδειγμα tokenization του BertTokenizer ενός tweet από το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Στον επόμενο πίνακα βλέπουμε αντίστοιχα παραδείγματα tokenization για τα μοντέλα DistilBERT, DeBERTa, XLNet και ELECTRA με τους DistilBertTokenizer, DebertaTokenizer, XLNetTokenizer και ElectraTokenizer αντίστοιχα, από τη βιβλιοθήκη transformers. Στα μοντέλα DistilBERT και ELECTRA, τα ειδικά tokens "[CLS]" και "[SEP]" συμβολίζονται όπως και στο BERT με τους αριθμούς 101 και 102. Αντίθετα στο DeBERTa συμβολίζονται με τους αριθμούς 1 και 2, ενώ στο XLNet χρησιμοποιούνται δύο ειδικά tokens που μπαίνουν στο τέλος της πρότασης και συμβολίζονται με τους αριθμούς 4 και 3. Τα tweets που παρουσιάζονται στα παρακάτω παραδείγματα είναι προ-επεξεργασμένα και προέρχονται από το σύνολο δεδομένων με τις πλημμύρες, από το σύνολο δεδομένων με τις καταιγίδες, από το σύνολο δεδομένων με τους σεισμούς και από το σύνολο δεδομένων με τις βιομηχανικές καταστροφές αντίστοιχα. Σημειώνουμε ότι τα tweets που χρησιμοποιήθηκαν σε όλα τα παραδείγματα είναι σχετικά με την αντίστοιχη καταστροφή.

[6] https://huggingface.co/transformers/

Μοντέλο	Παράδειγμα Tokenization
DistilBERT	queensland flood crisis claimed four lives ['queensland', 'flood', 'crisis', 'claimed', 'four', 'lives'] [101, 5322, 7186, 5325, 3555, 2176, 3268, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
DeBERTa	hurricane caused new yorks time square almost look like ghost town ['hur', 'ricane', 'Ġcaused', 'Ġnew', 'Ġy', 'orks', 'Ġtime', 'Ġsquare', 'Ġalmost', 'Ġlook', 'Ġlike', 'Ġghost', 'Ġtown'] [1, 14898, 33280, 1726, 92, 1423, 41250, 86, 3925, 818, 356, 101, 15934, 1139, 2, 0, 0, 0, 0, 0]
XLNet	chile im praying youve getting many earthquakes please stay safe heart prayforchile ['_', 'chi', 'le', '_im', '_praying', '_you', 've', '_getting', '_many', '_earthquake', 's', '_please', '_stay', '_safe', '_heart', '_pray', 'for', 'chi', 'le'] [17, 2416, 529, 7693, 17909, 44, 189, 723, 142, 5270, 23, 1282, 1078, 1686, 758, 9454, 1383, 2416, 4, 3]
ELECTRA	condolences loss life following explosion chemical plant west texas ['condo', '##lence', '##s', 'loss', 'life', 'following', 'explosion', 'chemical', 'plant', 'west', 'texas'] [101, 25805, 22717, 2015, 3279, 2166, 2206, 7738, 5072, 3269, 2225, 3146, 102, 0, 0, 0, 0, 0, 0, 0]

Πίνακας 4: Παραδείγματα tokenization των DistilBertTokenizer, DebertaTokenizer, XLNetTokenizer και ElectraTokenizer αντίστοιχα.

4.1.4 Word Embedding

Ως word embedding χρησιμοποιούμε τα GloVe twitter embeddings, τα οποία έχουν εκπαιδευτεί πάνω σε δεδομένα από το Twitter. Τα GloVe embeddings είναι διαθέσιμα δημοσίως και τα συγκεκριμένα που χρησιμοποιούμε έχουν εκπαιδευτεί σε 2 δισεκατομμύρια tweets, δηλαδή σε 27 δισεκατομμύρια λέξεις, ενώ είναι διαθέσιμα στις διαστάσεις 25, 50, 100 και 200 [7]. Επιλέγουμε να χρησιμοποιήσουμε τη διάσταση 50 έτσι ώστε κατά την εκπαίδευση των δικτύων να υπάρχει μικρότερος υπολογιστικός φόρτος. Σημειώνουμε ότι αυτά τα embeddings χρησιμοποιούνται μόνο για τα νευρωνικά δίκτυα, καθώς το κάθε μοντέλο τύπου BERT έχει τα δικά του embeddings, όπως φάνηκε και στην παραπάνω υποενότητα.

4.2 Δομές Νευρωνικών Δικτύων

Σε αυτήν την ενότητα αναλύουμε τη δομή των νευρωνικών δικτύων που χρησιμοποιούμε για την πρόβλεψη των σχετικών tweets ανά καταστροφή. Επιπλέον, αναφερόμαστε στη βελτιστοποίηση των υπερ-παραμέτρων, που πραγματοποιήθηκε με τη βιβλιοθήκη Optuna. Σημειώνουμε ότι ο καλύτερος συνδυασμός υπερ-παραμέτρων επιλέχθηκε σε όλα τα μοντέλα με βάση το accuracy στα δεδομένα επαλήθευσης, ενώ εξετάστηκαν κάθε φορά 100 διαφορετικοί συνδυασμοί τιμών των υπερ-παραμέτρων. Επιπλέον, ο καθορισμός των εποχών σε όλα τα μοντέλα έγινε με τον πρόωρο τερματισμό (early stopping), όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 150 και ο προκαθορισμένος αριθμός εποχών όπου δεν παρουσιάζει βελτίωση το μοντέλο στα δεδομένα επαλήθευσης (patience) τέθηκε ίσος με 10. Το κριτήριο με το οποίο ελέγχει ο πρόωρος τερματισμός για το ενδεχόμενο βελτίωσης είναι το σφάλμα (loss) στα δεδομένα επαλήθευσης.

[7] https://nlp.stanford.edu/projects/glove/

4.2.1 Μοντέλα Βασισμένα στο MLP

Αρχικά, δημιουργούμε δύο MLP μοντέλα. Η διαφορά ανάμεσα σε αυτά τα δύο μοντέλα είναι ο τρόπος με τον οποίο συνδυάζονται τα word embeddings πριν την είσοδό τους στα μοντέλα. Στο πρώτο μοντέλο παίρνουμε τον μέσο όρο των embeddings των λέξεων, για κάθε στοιχείο του τανυστή, σε κάθε tweet ξεχωριστά. Συνεπώς, η αναπαράσταση για κάθε πρόταση θα έχει το ίδιο μέγεθος με το embedding μιας λέξης, δηλαδή 50 στη δική μας περίπτωση. Στο δεύτερο μοντέλο, εκτός από τον μέσο όρο, παίρνουμε την ελάχιστη και τη μέγιστη τιμή από όλες τις λέξεις ενός tweet για κάθε στοιχείο του τανυστή. Κάθε συσσωμάτωση (min/max/mean) παράγει έναν τανυστή που έχει την ίδια διάσταση με το embedding μιας λέξης. Στη συνέχεια, συνενώνουμε τους τρεις τανυστές που προκύπτουν για το κάθε tweet και προκύπτει μία αναπαράσταση πρότασης, της οποίας η διάσταση είναι τρεις φορές η διάσταση του embedding μιας λέξης, δηλαδή 150 στη δική μας περίπτωση [51]. Συνεπώς το πρώτο μοντέλο έχει μέγεθος εισόδου 50 και το δεύτερο έχει μέγεθος εισόδου 150. Και τα δύο MLP μοντέλα έχουν δύο κρυφά επίπεδα, στα οποία τους αριθμούς των κρυφών κόμβων τους θεωρούμε ως υπερ-παραμέτρους. Το κάθε κρυφό επίπεδο ακολουθείται από μία συνάρτηση ReLU και από Dropout, όπου οι πιθανότητες του κάθε επιπέδου Dropout θεωρούνται ως υπερ-παράμετροι. Επιπλέον το κάθε MLP μοντέλο έχει ένα επίπεδο εξόδου, με ένα κόμβο εξόδου, που ακολουθείται από μία σιγμοειδή συνάρτηση, καθώς η ταξινόμηση που κάνουμε είναι δυαδική. Όπως αναφέρουμε και στη θεωρία, εάν η έξοδος της σιγμοειδούς συνάρτησης είναι στο διάστημα (0,0.5), τότε το μοντέλο προβλέπει την κλάση με ετικέτα 0, ενώ αν η έξοδος είναι στο διάστημα (0.5,1), τότε το μοντέλο προβλέπει την κλάση με ετικέτα 1. Στη δική μας περίπτωση, η κλάση με ετικέτα 1 αντιστοιχεί στα tweets που είναι σχετικά με την αντίστοιχη καταστροφή. Ως υπερ-παραμέτρους θεωρούμε επίσης τον αλγόριθμο βελτιστοποίησης, τον ρυθμό μάθησης (learning rate) και το μέγεθος του batch. Ως συνάρτηση κόστους χρησιμοποιούμε τη BCE. Στον παρακάτω πίνακα βλέπουμε τα εύρη τιμών που θέσαμε στην κάθε υπερ-παράμετρο του κάθε μοντέλου:

Μοντέλο	Υπερ-παράμετροι	Τιμές
MLP Mean	Αριθμός κόμβων πρώτου κρυφού επιπέδου	30-45
	Αριθμός κόμβων δεύτερου κρυφού επιπέδου	10-29
	Πιθανότητα πρώτου επιπέδου Dropout	0.2-0.5
	Πιθανότητα δεύτερου επιπέδου Dropout	0.2-0.6
	Αλγόριθμος βελτιστοποίησης	Adam, Adagrad, RMSprop
	Ρυθμός μάθησης	0.0005-0.05
	Μέγεθος του batch	32, 64, 128
MLP MMM	Αριθμός κόμβων πρώτου κρυφού επιπέδου	80-130
	Αριθμός κόμβων δεύτερου κρυφού επιπέδου	20-79
	Πιθανότητα πρώτου επιπέδου Dropout	0.2-0.5
	Πιθανότητα δεύτερου επιπέδου Dropout	0.2-0.6
	Αλγόριθμος βελτιστοποίησης	Adam, Adagrad, RMSprop
	Ρυθμός μάθησης	0.0005-0.05
	Μέγεθος του batch	32, 64, 128

Πίνακας 5: Τα εύρη τιμών για την κάθε υπερ-παράμετρο για τα μοντέλα MLP Mean και MLP MMM.

Το εύρος τιμών των αριθμών των κρυφών κόμβων ορίστηκαν έτσι ώστε να καλυφθεί σχεδόν όλο το εύρος των πιθανών τιμών στο κάθε κρυφό επίπεδο διατηρώντας παράλληλα μία καθοδική πορεία στον αριθμό των κόμβων στο κάθε επίπεδο. Το εύρος των τιμών στις πιθανότητες του Dropout ορίστηκε με βάση τη βιβλιογραφία [24][28][29]. Οι αλγόριθμοι βελτιστοποίησης ορίστηκαν επίσης με βάση τη βιβλιογραφία. Στη βιβλιογραφία συνήθως επιλεγόταν ο αλγόριθμος Adam [20][23][26][27][52][53] και ο

αλγόριθμος ADADELTA [21][24][28][29][30], ενώ σε κάποιες έρευνες χρησιμοποιήθηκαν οι αλγόριθμοι RMSprop [52] και Adagrad [29][54]. Ο λόγος που δεν επιλέξαμε τον αλγόριθμο ADADELTA είναι επειδή η προτεινόμενη τιμή του ρυθμού μάθησης για αυτόν τον αλγόριθμο, 1, απέχει πολύ από τις προτεινόμενες τιμές για τους υπόλοιπους αλγορίθμους. Το εύρος για την τιμή του ρυθμού μάθησης ορίστηκε με βάση τις προτεινόμενες τιμές για τους αλγορίθμους Adam, Adagrad και RMSprop που είναι αντίστοιχα 0.001, 0.01 και 0.01. Επιπλέον, ως συνάρτηση κόστους επιλέχθηκε η BCE με βάση τη βιβλιογραφία [20][21][23][27][28][29][52][54]. Όσον αφορά το μέγεθος του batch, για τα μεγαλύτερα σύνολα δεδομένων, δηλαδή για τα σύνολα δεδομένων για τις καταιγίδες, τις πλημμύρες και τους σεισμούς, ως πιθανές τιμές ορίστηκαν το 64 και το 128, ενώ για τα μικρότερα σύνολα δεδομένων, δηλαδή για τα σύνολα δεδομένων για τις βιομηχανικές και τις κοινωνικές καταστροφές, ως πιθανές τιμές ορίστηκαν το 32 και το 64. Στον επόμενο πίνακα βλέπουμε τα αποτελέσματα της βελτιστοποίησης για την κάθε υπερ-παράμετρο, του κάθε μοντέλου για το κάθε σύνολο δεδομένων:

Σύνολο Δεδομένων	Μοντέλο	Υπερ-παράμετροι	Τιμές
Καταιγίδες	MLP Mean	Αριθμός κόμβων πρώτου κρυφού επιπέδου	42
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	28
		Πιθανότητα πρώτου επιπέδου Dropout	0.22407
		Πιθανότητα δεύτερου επιπέδου Dropout	0.47841
		Αλγόριθμος βελτιστοποίησης	RMSprop
		Ρυθμός μάθησης	0.00354
		Μέγεθος του batch	128
	MLP MMM	Αριθμός κόμβων πρώτου κρυφού επιπέδου	97
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	48
		Πιθανότητα πρώτου επιπέδου Dropout	0.29101
		Πιθανότητα δεύτερου επιπέδου Dropout	0.56695
		Αλγόριθμος βελτιστοποίησης	RMSprop
		Ρυθμός μάθησης	0.00184
		Μέγεθος του batch	128
Πλημμύρες	MLP Mean	Αριθμός κόμβων πρώτου κρυφού επιπέδου	41
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	25
		Πιθανότητα πρώτου επιπέδου Dropout	0.24522
		Πιθανότητα δεύτερου επιπέδου Dropout	0.34628
		Αλγόριθμος βελτιστοποίησης	Adam
		Ρυθμός μάθησης	0.01004
		Μέγεθος του batch	128
	MLP MMM	Αριθμός κόμβων πρώτου κρυφού επιπέδου	89
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	32
		Πιθανότητα πρώτου επιπέδου Dropout	0.22359
		Πιθανότητα δεύτερου επιπέδου Dropout	0.49324
		Αλγόριθμος βελτιστοποίησης	RMSprop
		Ρυθμός μάθησης	0.00069
		Μέγεθος του batch	64

Σεισμοί	MLP Mean	Αριθμός κόμβων πρώτου κρυφού επιπέδου	35
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	23
		Πιθανότητα πρώτου επιπέδου Dropout	0.23413
		Πιθανότητα δεύτερου επιπέδου Dropout	0.2847
		Αλγόριθμος βελτιστοποίησης	Adagrad
		Ρυθμός μάθησης	0.03992
		Μέγεθος του batch	64
	MLP MMM	Αριθμός κόμβων πρώτου κρυφού επιπέδου	107
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	32
		Πιθανότητα πρώτου επιπέδου Dropout	0.3998
		Πιθανότητα δεύτερου επιπέδου Dropout	0.27344
		Αλγόριθμος βελτιστοποίησης	RMSprop
		Ρυθμός μάθησης	0.00089
		Μέγεθος του batch	128
Βιομ. Καταστροφές	MLP Mean	Αριθμός κόμβων πρώτου κρυφού επιπέδου	33
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	27
		Πιθανότητα πρώτου επιπέδου Dropout	0.37121
		Πιθανότητα δεύτερου επιπέδου Dropout	0.28392
		Αλγόριθμος βελτιστοποίησης	RMSprop
		Ρυθμός μάθησης	0.00977
		Μέγεθος του batch	64
	MLP MMM	Αριθμός κόμβων πρώτου κρυφού επιπέδου	104
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	36
		Πιθανότητα πρώτου επιπέδου Dropout	0.43195
		Πιθανότητα δεύτερου επιπέδου Dropout	0.30564
		Αλγόριθμος βελτιστοποίησης	Adam
		Ρυθμός μάθησης	0.00319
		Μέγεθος του batch	64
Κοιν. Καταστροφές	MLP Mean	Αριθμός κόμβων πρώτου κρυφού επιπέδου	33
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	19
		Πιθανότητα πρώτου επιπέδου Dropout	0.39363
		Πιθανότητα δεύτερου επιπέδου Dropout	0.2291
		Αλγόριθμος βελτιστοποίησης	Adam
		Ρυθμός μάθησης	0.00223
		Μέγεθος του batch	64
	MLP MMM	Αριθμός κόμβων πρώτου κρυφού επιπέδου	128
		Αριθμός κόμβων δεύτερου κρυφού επιπέδου	53
		Πιθανότητα πρώτου επιπέδου Dropout	0.25428
		Πιθανότητα δεύτερου επιπέδου Dropout	0.39416
		Αλγόριθμος βελτιστοποίησης	Adam
		Ρυθμός μάθησης	0.00138
		Μέγεθος του batch	64

Πίνακας 6: Τα αποτελέσματα της βελτιστοποίησης για την κάθε υπερ-παράμετρο για το κάθε σύνολο δεδομένων για τα μοντέλα MLP Mean και MLP MMM.

4.2.2 Μοντέλα Βασισμένα στο LSTM

Δημιουργούμε τέσσερα μοντέλα βασισμένα στο LSTM. Ένα απλό μοντέλο LSTM, ένα LSTM μοντέλο με μηχανισμό προσοχής, ένα BiLSTM μοντέλο και ένα BiLSTM μοντέλο με μηχανισμό προσοχής. Σε όλα τα μοντέλα αρχικά προηγείται το επίπεδο με τα word embeddings. Το LSTM μοντέλο έχει μέγεθος εισόδου 50, ίσο με τη διάσταση των word embeddings, και μέγεθος εξόδου ίσο με τον αριθμό των κρυφών κόμβων του μοντέλου, που αποτελεί μία υπερ-παράμετρος που ρυθμίζεται μέσω της βελτιστοποίησης. Το LSTM έχει ένα επίπεδο, ενώ ακολουθείται από ένα επίπεδο Dropout, του οποίου η πιθανότητα είναι επίσης μία υπερ-παράμετρος. Στη συνέχεια, ακολουθείται από το επίπεδο εξόδου, το οποίο έχει έναν κόμβο εξόδου, και από μία σιγμοειδή συνάρτηση. Η διαφορά του μοντέλου LSTM με μηχανισμό προσοχής από το απλό μοντέλο LSTM είναι ότι παρεμβάλεται ο μηχανισμός προσοχής ανάμεσα στις εξόδους του LSTM και στο επίπεδο Dropout. Το μοντέλο αυτό έχει επίσης ένα επίπεδο στο LSTM και τις ίδιες υπερ-παραμέτρους. Ο μηχανισμός προσοχής που χρησιμοποιούμε είναι ο Ιεραρχικός Μηχανισμός Προσοχής, τον οποίο αναφέραμε στην υποενότητα 3.8.5. Το BiLSTM μοντέλο διαφέρει από το μοντέλο LSTM μόνο στο γεγονός ότι είναι αμφίδρομο (Bidirectional), ενώ όπως και τα υπόλοιπα μοντέλα το BiLSTM έχει ένα επίπεδο και τις ίδιες υπερ-παραμέτρους. Επιπλέον, εφόσον το μοντέλο αυτό αποτελείται από δύο LSTMs, η είσοδος στο επίπεδο εξόδου έχει μέγεθος διπλάσιο από τον αριθμό των κρυφών κόμβων των LSTMs. Τέλος, το BiLSTM μοντέλο με μηχανισμό προσοχής διαφέρει από τα υπόλοιπα μοντέλα στο ότι εφαρμόζεται μηχανισμός προσοχής δύο φορές, μία φορά για την έξοδο του LSTM που επεξεργάζεται τα δεδομένα με την κανονική τους κατεύθυνση και μία φορά για την έξοδο του LSTM που επεξεργάζεται τα δεδομένα από την ανάποδη κατεύθυνση. Ο μηχανισμός προσοχής που εφαρμόζεται και σε αυτήν την περίπτωση είναι ο Ιεραρχικός Μηχανισμός Προσοχής. Ως υπερ-παραμέτρους θεωρούμε επίσης τον ρυθμό μάθησης και το μέγεθος του batch, ενώ ως αλγόριθμο βελτιστοποίησης χρησιμοποιούμε τον Adam και ως συνάρτηση κόστους χρησιμοποιούμε τη BCE. Σε όλα τα μοντέλα θέσαμε τα ίδια εύρη τιμών για όλες τις υπερ-παραμέτρους, τα οποία φαίνονται στον ακόλουθο πίνακα:

Υπερ-παράμετροι	Τιμές
Αριθμός κρυφών κόμβων	30-100, ανά 10
Πιθανότητα επιπέδου Dropout	0.35-0.55
Ρυθμός μάθησης	0.0007-0.003
Μέγεθος του batch	32, 64, 128

Πίνακας 7: Τα εύρη τιμών για την κάθε υπερ-παράμετρο για τα μοντέλα LSTM, LSTM Attention, BiLSTM και BiLSTM Attention.

Το εύρος τιμών των αριθμών των κρυφών κόμβων ορίστηκε έτσι ώστε να καλυφθεί ένα εύρος τιμών γύρω από το μέγεθος της εισόδου. Επίσης, σε αυτά τα μοντέλα επιλέξαμε τον αλγόριθμο βελτιστοποίησης Adam καθώς με τον RMSprop η υπερπροσαρμογή (overfitting) γινόταν μέσα σε μόλις 2-3 εποχές σε όλα τα μοντέλα και σε όλα τα σύνολα δεδομένων, ενώ το ίδιο συνέβαινε εάν αυξάναμε τον αριθμό των επιπέδων στα LSTMs. Ο αλγόριθμος βελτιστοποίησης Adagrad από την άλλη έδινε ελαφρώς χειρότερα αποτελέσματα, ενώ η εκπαίδευση διαρκούσε πολλές εποχές, πάνω από 50 στις περισσότερες περιπτώσεις. Οπότε επιλέξαμε τον Adam που δίνει τα καλύτερα αποτελέσματα, ενώ η υπερπροσαρμογή στις περισσότερες περιπτώσεις καθυστερεί για 5 με 10 εποχές, και συνεπώς το μοντέλο προλαβαίνει να εκπαιδευτεί επαρκώς. Επιπλέον, για τον παραπάνω λόγο αυξήσαμε και το διάστημα της πιθανότητας του επιπέδου Dropout. Το εύρος για την τιμή του ρυθμού μάθησης ορίστηκε με βάση την προτεινόμενη τιμή για τον αλγόριθμο Adam, 0.001, ενώ η BCE όπως και στην περίπτωση των MLPs μοντέλων επιλέχθηκε με βάση τη βιβλιογραφία. Ως πιθανές τιμές του μεγέθους του batch, επίσης όπως και στα μοντέλα της προηγούμενης ενότητας, ορίστηκαν το 64 και το 128

για τα μεγαλύτερα σύνολα δεδομένων, ενώ το 32 και το 64 για τα δύο μικρότερα σύνολα δεδομένων. Τα αποτελέσματα της βελτιστοποίησης για την κάθε υπερ-παράμετρο, του κάθε μοντέλου για το κάθε σύνολο δεδομένων παρουσιάζονται στον παρακάτω πίνακα:

Σύνολο Δεδομένων	Μοντέλο	Υπερ-παράμετροι	Τιμές
Καταιγίδες	LSTM	Αριθμός κρυφών κόμβων	60
		Πιθανότητα επιπέδου Dropout	0.39446
		Ρυθμός μάθησης	0.00298
		Μέγεθος του batch	64
	LSTM Attention	Αριθμός κρυφών κόμβων	40
		Πιθανότητα επιπέδου Dropout	0.351
		Ρυθμός μάθησης	0.00283
		Μέγεθος του batch	128
	BiLSTM	Αριθμός κρυφών κόμβων	30
		Πιθανότητα επιπέδου Dropout	0.35409
		Ρυθμός μάθησης	0.0023
		Μέγεθος του batch	64
	BiLSTM Attention	Αριθμός κρυφών κόμβων	80
		Πιθανότητα επιπέδου Dropout	0.42056
		Ρυθμός μάθησης	0.0028
		Μέγεθος του batch	128
Πλημμύρες	LSTM	Αριθμός κρυφών κόμβων	40
		Πιθανότητα επιπέδου Dropout	0.40126
		Ρυθμός μάθησης	0.00261
		Μέγεθος του batch	64
	LSTM Attention	Αριθμός κρυφών κόμβων	50
		Πιθανότητα επιπέδου Dropout	0.35128
		Ρυθμός μάθησης	0.00205
		Μέγεθος του batch	128
	BiLSTM	Αριθμός κρυφών κόμβων	70
		Πιθανότητα επιπέδου Dropout	0.36147
		Ρυθμός μάθησης	0.00241
		Μέγεθος του batch	64
	BiLSTM Attention	Αριθμός κρυφών κόμβων	80
		Πιθανότητα επιπέδου Dropout	0.53875
		Ρυθμός μάθησης	0.00177
		Μέγεθος του batch	64
Σεισμοί	LSTM	Αριθμός κρυφών κόμβων	100
		Πιθανότητα επιπέδου Dropout	0.37285
		Ρυθμός μάθησης	0.00297
		Μέγεθος του batch	64
	LSTM Attention	Αριθμός κρυφών κόμβων	60
		Πιθανότητα επιπέδου Dropout	0.38713
		Ρυθμός μάθησης	0.00276
		Μέγεθος του batch	64

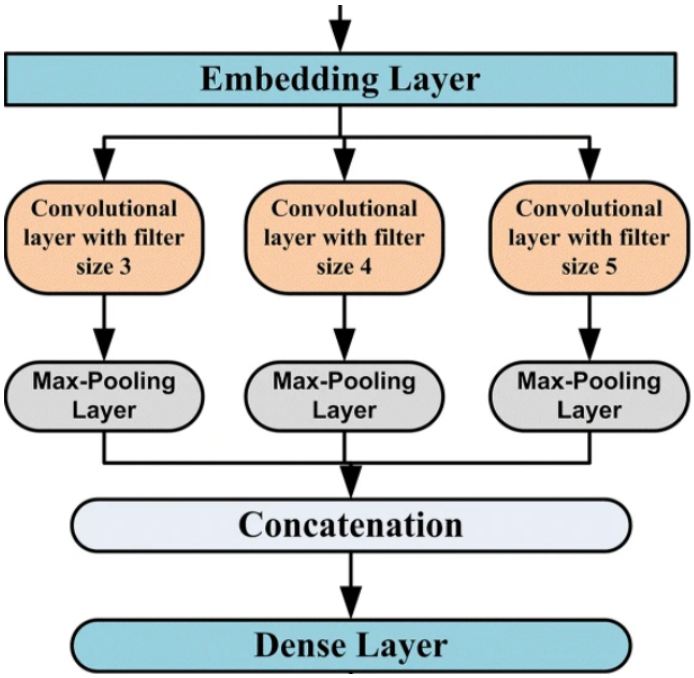
Σεισμοί	BiLSTM	Αριθμός κρυφών κόμβων	100
		Πιθανότητα επιπέδου Dropout	0.35153
		Ρυθμός μάθησης	0.0025
		Μέγεθος του batch	64
	BiLSTM Attention	Αριθμός κρυφών κόμβων	100
		Πιθανότητα επιπέδου Dropout	0.53843
		Ρυθμός μάθησης	0.00298
		Μέγεθος του batch	64
Βιομ. Καταστροφές	LSTM	Αριθμός κρυφών κόμβων	40
		Πιθανότητα επιπέδου Dropout	0.42246
		Ρυθμός μάθησης	0.00086
		Μέγεθος του batch	32
	LSTM Attention	Αριθμός κρυφών κόμβων	60
		Πιθανότητα επιπέδου Dropout	0.4808
		Ρυθμός μάθησης	0.00286
		Μέγεθος του batch	32
	BiLSTM	Αριθμός κρυφών κόμβων	40
		Πιθανότητα επιπέδου Dropout	0.46169
		Ρυθμός μάθησης	0.00274
		Μέγεθος του batch	32
	BiLSTM Attention	Αριθμός κρυφών κόμβων	100
		Πιθανότητα επιπέδου Dropout	0.39434
		Ρυθμός μάθησης	0.00227
		Μέγεθος του batch	32
Κοιν. Καταστροφές	LSTM	Αριθμός κρυφών κόμβων	50
		Πιθανότητα επιπέδου Dropout	0.38923
		Ρυθμός μάθησης	0.00261
		Μέγεθος του batch	32
	LSTM Attention	Αριθμός κρυφών κόμβων	50
		Πιθανότητα επιπέδου Dropout	0.42111
		Ρυθμός μάθησης	0.0026
		Μέγεθος του batch	32
	BiLSTM	Αριθμός κρυφών κόμβων	60
		Πιθανότητα επιπέδου Dropout	0.47177
		Ρυθμός μάθησης	0.00133
		Μέγεθος του batch	64
	BiLSTM Attention	Αριθμός κρυφών κόμβων	70
		Πιθανότητα επιπέδου Dropout	0.54917
		Ρυθμός μάθησης	0.00126
		Μέγεθος του batch	32

Πίνακας 8: Τα αποτελέσματα της βελτιστοποίησης για την κάθε υπερ-παράμετρο για το κάθε σύνολο δεδομένων για τα μοντέλα LSTM, LSTM Attention, BiLSTM και BiLSTM Attention.

4.2.3 Μοντέλο CNN

Δημιουργούμε ένα CNN μοντέλο βασισμένο στη βιβλιογραφία [23][24][25][28][29]. Το μοντέλο αυτό αρχικά αποτελείται από το επίπεδο με τα word embeddings. Έπειτα, ακολουθείται από τρία

CNNs τα οποία λειτουργούν παράλληλα. Όπως έχουμε πει στη θεωρία, τα CNNs στα προβλήματα Επεξεργασίας Φυσικής Γλώσσας λειτουργούν σαν n-grams. Το καθένα λοιπόν από τα επιμέρους CNNs έχει διαφορετικό μέγεθος πυρήνα έτσι ώστε να καλύπτονται διαφορετικά n-grams. Επιλέξαμε τα μεγέθη 2, 3 και 4, καθώς το μέγεθος των προτάσεων είναι αρκετά μικρό. Συνολικά, το καθένα από τα παράλληλα CNNs αποτελείται από ένα συνελικτικό επίπεδο, όπου η είσοδος έχει διάσταση 50, όσο δηλαδή είναι το μέγεθος των word embeddings, μία συνάρτηση ReLU και ένα επίπεδο μέγιστης ομαδοποίησης (max pooling). Έπειτα, η έξοδος από τα τρία αυτά CNNs συνενώνεται και περνάει από ένα επίπεδο Dropout. Σημειώνουμε ότι η έξοδος του κάθε συνελικτικού επιπέδου έχει μέγεθος $batch\_size \times \#filters \times (12 - kernel\_size + 1)$, ενώ η έξοδος του κάθε επιπέδου μέγιστης ομαδοποίησης έχει μέγεθος $batch\_size \times \#filters$. Ο αριθμός των φίλτρων, που είναι ο ίδιος σε όλα τα επιμέρους CNNs, καθώς και η πιθανότητα του επιπέδου Dropout αποτελούν τις υπερ-παραμέτρους του μοντέλου. Στο τέλος έχουμε το επίπεδο εξόδου, το οποίο ως είσοδο έχει τρεις φορές τον αριθμό των φίλτρων και ως έξοδο έναν κόμβο, ενώ ακολουθείται από τη σιγμοειδή συνάρτηση. Στην επόμενη εικόνα βλέπουμε ένα παρόμοιο μοντέλο από τη βιβλιογραφία:



Εικόνα 18: Ένα παράδειγμα μοντέλου που αποτελείται από τρία παράλληλα CNNs με διαφορετικό μέγεθος πυρήνα το καθένα [24].

Επίσης, ως υπερ-παραμέτρους θεωρούμε τον ρυθμό μάθησης και το μέγεθος του batch, ενώ ως αλγόριθμο βελτιστοποίησης χρησιμοποιούμε τον Adam και ως συνάρτηση κόστους χρησιμοποιούμε την BCE. Το εύρος των τιμών για όλες τις υπερ-παραμέτρους φαίνεται στον παρακάτω πίνακα:

Υπερ-παράμετροι	Τιμές
Αριθμός φίλτρων	50-200, ανά 50
Πιθανότητα επιπέδου Dropout	0.45-0.55
Ρυθμός μάθησης	0.0007-0.003
Μέγεθος του batch	32, 64, 128

Πίνακας 9: Τα εύρη τιμών για την κάθε υπερ-παράμετρο για το μοντέλο CNN.

Το εύρος των τιμών για τον αριθμό των φίλτρων ορίστηκε με βάση τη βιβλιογραφία [22][23][24][25] [28][29]. Επίσης, επιλέξαμε τον αλγόριθμο Adam για τον ίδιο λόγο που τον επιλέξαμε και για τα μοντέλα LSTMs, ενώ αυξήσαμε πολύ το διάστημα της πιθανότητας του επιπέδου Dropout. Όπως και στην προηγούμενη υποενότητα, το εύρος για την τιμή του ρυθμού μάθησης ορίστηκε με βάση την προτεινόμενη τιμή για τον αλγόριθμο Adam, 0.001, ενώ η BCE επιλέχθηκε με βάση τη βιβλιογραφία. Τέλος, ως πιθανές τιμές του μεγέθους του batch ορίστηκαν το 64 και το 128 για τα τρία μεγαλύτερα σύνολα δεδομένων και το 32 και το 64 για τα δύο μικρότερα σύνολα δεδομένων. Στον ακόλουθο πίνακα φαίνονται τα αποτελέσματα της βελτιστοποίησης για την κάθε υπερ-παράμετρο του κάθε μοντέλου:

Σύνολο Δεδομένων	Υπερ-παράμετροι	Τιμές
Καταιγίδες	Αριθμός φίλτρων	200
	Πιθανότητα επιπέδου Dropout	0.47288
	Ρυθμός μάθησης	0.00123
	Μέγεθος του batch	64
Πλημμύρες	Αριθμός φίλτρων	100
	Πιθανότητα επιπέδου Dropout	0.45323
	Ρυθμός μάθησης	0.00296
	Μέγεθος του batch	64
Σεισμοί	Αριθμός φίλτρων	200
	Πιθανότητα επιπέδου Dropout	0.46489
	Ρυθμός μάθησης	0.00075
	Μέγεθος του batch	64
Βιομ. Καταστροφές	Αριθμός φίλτρων	200
	Πιθανότητα επιπέδου Dropout	0.49465
	Ρυθμός μάθησης	0.00142
	Μέγεθος του batch	64
Κοιν. Καταστροφές	Αριθμός φίλτρων	150
	Πιθανότητα επιπέδου Dropout	0.49102
	Ρυθμός μάθησης	0.00087
	Μέγεθος του batch	64

Πίνακας 10: Τα αποτελέσματα της βελτιστοποίησης για την κάθε υπερ-παράμετρο για το κάθε σύνολο δεδομένων για το μοντέλο CNN.

4.3 Μεταφορά Μάθησης

Στην ενότητα αυτή αναφέρουμε τη δομή των προ-εκπαιδευμένων γλωσσικών μοντέλων που χρησιμοποιούμε για την πρόβλεψη των σχετικών tweets ανά καταστροφή. Σε αυτά τα μοντέλα δεν εκτελέσαμε βελτιστοποίηση υπερ-παραμέτρων καθώς ακόμα και το fine-tuning έχει διάρκεια πολλών ωρών. Οπότε θέσαμε τις αντίστοιχες τιμές στην κάθε υπερ-παράμετρο που προτείνουν οι συγγραφείς του κάθε μοντέλου.

4.3.1 BERT

Χρησιμοποιούμε το BERT base μοντέλο της βιβλιοθήκης transformers, το οποίο αποτελείται από 12 επίπεδα transformers, όπου το κάθε επίπεδο έχει 768 κρυφούς κόμβους, ενώ συνολικά έχει 110M παραμέτρους. Για την τεχνική fine-tuning προσθέτουμε ένα επίπεδο εξόδου, όπως προτείνουν οι συγγραφείς [43], με 768 κόμβους εισόδου και με 1 κόμβο εξόδου, καθώς και μία σιγμοειδή συνάρτηση. Στον παρακάτω πίνακα φαίνονται οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς:

Υπερ-παράμετροι	Τιμές
Αριθμός εποχών	2-4
Αλγόριθμος βελτιστοποίησης	Adam
Ρυθμός μάθησης	5e-5, 3e-5, 2e-5
Μέγεθος του batch	16, 32

Πίνακας 11: Οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς για το μοντέλο BERT [43].

Ο καθορισμός των εποχών έγινε με τον πρόωρο τερματισμό, όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 6 (4+2) και ο προκαθορισμένος αριθμός εποχών όπου δεν παρουσιάζει βελτίωση το μοντέλο στα δεδομένα επαλήθευσης (patience) τέθηκε ίσος με 2. Ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο AdamW από τη βιβλιοθήκη transformers και ως ρυθμός μάθησης τέθηκε ο αριθμός 3e-5. Τέλος, ως μέγεθος του batch τέθηκε σε όλα τα σύνολα δεδομένων το 32.

4.3.2 DistilBERT

Επιλέγουμε το DistilBERT base μοντέλο της βιβλιοθήκης transformers το οποίο αποτελείται από 6 επίπεδα transformers, όπου το κάθε επίπεδο έχει 768 κρυφούς κόμβους. Όσον αφορά την τεχνική fine-tuning, προσθέτουμε ένα επίπεδο εξόδου, με 768 κόμβους εισόδου και με 1 κόμβο εξόδου, καθώς και μία σιγμοειδή συνάρτηση. Οι τιμές που προτείνονται από τους συγγραφείς [45] για το fine-tuning είναι οι ίδιες που προτείνονται και για το BERT. Συνεπώς, ο καθορισμός των εποχών έγινε με τον πρόωρο τερματισμό, όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 6 και το patience τέθηκε ίσο με 2. Επιπλέον, ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο AdamW από τη βιβλιοθήκη transformers και ως ρυθμός μάθησης τέθηκε ο αριθμός 3e-5, ενώ το μέγεθος του batch τέθηκε σε όλα τα σύνολα δεδομένων ίσο με 32.

4.3.3 DeBERTa

Χρησιμοποιούμε το BeBERTa base της βιβλιοθήκης transformers, το οποίο όπως και το BERT αποτελείται από 12 επίπεδα transformers, όπου το καθένα έχει 768 κρυφούς κόμβους. Έπειτα, για την τεχνική fine-tuning προσθέτουμε ένα επίπεδο εξόδου, με 768 κόμβους εισόδου και με 1 κόμβο εξόδου, καθώς και μία σιγμοειδή συνάρτηση. Οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς [46] για το base μοντέλο φαίνονται στον επόμενο πίνακα:

Υπερ-παράμετροι	Τιμές
Μέγιστος αριθμός εποχών	10
Αλγόριθμος βελτιστοποίησης	Adam
Adam ϵ	1e-6
Ρυθμός μάθησης	1.5e-5, 2e-5, 3e-5, 4e-5
Μέγεθος του batch	16, 32, 48, 64

Πίνακας 12: Οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς για το μοντέλο DeBERTa [46].

Όπως και στα προηγούμενα μοντέλα ο καθορισμός των εποχών έγινε με τον πρόωρο τερματισμό, όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 10 και το patience τέθηκε ίσο με 2. Ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο AdamW από τη βιβλιοθήκη transformers και ως ρυθμός μάθησης

τέθηκε ο αριθμός 2e-5, ενώ το Adam ϵ τέθηκε ίσο με 1e-6. Τέλος, ως μέγεθος του batch τέθηκε σε όλα τα σύνολα δεδομένων το 32.

4.3.4 XLNet

Επιλέγουμε το XLNet base της βιβλιοθήκης transformers, το οποίο αποτελείται επίσης από 12 επίπεδα transformers, όπου το κάθε επίπεδο έχει 768 κρυφούς κόμβους. Για την τεχνική fine-tuning προσθέτουμε ένα επίπεδο εξόδου, με 768 κόμβους εισόδου και με 1 κόμβο εξόδου, καθώς και μία σιγμοειδή συνάρτηση. Στον ακόλουθο πίνακα βλέπουμε τις τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς [48] για το base μοντέλο:

Υπερ-παράμετροι	Τιμές
Αριθμός εποχών	2-4
Αλγόριθμος βελτιστοποίησης	Adam
Adam ϵ	1e-6
Ρυθμός μάθησης	2e-5
Φθορά βαρών	0.01
Μέγεθος του batch	32-128

Πίνακας 13: Οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς για το μοντέλο XLNet [48].

Ο καθορισμός των εποχών έγινε όπως και στα προηγούμενα μοντέλα με τον πρόωρο τερματισμό, όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 6 και το patience τέθηκε ίσο με 2. Επίσης, ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο AdamW της βιβλιοθήκης transformers και ως ρυθμός μάθησης τέθηκε ο αριθμός 2e-5, ενώ το Adam ϵ και η φθορά βαρών (weight decay) τέθηκαν ίσα με 1e-6 και 0.01 αντιστοίχως. Επιπλέον, το μέγεθος του batch τέθηκε σε όλα τα σύνολα δεδομένων ίσο με 32.

4.3.5 ELECTRA

Χρησιμοποιούμε το ELECTRA small της βιβλιοθήκης transformers, το οποίο περιλαμβάνει 12 επίπεδα transformers, όπου το καθένα έχει 256 κρυφούς κόμβους. Ο λόγος που επιλέγουμε το small αντί του base είναι ότι τα σύνολα δεδομένων μας είναι αρκετά μικρά για το base μοντέλο, με αποτέλεσμα να μην μπορεί να βγάλει σωστά αποτελέσματα. Στη συνέχεια, για την τεχνική fine-tuning προσθέτουμε ένα επίπεδο εξόδου, με 256 κόμβους εισόδου και με 1 κόμβο εξόδου, καθώς και μία σιγμοειδή συνάρτηση. Τις τιμές για τις υπερ-παραμέτρους που προτείνουν οι συγγραφείς [49] για το small μοντέλο τις βλέπουμε στον παρακάτω πίνακα:

Υπερ-παράμετροι	Τιμές
Μέγιστος αριθμός εποχών	10
Αλγόριθμος βελτιστοποίησης	Adam
Adam ϵ	1e-6
Ρυθμός μάθησης	1e-4
Φθορά βαρών	0
Μέγεθος του batch	32

Πίνακας 14: Οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς για το μοντέλο ELECTRA [49].

Όπως σε όλα τα προηγούμενα μοντέλα, ο καθορισμός των εποχών έγινε με τον πρόωρο τερματισμό, όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 10 και το patience τέθηκε ίσο με 2. Σημειώνουμε ότι το κριτήριο σε όλα τα μοντέλα με το οποίο ελέγχει ο πρόωρος τερματισμός για το ενδεχόμενο βελτίωσης είναι το σφάλμα (loss) στα δεδομένα επαλήθευσης. Ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο AdamW της βιβλιοθήκης transformers και ως ρυθμός μάθησης τέθηκε ο αριθμός 1e-4, ενώ το Adam ϵ και η φθορά βαρών τέθηκαν ίσα με 1e-6 και 0 αντιστοίχως. Τέλος, ως μέγεθος του batch τέθηκε σε όλα τα σύνολα δεδομένων το 32.

5 Πειράματα και Αποτελέσματα

Σε αυτό το κεφάλαιο παρουσιάζουμε τα αποτελέσματα από όλα τα πειράματα που εκτελέσαμε. Για όλα τα πειράματα χρησιμοποιήθηκε η βιβλιοθήκη PyTorch [8], ενώ όλα πραγματοποιήθηκαν σε επεξεργαστή Intel(R) Core(TM) i7-4790 3.6GHz και σε RAM 8GB. Αρχικά, παρουσιάζουμε τα αποτελέσματα από τα νευρωνικά δίκτυα, στη συνέχεια τα αποτελέσματα από τα πειράματα στην κατηγορία μεταφορά μάθησης και τελικά συγκρίνουμε τα συνολικά αποτελέσματα.

5.1 Νευρωνικά Δίκτυα

Σε αυτήν την ενότητα παρουσιάζουμε τα αποτελέσματα των πειραμάτων για τα νευρωνικά δίκτυα, στα οποία χρησιμοποιήθηκαν οι βέλτιστες τιμές για τις υπερ-παραμέτρους, που προέκυψαν από τη βελτιστοποίηση. Τα αποτελέσματα τα παρουσιάζουμε ανά σύνολο δεδομένων, ώστε να μπορέσουμε να κάνουμε σύγκριση. Σημειώνουμε επίσης ότι όλα τα αποτελέσματα προέρχονται από την καλύτερη εποχή με βάση τον πρόωρο τερματισμό (early stopping), ενώ αφορούν μόνο τα δεδομένα επαλήθευσης.

Στον παρακάτω πίνακα βλέπουμε τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τις καταιγίδες. Από τα αποτελέσματα παρατηρούμε αρχικά ότι όλες οι μετρικές έχουν παρόμοια τιμή. Αυτό σημαίνει ότι τα FP με τα FN έχουν πολύ κοντινές τιμές μεταξύ τους. Δηλαδή, τα μοντέλα μπερδεύουν τα αποτελέσματα μεταξύ των δύο κλάσεων το ίδιο και δεν δυσκολεύονται στη μία κλάση περισσότερο. Επιπλέον, βλέπουμε ότι όλα τα μοντέλα έχουν αρκετά ικανοποιητικά αποτελέσματα, καθώς όλα είναι πάνω από 90%. Τα MLPs μοντέλα έχουν το χειρότερο accuracy, με το μοντέλο MLP Mean να έχει συνολικά τα χειρότερα αποτελέσματα. Όπως ήταν λογικό, το MLP MMM έχει ελαφρώς καλύτερα αποτελέσματα, καθώς η είσοδος στο μοντέλο περιέχει περισσότερες πληροφορίες για την κάθε πρόταση. Επίσης, όπως ήταν αναμενόμενο, τα μοντέλα που είναι βασισμένα στο LSTM έχουν τα καλύτερα αποτελέσματα, ενώ το μοντέλο BiLSTM Attention έχει λίγο χαμηλότερο accuracy από τα υπόλοιπα. Ωστόσο, το accuracy του μοντέλου CNN βρίσκεται στη μέση μεταξύ των αντίστοιχων τιμών για τα MLPs και τα LSTMs. Τέλος, για το σύνολο δεδομένων με τις καταιγίδες τα καλύτερα αποτελέσματα τα έχει ευκρινώς το μοντέλο LSTM.

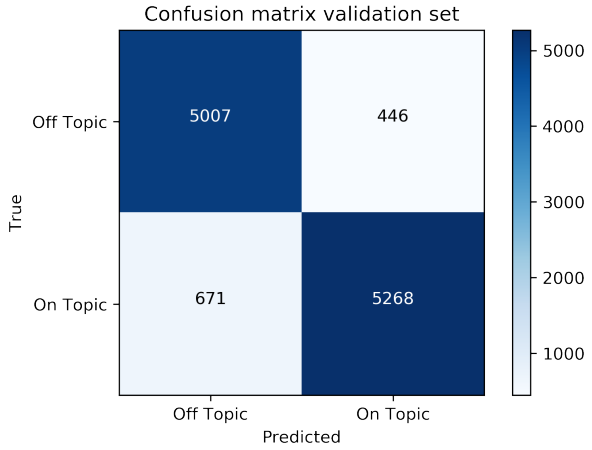
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
MLP Mean	90.20	90.3	90.2	90.2
MLP MMM	91.63	91.7	91.6	91.6
LSTM	**94.34**	**94.4**	**94.4**	**94.3**
LSTM Attention	94.18	94.2	94.2	94.2
BiLSTM	94.09	94.1	94.1	94.1
BiLSTM Attention	93.81	93.9	93.8	93.8
CNN	92.97	93.1	93.0	93.0

Πίνακας 15: Τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τις καταιγίδες.

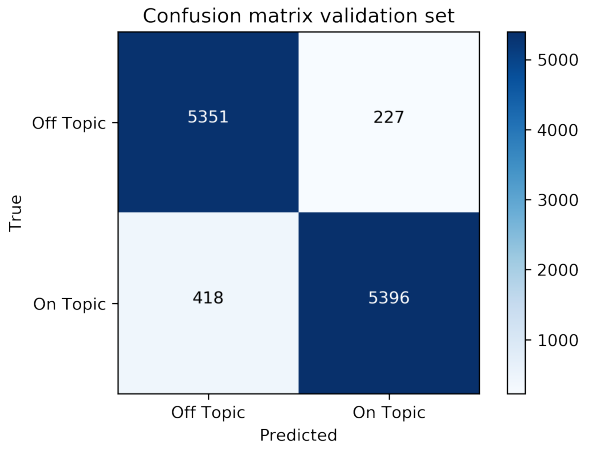
Ενδεικτικά παραθέτουμε τους πίνακες σύγχυσης (confusion matrices) από το χειρότερο και το καλύτερο μοντέλο, δηλαδή τα μοντέλα MLP Mean και LSTM. Από τις παρακάτω εικόνες επιβεβαιώνουμε αυτό που είχαμε εξηγήσει παραπάνω, ότι δηλαδή τα μοντέλα κάνουν παρόμοιο αριθμό λαθών στις δύο κλάσεις. Ωστόσο, βλέπουμε ότι τα μοντέλα κάνουν ελαφρώς περισσότερα λάθη στην πρόβλεψη των tweets που είναι σχετικά με την καταστροφή, δηλαδή τα κατατάσσουν ως μη σχετικά. Επιπλέον, όπως ήταν αναμενόμενο βλέπουμε ότι το MLP Mean συνολικά έχει περισσότερα σφάλματα από το LSTM.

[8] https://pytorch.org/

Σημειώνουμε ότι αυτές οι παρατηρήσεις επεκτείνονται και για τους πίνακες σύγχυσης των υπόλοιπων μοντέλων για το συγκεκριμένο σύνολο δεδομένων.



Εικόνα 19: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο MLP Mean για το σύνολο δεδομένων με τις καταιγίδες.



Εικόνα 20: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο LSTM για το σύνολο δεδομένων με τις καταιγίδες.

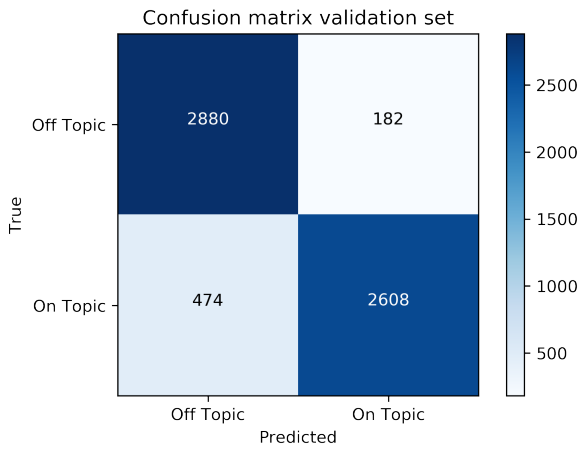
Τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τις πλημμύρες τα βλέπουμε στον επόμενο πίνακα. Αρχικά, όπως και πριν διαπιστώνουμε ότι όλες οι μετρικές έχουν παρόμοια τιμή και συνεπώς οι λάθος προβλέψεις είναι σχεδόν ίσα κατανεμημένες ανάμεσα στις δύο κλάσεις. Επίσης, παρατηρούμε ότι όλα τα αποτελέσματα είναι αρκετά καλά, γύρω στο 90%, ενώ οπώς και στο προηγούμενο σύνολο δεδομένων τα δύο MLPs μοντέλα έχουν τα χειρότερα αποτελέσματα, με το MLP Mean να έχει το χαμηλότερο accuracy. Τα υπόλοιπα μοντέλα έχουν αρκετά κοντινές τιμές μεταξύ τους, ενώ πάλι το CNN μοντέλο έχει χαμηλότερο accuracy από όλα τα μοντέλα που είναι βασιμένα στο LSTM. Τα καλύτερα αποτελέσματα για το σύνολο δεδομένων με τις πλημμύρες τα έχει το μοντέλο LSTM.

Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
MLP Mean	89.32	89.3	89.7	89.3
MLP MMM	90.00	90.1	90.0	90.0
LSTM	**91.72**	**91.7**	**91.8**	**91.7**
LSTM Attention	91.41	91.5	91.5	91.4
BiLSTM	91.47	91.5	91.6	91.5
BiLSTM Attention	91.59	91.6	91.7	91.6
CNN	91.36	91.4	91.5	91.4

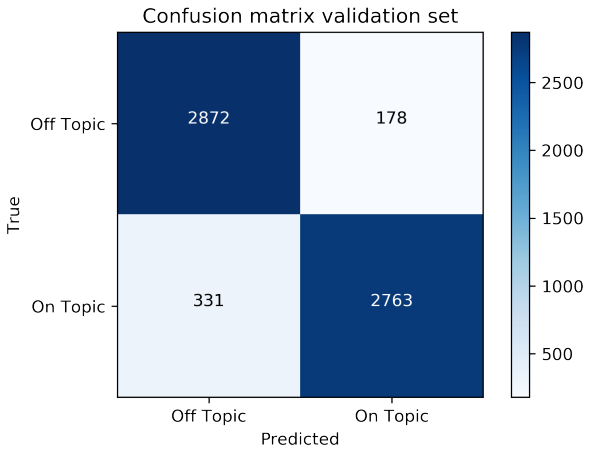
Πίνακας 16: Τα αποτελέσματα για το σύνολο δεδομένων με τις πλημμύρες όλων των νευρωνικών δικτύων.

Ενδεικτικά προσθέτουμε τους πίνακες σύγχυσης από τα μοντέλα MLP Mean και LSTM. Από τις παρακάτω εικόνες παρατηρούμε ότι παρόλο που το MLP Mean κάνει περισσότερα λάθη από το LSTM, σχεδόν όλα βρίσκονται στην πρόβλεψη των tweets που είναι σχετικά με την καταστροφή. Η διαπίστωση αυτή επεκτείνεται και στα υπόλοιπα μοντέλα, ότι δηλαδή ο αριθμός των λαθών στην πρόβλεψη των tweets που δεν είναι σχετικά με την καταστροφή παραμένει σχεδόν σταθερός σε όλα τα

μοντέλα. Άρα όλα τα μοντέλα παρουσιάζουν μία μεγαλύτερη δυσκολία σε αυτήν την κλάση, παρόλο που η διαφορά στα λάθη ανάμεσα στις δύο κλάσεις είναι αρκετά μικρή.



Εικόνα 21: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο MLP Mean για το σύνολο δεδομένων με τις πλημμύρες.



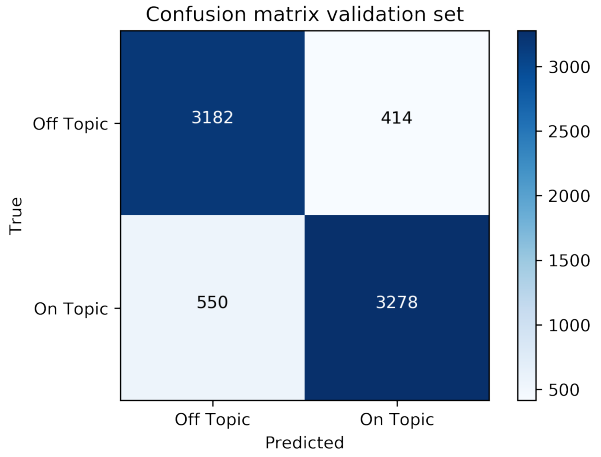
Εικόνα 22: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο LSTM για το σύνολο δεδομένων με τις πλημμύρες.

Στον ακόλουθο πίνακα παρουσιάζουμε τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τους σεισμούς. Όπως και στα δύο προηγούμενα σύνολα δεδομένων, οι μετρικές σε όλα τα μοντέλα έχουν πολύ κοντινή τιμή μεταξύ τους. Ωστόσο, βλέπουμε ότι τα αποτελέσματα σε αυτό το σύνολο δεδομένων είναι εμφανώς χαμηλότερα από τα αποτελέσματα των προηγουμένων συνόλων δεδομένων, γύρω στο 87%. Όπως ήταν αναμενόμενο το χειρότερο accuracy το έχει το μοντέλο MLP Mean, ενώ ακολουθεί το MLP MMM. Όμως η διαφορά στα αποτελέσματα μεταξύ των μοντέλων MLPs και LSTMs δεν είναι τόσο εμφανής όπως ήταν στην περίπτωση του συνόλου δεδομένων με τις καταιγίδες. Από τα LSTMs μοντέλα το χαμηλότερο accuracy το έχει το LSTM Attention μοντέλο, ενώ το υψηλότερο accuracy το έχει το μοντέλο BiLSTM. Τέλος, στο συγκεκριμένο σύνολο δεδομένων, το καλύτερο αποτέλεσμα το δίνει το μοντέλο CNN.

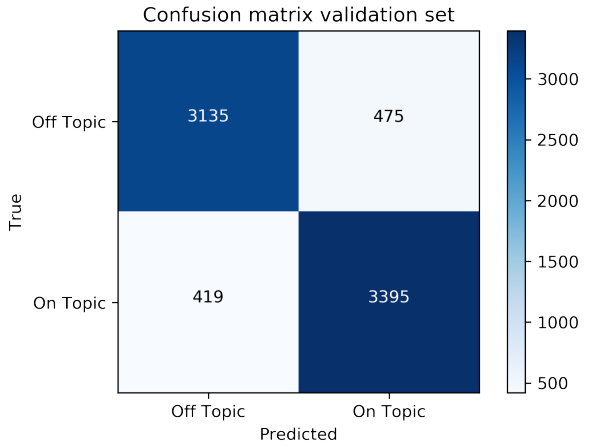
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
MLP Mean	85.25	85.3	85.3	85.3
MLP MMM	86.41	86.4	86.4	86.4
LSTM	87.46	87.5	87.5	87.5
LSTM Attention	87.02	87.1	87.0	87.0
BiLSTM	87.86	87.9	87.9	87.9
BiLSTM Attention	87.33	87.3	87.4	87.3
CNN	**87.96**	**87.9**	**88.0**	**87.9**

Πίνακας 17: Τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τους σεισμούς.

Από τους πίνακες σύγχυσης για αυτό το σύνολο δεδομένων, παρατηρούμε ότι τα λάθη των μοντέλων ανάμεσα στις δύο κλάσεις είναι όντως πολύ κοντά αριθμητικά, ενώ σε αντίθεση με τα δύο προηγούμενα σύνολα δεδομένων, τα περισσότερα λάθη δεν γίνονται πάντα στις προβλέψεις των tweets που είναι σχετικά με την καταστροφή. Παραθέτουμε ενδεικτικά τους πίνακες σύγχυσης από δύο μοντέλα, το LSTM Attention και το CNN, όπου είναι εμφανές αυτό που μόλις αναφέραμε.



Εικόνα 23: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο LSTM Attention για το σύνολο δεδομένων με τους σεισμούς.



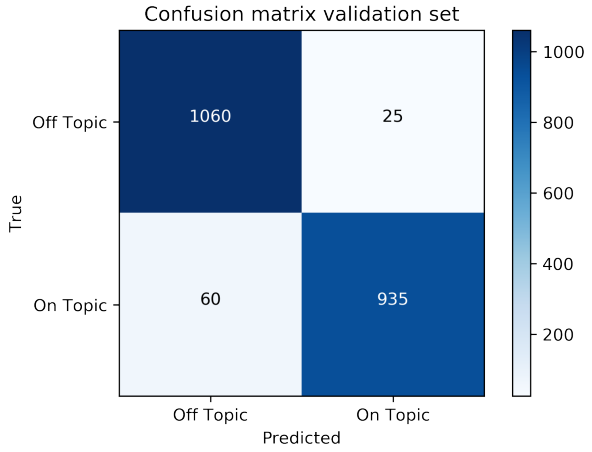
Εικόνα 24: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο CNN για το σύνολο δεδομένων με τους σεισμούς.

Τα αποτελέσματα για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές παρατίθενται στον παρακάτω πίνακα. Παρατηρούμε αρχικά ότι όλες οι μετρικές έχουν παρόμοιες τιμές μεταξύ τους, ενώ όλα τα μοντέλα έχουν εξαιρετικά αποτελέσματα, σχεδόν 96%. Σε αντίθεση με τα προηγούμενα σύνολα δεδομένων, τα μοντέλα MLPs έχουν παρόμοια αποτελέσματα με τα υπόλοιπα μοντέλα, ενώ το χειρότερο accuracy το έχει το μοντέλο MLP MMM. Το μοντέλο MLP έχει το αμέσως επόμενο χαμηλότερο accuracy αλλά η τιμή του είναι πολύ κοντά με τις αντίστοιχες τιμές των LSTMs και του CNN. Επιπλέον, όλα τα LSTMs δίνουν πρακτικά το ίδιο αποτέλεσμα, ενώ το υψηλότερο accuracy με μικρή διαφορά το πετυχαίνει το CNN μοντέλο.

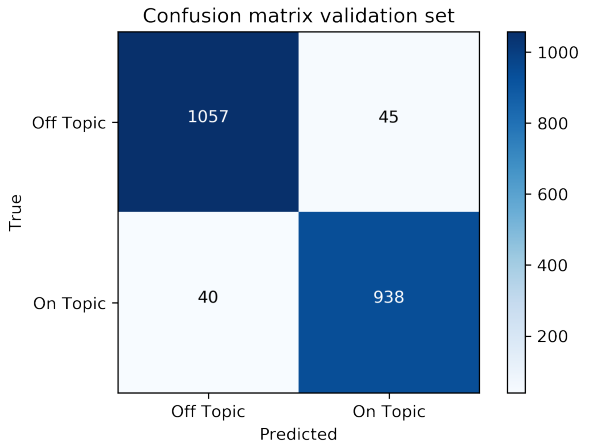
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
MLP Mean	95.51	95.5	95.5	95.5
MLP MMM	94.73	94.7	94.8	94.7
LSTM	95.91	95.9	95.9	95.9
LSTM Attention	95.87	95.9	95.9	95.9
BiLSTM	95.87	95.8	95.9	95.9
BiLSTM Attention	95.91	95.8	96.0	95.9
CNN	**96.05**	**96.0**	**96.1**	**96.0**

Πίνακας 18: Τα αποτελέσματα για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές όλων των νευρωνικών δικτύων.

Από τους πίνακες σύγχυσης για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές, βλέπουμε ότι σχεδόν όλα τα μοντέλα κάνουν λίγα παραπάνω λάθη στις προβλέψεις των tweets που είναι σχετικά με την καταστροφή. Συγκεκριμένα, στους δύο ακόλουθους πίνακες σύγχυσης, που είναι από τα μοντέλα BiLSTM Attention και LSTM, τα οποία έδωσαν το ίδιο accuracy, βλέπουμε ότι τα λάθη του BiLSTM Attention συγκεντρώνονται κυριώς στην κλάση όπου τα tweets είναι σχετικά με την καταστροφή, ενώ στην περίπτωση του LSTM τα λάθη είναι σχεδόν ίσα ανάμεσα στις δύο κλάσεις. Σημειώνουμε ότι σε αυτήν τη διαφορά στα λάθη στην περίπτωση του BiLSTM Attention οφείλεται η μικρή διαφορά στις μετρικές ανάκληση (recall) και ακρίβεια (precision) που βλέπουμε στον παραπάνω πίνακα.



Εικόνα 25: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο BiLSTM Attention για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές.



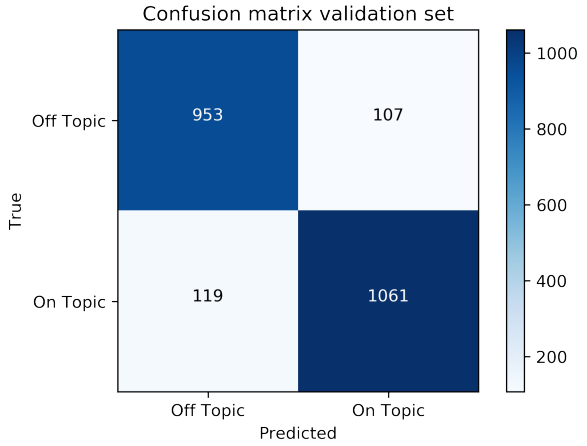
Εικόνα 26: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο LSTM για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές.

Τέλος, στον ακόλουθο πίνακα βλέπουμε τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τις κοινωνικές καταστροφές. Όπως και στα υπόλοιπα σύνολα δεδομένων παρατηρούμε ότι όλες οι μετρικές έχουν αρκετά κοντινές τιμές μεταξύ τους. Επιπλέον, όπως και στην περίπτωση του συνόλου δεδομένων με τις βιομηχανικές εποχές, όλα τα μοντέλα έχουν παρόμοια αποτελέσματα. Όπως ήταν αναμενόμενο, τα χειρότερα αποτελέσματα τα έχουν πάλι τα MLPs μοντέλα, ενώ το χαμηλότερο accuracy το έχει το μοντέλο MLP Mean. Το LSTM μοντέλο έχει εμφανώς τα καλύτερα αποτελέσματα, ενώ το αμέσως καλύτερο accuracy το δίνει το CNN. Τα υπόλοιπα LSTMs μοντέλα δίνουν παρόμοια αποτελέσματα.

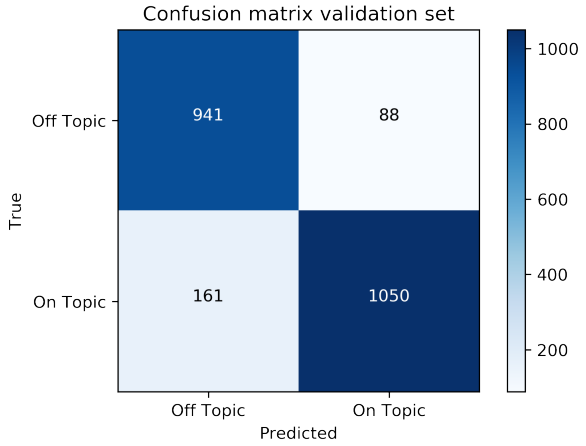
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
MLP Mean	87.86	87.9	87.8	87.8
MLP MMM	88.10	88.2	88.0	88.1
LSTM	**89.91**	**89.9**	**89.9**	**89.9**
LSTM Attention	88.80	88.9	88.7	88.8
BiLSTM	88.44	88.4	88.4	88.4
BiLSTM Attention	88.84	89.0	89.1	88.8
CNN	88.88	89.1	88.8	88.9

Πίνακας 19: Τα αποτελέσματα όλων των νευρωνικών δικτύων για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Παραθέτουμε ενδεικτικά τους πίνακες σύγχυσης των καλύτερων μοντέλων, δηλαδή των μοντέλων LSTM και CNN. Όπως διαπιστώνουμε από τις παρακάτω εικόνες, τα δύο μοντέλα κάνουν περισσότερα λάθη στην πρόβλεψη των tweets που είναι σχετικά με την καταστροφή, ενώ η διαφορά στα λάθη ανάμεσα στις δύο κλάσεις είναι μικρή. Το συμπέρασμα αυτό επεκτείνεται και στα υπόλοιπα μοντέλα.



Εικόνα 27: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο LSTM για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.



Εικόνα 28: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο CNN για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Συνολικά, όπως ήταν λογικό σε όλα τα σύνολα δεδομένων το χαμηλότερο accuracy το δίνει το μοντέλο MLP Mean, με εξαίρεση του συνόλου δεδομένων με τις βιομηχανικές καταστροφές όπου το χειρότερο αποτέλεσμα το έχει το MLP MMM μοντέλο. Αντιθέτως, τα καλύτερα αποτελέσματα τα δίνουν το LSTM ή το CNN, ενώ τα υπόλοιπα LSTMs μοντέλα έχουν αρκετά κοντινά αποτελέσματα μεταξύ τους σε όλα τα σύνολα δεδομένων. Συμπεραίνουμε λοιπόν ότι η αύξηση της πολυπλοκότητας του μοντέλου, όπως η πρόσθεση μηχανισμού προσοχής, δεν είναι πάντα απαραίτητη, ενώ ένα απλό LSTM μοντέλο αρκεί στις περισσότερες περιπτώσεις για να έχουμε πολύ ικανοποιητικά αποτελέσματα.

5.2 Μεταφορά Μάθησης

5.2.1 Fine-tuning στο Σύνολο Δεδομένων με τις Κοινωνικές Καταστροφές

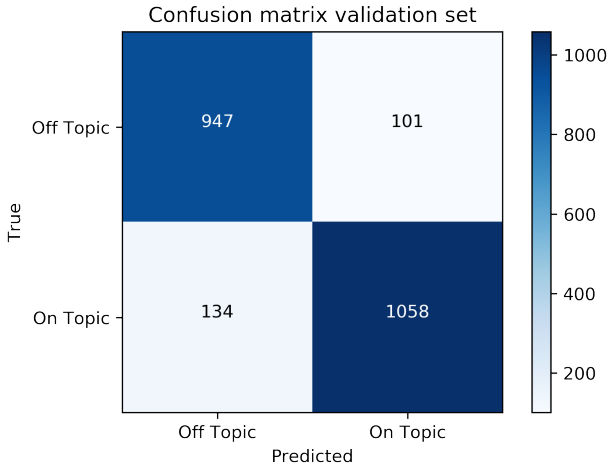
Σε αυτήν την υποενότητα δοκιμάζουμε την τεχνική fine-tuning με τη χρήση του καλύτερου μοντέλου του μεγαλύτερου συνόλου δεδομένων σε ένα μικρό σύνολο δεδομένων για να δούμε άμα θα υπάρξει βελτίωση των αντίστοιχων αποτελεσμάτων. Όπως έχουμε εξηγήσει στη θεωρία, η λογική είναι ότι το μοντέλο θα έχει ήδη γνώσεις πάνω σε μία καταστροφή, οπότε θα μπορεί να γενικεύσει και να μάθει καλύτερα από το μικρό σύνολο δεδομένων. Το μοντέλο που θα χρησιμοποιήσουμε είναι το LSTM, καθώς έδωσε τα καλύτερα αποτελέσματα στο σύνολο δεδομένων με τις καταιγίδες. Επίσης, αποφασίσαμε να εφαρμόσουμε την τεχνική σε ένα από τα δύο μικρότερα σύνολα δεδομένων, και συγκεκριμένα στο σύνολο δεδομένων με τις κοινωνικές καταστροφές όπου το καλύτερο accuracy ήταν 89.91%, και όχι στο σύνολο δεδομένων με τις βιομηχανικές καταστροφές, όπου το καλύτερο accuracy ήταν 96.05% και που είναι ήδη εξαιρετικό.

Για αυτό το πείραμα αποθηκεύουμε όλες τις παραμέτρους του LSTM μοντέλου. Το νέο μοντέλο θα είναι ακριβώς ίδιο με αυτό το LSTM με εξαίρεση το επίπεδο εξόδου, το οποίο θα εκπαιδευτεί από την αρχή. Αυτό το πετυχαίνουμε φορτώνοντας τις αποθηκευμένους παραμέτρους στο νέο μοντέλο και αντικαθιστώντας το τελευταίο επίπεδο με ένα νέο. Το μοντέλο αυτό θα έχει τις ίδιες τιμές στις υπερ-παραμέτρους: αριθμός κρυφών κόμβων (60) και πιθανότητα στο επίπεδο dropout (0.39446). Για την εκπαίδευσή του χρησιμοποιούμε τον αλγόριθμο βελτιστοποίησης Adam και τη BCE ως συνάρτηση κόστους. Στις μόνες παραμέτρους που εκτελούμε βελτιστοποίηση είναι ο ρυθμός μάθησης (learning rate) και το μέγεθος του batch. Το εύρος των τιμών που τέθηκε σε αυτές τις παραμέτρους είναι το ίδιο που τέθηκε σε όλα τα LSTMs μοντέλα. Δηλαδή, για τον ρυθμό μάθησης το εύρος είναι 0.0007-0.003,

ενώ για το μέγεθος του batch ως πιθανές τιμές τίθενται το 32 και το 64, όπως σε όλα τα προηγούμενα μοντέλα για το συγκεκριμένο σύνολο δεδομένων. Η βελτιστοποίηση των υπερ-παραμέτρων σε αυτήν την περίπτωση, όπως αναφέραμε και στο κεφάλαιο της μεθοδολογίας, πραγματοποιήθηκε με τη βιβλιοθήκη Optuna.

Σημειώνουμε ότι ο καλύτερος συνδυασμός υπερ-παραμέτρων επιλέχθηκε με βάση το accuracy στα δεδομένα επαλήθευσης, ενώ εξετάστηκαν 20 διαφορετικοί συνδυασμοί τιμών των υπερ-παραμέτρων. Επιπλέον, ο καθορισμός των εποχών σε όλα τα μοντέλα έγινε με τον πρόωρο τερματισμό, όπου ο αρχικός αριθμός των εποχών τέθηκε ίσος με 150 και ο προκαθορισμένος αριθμός εποχών όπου δεν παρουσιάζει βελτίωση το μοντέλο στα δεδομένα επαλήθευσης (patience) τέθηκε ίσος με 10. Επιπλέον, το κριτήριο με το οποίο ελέγχει ο πρόωρος τερματισμός για το ενδεχόμενο βελτίωσης είναι το σφάλμα (loss) στα δεδομένα επαλήθευσης, όπως σε όλα τα προηγούμενα πειράματα. Με βάση τα αποτελέσματα της βελτιστοποίησης, το μέγεθος του batch πήρε την τιμή 32, ενώ ο ρυθμός μάθησης την τιμή 0.00286.

Όπως ήταν αναμενόμενο, η εκπαίδευση διήρκησε μόνο δύο εποχές. Το accuracy της καλύτερης εποχής στα δεδομένα επαλήθευσης είναι 89.51%, ενώ οι τιμές στα μεγέθη ανάκληση, ακρίβεια και f1 score είναι αντίστοιχα 89.6%, 89.4% και 89.5%. Παρατηρούμε ότι όλες οι μετρικές έχουν αρκετά κοντινές τιμές μεταξύ τους που σημαίνει ότι ο αριθμός των λαθών ανάμεσα στις δύο κλάσεις είναι παρόμοιος, κάτι που φαίνεται και από τον πίνακα σύγχυσης στην παρακάτω εικόνα. Από τον πίνακα αυτόν βλέπουμε επίσης ότι το μοντέλο κάνει περισσότερα λάθη στην πρόβλεψη των tweets που είναι σχετικά με την καταστροφή.



Εικόνα 29: Πίνακας σύγχυσης (confusion matrix) για την τεχνική fine-tuning για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Το αποτέλεσμα της τεχνικής fine-tuning είναι αρκετά ικανοποιητικό. Το αποτέλεσμα δεν είναι το καλύτερο για αυτό το σύνολο δεδομένων, το οποίο δίνεται από το μοντέλο LSTM και είναι 89.91%, αλλά είναι το δεύτερο καλύτερο, καθώς όλα τα υπόλοιπα μοντέλα έδωσαν accuracy κάτω από 89%. Συμπεραίνουμε λοιπόν ότι σε αυτήν την περίπτωση είναι αρκετά αποτελεσματική αυτή η τεχνική, ενώ ταυτόχρονα ο χρόνος βελτιστοποίησης και εκπαίδευσης είναι πολύ μικρότερος.

5.2.2 Προ-εκπαιδευμένα Γλωσσικά Μοντέλα

Σε αυτήν την ενότητα παρουσιάζουμε τα αποτελέσματα των πειραμάτων για τα προ-εκπαιδευμένα γλωσσικά μοντέλα. Σε όλα τα πειράματα τέθηκαν οι τιμές στις υπερ-παραμέτρους που προτείνουν οι συγγραφείς του κάθε μοντέλου. Τα αποτελέσματα τα παρουσιάζουμε ανά σύνολο δεδομένων, ώστε να

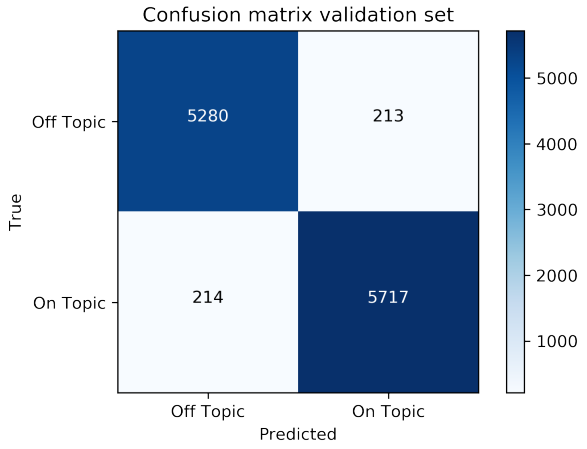
μπορέσουμε να κάνουμε σύγκριση. Σημειώνουμε επίσης ότι όλα τα αποτελέσματα προέρχονται από την καλύτερη επόχη με βάση τον πρόωρο τερματισμό, ενώ αφορούν μόνο τα δεδομένα επαλήθευσης.

Στον ακόλουθο πίνακα βλέπουμε τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το σύνολο δεδομένων με τις καταιγίδες. Αρχικά, όλες οι μετρικές έχουν παρόμοιες τιμές, που σημαίνει ότι τα μοντέλα δεν δυσκολεύονται περισσότερο στην πρόβλεψη κάποιας κλάσης. Παρατηρούμε επίσης ότι τα αποτελέσματα όλων των μοντέλων είναι εξαιρετικά, κοντά στο 96%, ενώ έχουν μικρή διαφορά μεταξύ τους. Συγκεκριμένα, το χειρότερο accuracy το έχει το μοντέλο DeBERTa, ενώ το ακολουθούν τα μοντέλα XLNet και ELECTRA. Το BERT δίνει σχεδόν το ίδιο accuracy με το DistilBERT, ενώ συνολικά το καλύτερο αποτέλεσμα για το σύνολο δεδομένων με τις καταιγίδες το δίνει το DistilBERT.

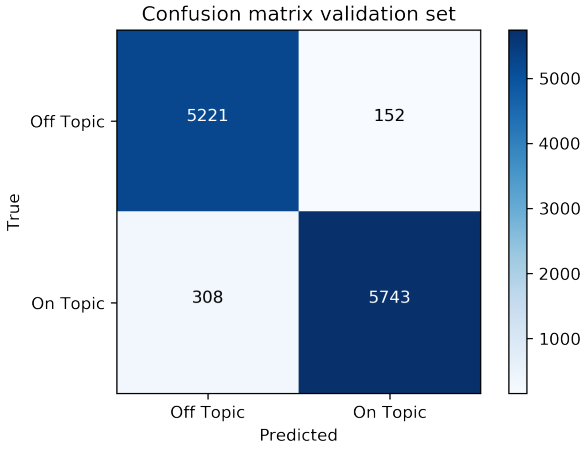
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
BERT	96.26	96.3	96.3	96.3
DistilBERT	**96.29**	**96.3**	**96.3**	**96.3**
DeBERTa	95.47	95.5	95.5	95.5
XLNet	95.63	95.7	95.6	95.6
ELECTRA	95.97	96.0	95.9	96.0

Πίνακας 20: Τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το σύνολο δεδομένων με τις καταιγίδες.

Από τους αντίστοιχους πίνακες σύγχυσης παρατηρούμε ότι όντως τα λάθη ανάμεσα στις δύο κλάσεις είναι κατανεμημένα σχεδόν ισόνομα, με εξαίρεση τα μοντέλα XLNet και ELECTRA που κάνουν περίπου τα διπλάσια σφάλματα στην πρόβλεψη των tweets που είναι σχετικά την καταστροφή. Ενδεικτικά παραθέτουμε τους πίνακες σύγχυσης από τα μοντέλα BERT και ELECTRA, όπου φαίνονται οι παρατηρήσεις μας.



Εικόνα 30: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο BERT για το σύνολο δεδομένων με τις καταιγίδες.



Εικόνα 31: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο ELECTRA για το σύνολο δεδομένων με τις καταιγίδες.

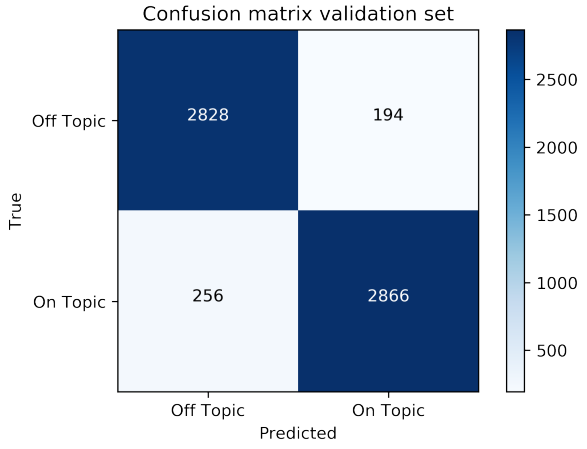
Τα αποτελέσματα για το σύνολο δεδομένων με τις πλημμύρες για όλα τα προ-εκπαιδευμένα γλωσσικά μοντέλα φαίνονται στον παρακάτω πίνακα. Αρχικά, παρατηρούμε ότι όλα τα μεγέθη έχουν πολύ κοντινές τιμές ανά μοντέλο, με εξαίρεση το ELECTRA όπου υπάρχει μία απόκλιση στην ακρίβεια, που σημαίνει ότι το μοντέλο πιθανώς κάνει περισσότερα σφάλματα στην πρόβλεψη των tweets μιας κλάσης. Επιπλέον,

το accuracy όλων των μοντέλων είναι κοντά στο 93%, ενώ υπάρχουν μικρές διαφορές στις τιμές. Την χειρότερη τιμή την παρουσιάζει το μοντέλο XLNet, ενώ πολύ κοντά του βρίσκεται το DeBERTa. Επίσης, τα μοντέλα BERT και ELECTRA έχουν παρόμοια αποτελέσματα, ενώ το καλύτερο accuracy το δίνει το DistilBERT.

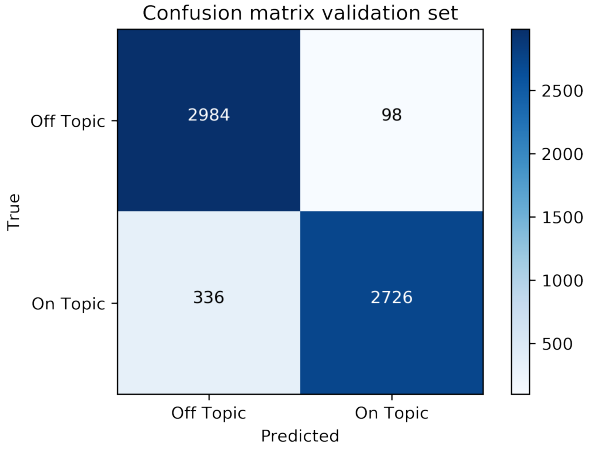
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
BERT	92.90	92.9	93.0	92.9
DistilBERT	**93.16**	**93.2**	**93.3**	**93.2**
DeBERTa	92.74	92.7	92.8	92.7
XLNet	92.68	92.7	92.7	92.7
ELECTRA	92.94	92.9	93.2	92.9

Πίνακας 21: Τα αποτελέσματα για το σύνολο δεδομένων με τις πλημμύρες όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων.

Από τους πίνακες σύγχυσης για το σύνολο δεδομένων με τις πλημμύρες βλέπουμε ότι όλα τα μοντέλα κάνουν περισσότερα σφάλματα στην πρόβλεψη των tweets που είναι σχετικά με την καταστροφή. Αυτό μπορούμε να το παρατηρήσουμε και από τους δύο ακόλουθους πίνακες σύγχυσης, που αφορούν τα μοντέλα XLNet και ELECTRA αντίστοιχα. Συγκεκριμένα, στο ELECTRA υπάρχει αρκετή απόκλιση στις τιμές των σφαλμάτων μεταξύ των δύο κλάσεων, όπως σωστά υποθέσαμε από την απόκλιση των μετρικών αξιολόγησης.



Εικόνα 32: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο XLNet για το σύνολο δεδομένων με τις πλημμύρες.



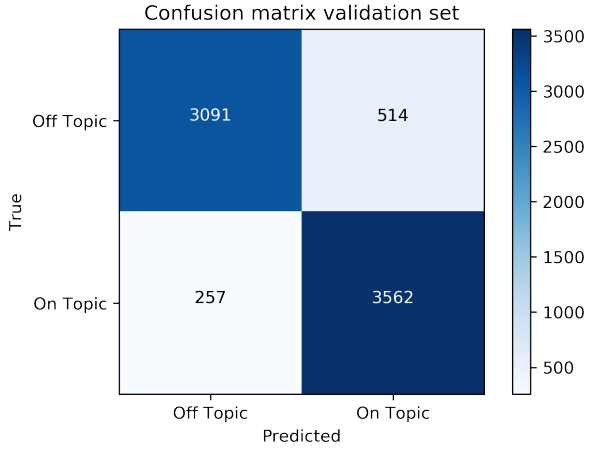
Εικόνα 33: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο ELECTRA για το σύνολο δεδομένων με τις πλημμύρες.

Στον επόμενο πίνακα παρουσιάζουμε τα αποτελέσματα για το σύνολο δεδομένων με τους σεισμούς για όλα τα προ-εκπαιδευμένα γλωσσικά μοντέλα. Όπως σε όλα τα προγούμενα μοντέλα, βλέπουμε ότι οι μετρικές έχουν παρόμοιες τιμές με εξαίρεση το μοντέλο DeBERTa, όπου η ακρίβεια είναι λίγο μεγαλύτερη από τα υπόλοιπα. Συνεπώς, αυτό σημαίνει ότι θα υπάρχει μία εμφανής απόκλιση στα λάθη που κάνει το μοντέλο στις δύο κλάσεις. Επίσης, παρατηρούμε ότι το χειρότερο accuracy το παρουσιάζει το μοντέλο ELECTRA, ενώ τα μοντέλα DeBERTA και XLNet που το ακολουθούν δίνουν πρακτικά το ίδιο αποτέλεσμα. Επιπλέον, το BERT δίνει ένα αρκετά καλό accuracy, ενώ τελικά το υψηλότερο accuracy με αρκετή διαφορά από τα υπόλοιπα μοντέλα το δίνει το DistilBERT.

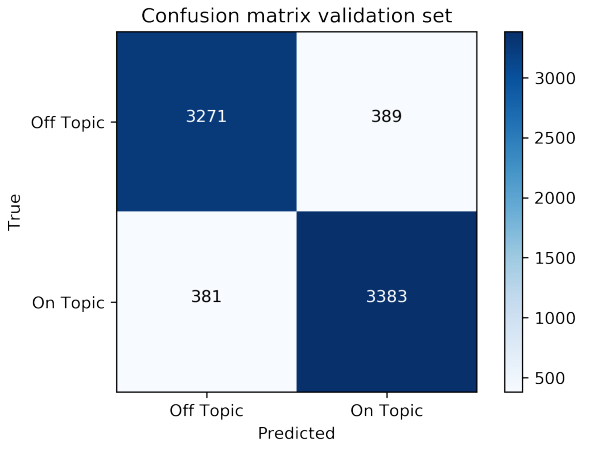
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
BERT	90.41	90.4	90.4	90.4
DistilBERT	**90.72**	**90.8**	**90.7**	**90.7**
DeBERTa	89.62	89.5	89.9	89.6
XLNet	89.63	89.6	89.6	89.6
ELECTRA	89.09	89.1	89.1	89.1

Πίνακας 22: Τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το σύνολο δεδομένων με τους σεισμούς.

Από τους πίνακες σύγχυσης για το σύνολο δεδομένων με τους σεισμούς παρατηρούμε ότι τα περισσότερα λάθη γίνονται στην πρόβλεψη των tweets που δεν είναι σχετικά με την καταστροφή. Παραθέτουμε τους πίνακες σύγχυσης από τα δύο μοντέλα που έδωσαν σχεδόν το ίδο accuracy, δηλαδή το DeBERTa και το XLNet. Όπως ήταν αναμενόμενο από τις παραπάνω μετρικές, υπάρχει μικρή διαφορά λαθών μεταξύ των δύο κλάσεων στην περίπτωση του XLNet και εμφανής διαφορά στην περίπτωση του DeBERTA.



Εικόνα 34: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο DeBERTa για το σύνολο δεδομένων με τους σεισμούς.



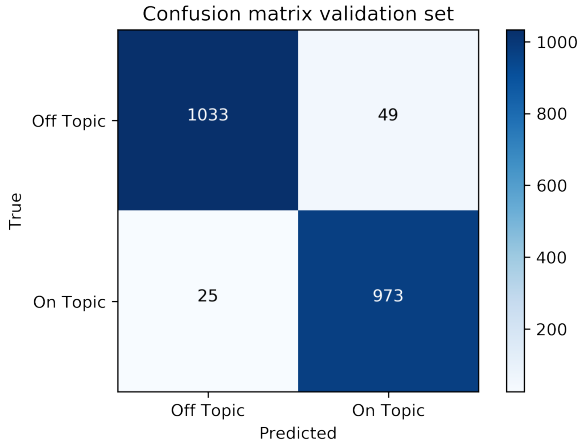
Εικόνα 35: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο XLNet για το σύνολο δεδομένων με τους σεισμούς.

Τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές παρατίθενται στον παρακάτω πίνακα. Αρχικά, παρατηρούμε ότι υπάρχει μία πολύ μικρή διαφορά στα μεγέθη αξιολόγησης σχεδόν σε όλα τα μοντέλα, οπότε περιμένουμε να γίνονται περισσότερα λάθη στη μία ή στην άλλη κλάση. Επιπλέον, βλέπουμε ότι όλα τα μοντέλα έχουν πολύ ικανοποιητικά αποτελέσματα, γύρω στο 96%, με το XLNet να έχει το χαμηλότερο accuracy. Το ακολουθεί το BERT με accuracy πολύ κοντινο σε αυτό του XLNet. Στη συνέχεια, ακολουθούν τα μοντέλα ELECTRA και DistilBERT, ενώ τελικά τα καλύτερα αποτελέσματα για το συγκεκριμένο σύνολο δεδομένων τα δίνει με διαφορά το DeBERTa.

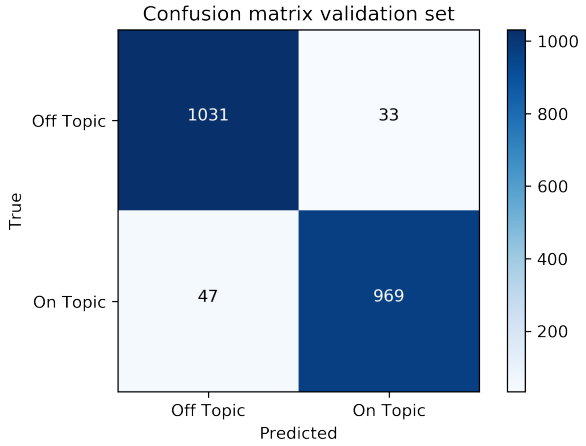
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
BERT	95.48	95.6	95.5	95.5
DistilBERT	96.44	96.5	96.4	96.4
DeBERTa	**96.83**	**96.8**	**96.8**	**96.8**
XLNet	95.43	95.4	95.5	95.4
ELECTRA	96.15	96.1	95.2	96.2

Πίνακας 23: Τα αποτελέσματα για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων.

Παραθέτουμε ενδεικτικά τους πίνακες σύγχυσης των μοντέλων DistilBERT και ELECTRA. Όπως παρατηρούμε από τις παρακάτω εικόνες, το ένα μοντέλο κάνει περισσότερα λάθη στη μία κλάση, ενώ το άλλο κάνει περισσότερα λάθη στην άλλη κλάση. Οι αποκλίσεις στα λάθη είναι αρκετά μικρές αλλά καθώς το σύνολο δεδομένων είναι αρκετα μικρό είναι λογικό να επηρεάζονται τα μεγέθη αξιολόγησης. Τέλος, αυτές οι παρατηρήσεις επεκτείνονται στους πίνακες σύγχυσης των υπόλοιπων μοντέλων.



Εικόνα 36: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο DistilBERT για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές.



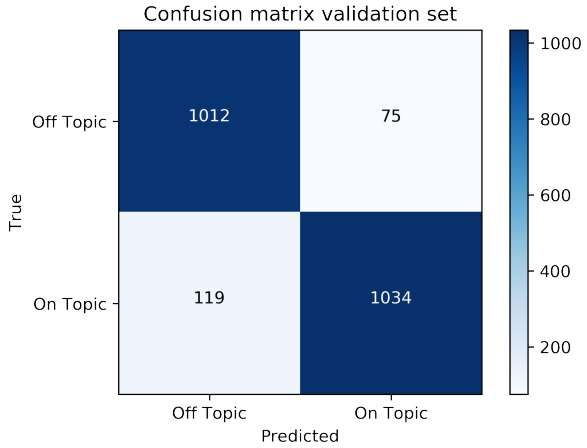
Εικόνα 37: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο ELECTRA για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές.

Τέλος, στον παρακάτω πίνακα βλέπουμε τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το σύνολο δεδομένων με τις κοινωνικές καταστροφές. Όλα τα μεγέθη αξιολόγησης έχουν παρόμοιες τιμές μεταξύ τους στο κάθε μοντέλο, με εξαίρεση το μοντέλο ELECTRA όπου υπάρχει μία απόκλιση. Επιπλέον, βλέπουμε ότι με εξαίρεση το DistilBERT, όλα τα υπόλοιπα μοντέλα έδωσαν παρόμοια αποτελέσματα. Το χαμηλότερο accuracy το έχουν τα μοντέλα XLNet και ELECTRA, ενώ το υψηλότερο accuracy το έδωσε το DistilBERT με φανερή διαφορά.

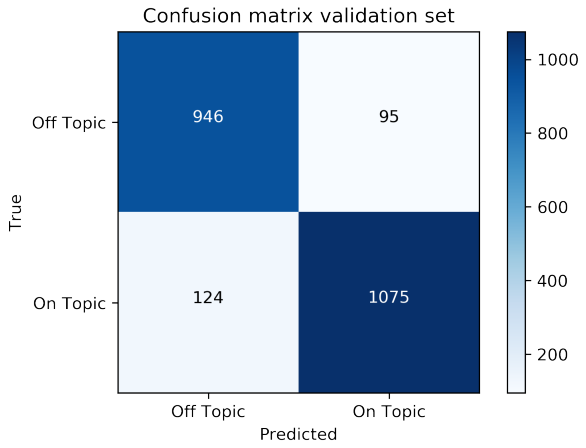
Μοντέλο	Accuracy(%)	Recall(%)	Precision(%)	F1 score(%)
BERT	90.36	90.4	90.4	90.4
DistilBERT	**91.34**	**91.4**	**91.4**	**91.3**
DeBERTa	90.54	90.7	90.6	90.5
XLNet	90.13	90.1	90.1	90.1
ELECTRA	90.22	90.3	90.1	90.2

Πίνακας 24: Τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Από τους πίνακες σύγχυσης για το σύνολο δεδομένων με τις κοινωνικές καταστροφές παρατηρούμε ότι όλα τα μοντέλα κάνουν περισσότερα λάθη στην πρόβλεψη των tweets που είναι σχετικά με την καταστροφή. Ωστόσο, όπως βλέπουμε και από τις παρακάτω δύο εικόνες, οι διαφορές στα λάθη ανάμεσα στις δύο κλάσεις είναι σχετικά μικρές.



Εικόνα 38: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο DistilBERT για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.



Εικόνα 39: Πίνακας σύγχυσης (confusion matrix) για το μοντέλο ELECTRA για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Συνολικά, τα καλύτερα αποτελέσματα σχεδόν σε όλα τα σύνολα δεδομένων τα έχει το DistilBERT, με εξαίρεση το σύνολο δεδομένων με τις βιομηχανικές καταστροφές όπου το καλύτερο accuracy το δίνει το DeBERTa. Ωστόσο, ακόμα και σε αυτήν την περίπτωση το δεύτερο καλύτερο αποτέλεσμα το δίνει το DistilBERT. Αυτό αποτελεί έκπληξη δεδομένου ότι το DistilBERT είναι μία μικρότερη έκδοση του BERT. Όμως, το μοντέλο αυτό έχει πολύ μικρότερο υπολογιστικό κόστος από τα υπόλοιπα, οπότε θα μπορούσαμε να πούμε ότι πήραμε το ιδανικό αποτέλεσμα. Δηλαδή, ακόμα και στην περίπτωση του συνόλου δεδομένων με τις βιομηχανικές καταστροφές όπου το καλύτερο αποτέλεσμα το δίνει το DeBERTa, ίσως τελικά να είναι προτιμότερο να επιλέξουμε το DistilBERT, καθώς το DeBERTa είναι το μοντέλο με το μεγαλύτερο υπολογιστικό κόστος. Αντιθέτως, το ποιο μοντέλο παρουσιάζει το χειρότερο αποτέλεσμα ποικίλει ανάλογα με το σύνολο δεδομένων. Όμως, συνολικά μπορούμε να πούμε ότι το XLNet δίνει το χαμηλότερο accuracy, ενώ ακολουθούν το DeBERTa και το ELECTRA.

5.2.3 Επίδραση του Μεγέθους του Batch στα Προ-εκπαιδευμένα Γλωσσικά Μοντέλα

Καθώς δεν κάναμε βελτιστοποίηση των υπερ-παραμέτρων στα προ-εκπαιδευμένα γλωσσικά μοντέλα, λόγω μεγάλης διάρκειας του fine-tuning, αποφασίσαμε να εκτιμήσουμε την επίδραση μιας υπερ-παραμέτρου. Επιλέξαμε το μέγεθος του batch καθώς είναι συγκεκριμένες οι τιμές που μπορούμε να του δώσουμε, ενώ σε κάποια μοντέλα, όπως είναι το DeBERTa και το XLNet, οι συγγραφείς πρότειναν και batch μεγαλύτερου μεγέθους. Τα δύο επιπλέον μεγέθη του batch που εξετάζουμε είναι το 64 και το 128. Σημειώνουμε ότι δεν αλλάζουμε καμία άλλη τιμή στις υπερ-παραμέτρους του κάθε μοντέλου. Τα αποτελέσματα τα παρουσιάζουμε ανά σύνολο δεδομένων, ενώ εξετάζουμε μόνο το μέγεθος accuracy. Επίσης, προσθέτουμε το accuracy των αντίστοιχων αποτελεσμάτων της προηγούμενης ενότητας, όπου σε όλες τις περιπτώσεις το μέγεθος του batch τέθηκε ίσο με 32, ώστε να μπορεί να γίνει σύγκριση. Τέλος, όλα τα αποτελέσματα προέρχονται από την καλύτερη εποχή με βάση τον πρόωρο τερματισμό, ενώ αφορούν μόνο τα δεδομένα επαλήθευσης.

Στον παρακάτω πίνακα βλέπουμε τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τις καταιγίδες. Αρχικά, για το μοντέλο BERT παρατηρούμε ότι όλες οι τιμές για το μέγεθος accuracy είναι κοντά μεταξύ τους, ενώ την χειρότερη τιμή την έχουμε για μέγεθος του batch ίσο με 64 και την καλύτερη για μέγεθος του batch ίσο με 128. Για το μοντέλο DistilBERT, βλέπουμε ότι επίσης όλα τα αποτελέσματα είναι κοντά μεταξύ τους. Όπως και στο μοντέλο BERT, το χειρότερο accuracy το παίρνουμε για μέγεθος του batch ίσο με 64 και το καλύτερο για μέγεθος του batch ίσο με 128. Όσον αφορά το DeBERTa, παρατηρούμε ότι υπάρχει μία αύξηση στο accuracy όσο αυξάνεται και το μέγεθος του batch, ενώ το ίδιο ισχύει και για το XLNet. Αντίθετα, στο μοντέλο ELECTRA παρατηρείται μία μείωση στη μετρική καθώς αυξάνεται το μέγεθος του batch, αλλά τα αποτελέσματα είναι πολύ κοντά μεταξύ τους. Συνολικά, το μοντέλο DistilBERT συνεχίζει όπως και πριν να έχει το καλύτερο αποτέλεσμα, αυτή τη φορά για μέγεθος του batch ίσο με 128. Το χαμηλότερο accuracy όπως και πριν το έχουμε από το μοντέλο DeBERTa, αλλά το χαμηλότερο από τα καλύτερα accuracy ανά μοντέλο το παρουσιάζει το μοντέλο XLNet. Το BERT συνεχίζει να έχει το δεύτερο καλύτερο accuracy, ενώ πλέον το τρίτο καλύτερο αποτέλεσμα το έχει το DeBERTa. Τέλος, παρατηρούμε ότι για το σύνολο δεδομένων με τις καταιγίδες η αύξηση του μεγέθους του batch έφερε καλύτερα αποτελέσματα.

	Μέγεθος του batch		
Μοντέλο	32	64	128
BERT	96.26	96.15	96.34
DistilBERT	96.29	95.89	**96.39**
DeBERTa	95.47	96.07	96.12
XLNet	95.63	95.67	95.73
ELECTRA	95.97	95.86	95.82

Πίνακας 25: Το accuracy(%) όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τις καταιγίδες.

Τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τις πλημμύρες παρατίθενται στον ακόλουθο πίνακα. Πρώτα από όλα, για το μοντέλο BERT βλέπουμε ότι υπάρχει μία αισθητή αύξηση στο accuracy για τα δύο μεγαλύτερα μεγέθη του batch. Την καλύτερη τιμή την έχουμε για μέγεθος του batch ίσο με 64 και τη χειρότερη για μέγεθος του batch ίσο με 32. Επιπλέον, για το DistilBERT, παρατηρούμε ότι όπως και στο σύνολο δεδομένων με τις καταιγίδες, το υψηλότερο accuracy το παίρνουμε για μέγεθος του batch ίσο με 128 και το χαμηλότερο για μέγεθος του batch ίσο με 64. Όσον αφορά το DeBERTa, παρατηρούμε

ότι υπάρχει μία μείωση και μία αύξηση για τα μεγέθη του batch 64 και 128 αντίστοιχα, συγκριτικά με το αποτέλεσμα για το μέγεθος του batch ίσο με 32. Αντίθετα, στο XLNet υπάρχει μία αύξηση στο accuracy παράλληλα με την αύξηση του μεγέθους του batch, ενώ στο μοντέλο ELECTRA το καλύτερο accuracy το έχουμε για μέγεθος του batch ίσο με 64 και το χειρότερο για μέγεθος του batch ίσο με 128. Συνολικά, το καλύτερο αποτέλεσμα αυτή τη φορά το δίνει το XLNet, ενώ το ακολουθεί το BERT και με μικρή διαφορά το DistilBERT. Παρόλο που το χαμηλότερο accuracy πριν το έδινε το XLNet, τώρα το δίνει το DeBERTa, ενώ το DeBERTa δίνει επίσης το χαμηλότερο από τα καλύτερα accuracy ανά μοντέλο. Τέλος, παρατηρούμε ότι για το συγκεκριμένο σύνολο δεδομένων η αύξηση του μεγέθους του batch έφερε καλύτερα αποτελέσματα σε όλες τις περιπτώσεις.

	Μέγεθος του batch		
Μοντέλο	32	64	128
BERT	92.90	93.46	93.33
DistilBERT	93.16	92.68	93.44
DeBERTa	92.74	92.55	93.02
XLNet	92.68	92.73	**93.62**
ELECTRA	92.94	93.33	92.82

Πίνακας 26: Το accuracy(%) για το σύνολο δεδομένων με τις πλημμύρες όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch.

Στον επόμενο πίνακα παρουσιάζουμε τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τους σεισμούς. Αρχικά, για το μοντέλο BERT παρατηρούμε ότι το καλύτερο accuracy το έχουμε για μέγεθος του batch ίσο με 128 και το χειρότερο για μέγεθος του batch ίσο με 64. Αντίστοιχα, για το DistilBERT μοντέλο την υψηλότερη τιμή την παίρνουμε για μέγεθος του batch ίσο με 32, ενώ τη χαμηλότερη για μέγεθος του batch ίσο με 64. Ύστερα, στο μοντέλο DeBERTa παρατηρούμε ότι όλες οι τιμές είναι αρκετά κοντινές μεταξύ τους, ενώ υπάρχει μία μείωση και μία αύξηση για τα μεγέθη του batch 64 και 128 αντίστοιχα. Όσον αφορά το XLNet παρατηρούμε μία μείωση κατά την αύξηση του μέγεθος του batch. Αντίθετα, στο ELECTRA βλέπουμε ότι το accuracy αυξάνεται καθώς αυξάνεται το μέγεθος του batch. Συνολικά, το καλύτερο αποτέλεσμα το δίνει το BERT, ενώ το ακολουθεί με μικρή διαφορά το DistilBERT. Στην προηγούμενη υποενότητα είδαμε ότι το χαμηλότερο accuracy το έδινε το ELECTRA, ενώ τώρα το ίδιο μοντέλο δίνει το τρίτο καλύτερο αποτέλεσμα. Επιπλέον, το XLNet δίνει το συνολικά χειρότερο αποτέλεσμα, καθώς και το χαμηλότερο από τα καλύτερα accuracy ανά μοντέλο. Τέλος, παρατηρούμε ότι για το σύνολο δεδομένων με τους σεισμούς η αύξηση του μεγέθους του batch έφερε καλύτερα αποτελέσματα σε μερικές περιπτώσεις και χειρότερα σε άλλες.

	Μέγεθος του batch		
Μοντέλο	32	64	128
BERT	90.41	89.43	**90.87**
DistilBERT	90.72	89.39	90.61
DeBERTa	89.62	89.28	89.82
XLNet	89.63	88.87	88.21
ELECTRA	89.09	89.32	90.13

Πίνακας 27: Το accuracy(%) όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τους σεισμούς.

Τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του

batch για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές παρατίθενται στον παρακάτω πίνακα. Αρχικά, για το μοντέλο BERT βλέπουμε ότι το χειρότερο accuracy το έχουμε για μέγεθος του batch ίσο με 32 και το καλύτερο για μέγεθος του batch ίσο με 64. Επίσης, στο DIstilBERT το καλύτερο αποτέλεσμα το παίρνουμε για μέγεθος του batch ίσο με 32, ενώ το χειρότερο για μέγεθος του batch ίσο με 64. Όσον αφορά το DeBERTa, παρατηρούμε ότι υπάρχει μία μικρή μείωση και μία μικρή αύξηση στη μετρική για τα μεγέθη του batch 64 και 128 αντίστοιχα. Επιπλέον, στα μοντέλα XLNet και ELECTRA το accuracy αυξάνεται με την αύξηση του μεγέθους του batch. Συνολικά, όπως και πριν το καλύτερο αποτέλεσμα το δίνει το DeBERTa, ενώ ακολουθεί το XLNet και το DistilBERT με μικρή διαφορά από το XLNet. Το συνολικά χαμηλότερο accuracy το παίρνουμε από το μοντέλο XLNet, αλλά το χαμηλότερο από τα καλύτερα accuracy ανά μοντέλο το δίνει το ELECTRA. Τέλος, παρατηρούμε ότι για το συγκεκριμένο σύνολο δεδομένων η αύξηση του μεγέθους του batch έφερε καλύτερα αποτελέσματα.

	Μέγεθος του batch		
Μοντέλο	32	64	128
BERT	95.48	96.34	96.19
DistilBERT	96.44	95.95	96.00
DeBERTa	96.83	96.53	**97.02**
XLNet	95.43	96.00	96.48
ELECTRA	96.15	96.19	96.24

Πίνακας 28: Το accuracy(%) για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch.

Τέλος, στον ακόλουθο πίνακα βλέπουμε τα αποτελέσματα όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τις κοινωνικές καταστροφές. Στο μοντέλο BERT παρατηρούμε μία αύξηση και μία μεγάλη μείωση της μετρικής για τα μεγέθη του batch 64 και 128 αντίστοιχα. Αντίθετα, στο DistilBERT υπάρχει μία μεγάλη μείωση και μία ελάχιστη αύξηση του accuracy για τα μεγέθη του batch 64 και 128 αντίστοιχα. Όσον αφορά το DeBERTa, το accuracy μειώνεται με την αύξηση του μεγέθους του batch. Επίσης, για το μοντέλο XLNet το χειρότερο αποτέλεσμα το παίρνουμε για μέγεθος του batch ίσο με 64 και το καλύτερο για μέγεθος του batch ίσο με 32, ενώ η ίδια αντιστοιχία ισχύει και για το μοντέλο ELECTRA, μόνο που υπάρχουν μικρότερες διαφορές μεταξύ των τιμών. Συνολικά, το DistilBERT συνεχίζει να έχει το καλύτερο αποτέλεσμα, ενώ ακολουθεί το BERT. Παράλληλα το BERT έχει το συνολικά χειρότερο accuracy, ενώ το χαμηλότερο από τα καλύτερα accuracy ανά μοντέλο το δίνει όπως και πριν το XLNet. Τέλος, παρατηρούμε ότι για το σύνολο δεδομένων με τις κοινωνικές καταστροφές η αύξηση του μεγέθους του batch έφερε χειρότερα αποτελέσματα.

	Μέγεθος του batch		
Μοντέλο	32	64	128
BERT	90.36	91.12	88.79
DistilBERT	91.34	89.69	**91.36**
DeBERTa	90.54	90.27	89.61
XLNet	90.13	89.02	89.20
ELECTRA	90.22	89.87	90.17

Πίνακας 29: Το accuracy(%) όλων των προ-εκπαιδευμένων γλωσσικών μοντέλων για το κάθε μέγεθος του batch για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Συμπερασματικά, στο μοντέλο BERT το καλύτερο αποτέλεσμα στο κάθε σύνολο δεδομένων το παίρνουμε για μέγεθος του batch 64 ή 128, αλλά ποτέ για μέγεθος του batch 32, που είναι και αυτό που προτείνουν οι συγγραφείς. Επιπλέον, στο DistilBERT σε όλα τα σύνολα δεδομένων το χειρότερο αποτέλεσμα το έχουμε για μέγεθος του batch ίσο με 64, ενώ το καλύτερο accuracy το παίρνουμε για μέγεθος του batch 32 ή 128. Σε όλες τις περιπτώσεις όμως, οι διαφορές είναι πολύ μικρές ανάμεσα στα αποτελέσματα μεταξύ αυτών των δύο μεγεθών του batch. Όσον αφορά το DeBERTa,το υψηλότερο accuracy το συναντάμε σε όλα τα σύνολα δεδομένων για μέγεθος του batch ίσο με 128, με εξαίρεση το σύνολο δεδομένων με τις κοινωνικές καταστροφές. Θυμίζουμε ότι σε αυτό το μοντέλο οι συγγραφείς προτείνουν μέγεθος του batch μέχρι 64, αλλά σίγουρα το αποτέλεσμα προκαλεί μικρότερη έκπληξη από ότι για τα μοντέλα BERT και DistilBERT, όπου οι συγγραφείς προτείνουν μέγεθος του batch 16 ή 32. Επίσης, στο μοντέλο XLNet τα καλύτερα αποτελέσματα παρατηρούνται συνήθως για μέγεθος του batch ίσο με 128 και σε κάποιες περιπτώσεις για μέγεθος του batch ίσο με 32. Αυτή η παρατήρηση δεν αποτελεί έκπληξη καθώς οι συγγραφείς προτείνουν μέγεθος του batch μέχρι 128. Αντίθετα, στο μοντέλο ELECTRA το υψηλότερο, καθώς και το χαμηλότερο accuracy τα συναντάμε για όλα τα μεγέθη του batch. Συνολικά, τα καλύτερα αποτελέσματα ανά μοντέλο και ανά σύνολο δεδομένων τα παίρνουμε για μέγεθος του batch 128 στις τουλάχιστον μισές περιπτώσεις και για μέγεθος του batch 32 σχεδόν στις υπόλοιπες περιπτώσεις, ενώ τα χειρότερα αποτελέσματα τα έχουμε για μέγεθος του batch 64 επίσης στις μισές περιπτώσεις.

Αντίθετα από τα συμπεράσματα τις προηγούμενης υποενότητας όπου το DistilBERT είχε τα καλύτερα αποτελέσματα σε σχεδόν όλες τις περιπτώσεις, πλέον έχει το καλύτερο accuracy σε δύο μόνο σύνολα δεδομένων, στο σύνολο δεδομένων με τις καταιγίδες και στο σύνολο δεδομένων με τις κοινωνικές καταστροφές, ενώ το DeBERTa όπως και πριν έχει το καλύτερο αποτέλεσμα στο σύνολο δεδομένων με τις βιομηχανικές καταστροφές. Όμως, αξίζει να σημειωθεί ότι το DistilBERT έχει από τα καλύτερα αποτελέσματα στα υπόλοιπα σύνολα δεδομένων. Επιπλέον, το μοντέλο BERT έχει εξαιρετικά αποτελέσματα καθώς έχει το υψηλότερο accuracy στο σύνολο δεδομένων με τους σεισμούς, ενώ σε άλλα τρία σύνολα δεδομένων έχει το δεύτερο καλύτερο accuracy. Το XLNet επίσης έχει το καλύτερο αποτέλεσμα στο σύνολο δεδομένων με τις πλημμύρες, αλλά εκτός αυτού έχει μάλλον μαζί με το ELECTRA τα συνολικά χειρότερα αποτελέσματα. Τέλος, καθώς σε όλες τις περιπτώσεις το accuracy που δίνει το DistilBERT είναι πολύ κοντά στο καλύτερο, ίσως είναι προτιμότερο να επιλέξουμε αυτό το μοντέλο σε όλα τα σύνολα δεδομένων ώστε να γλιτώσουμε τον υπολογιστικό φόρτο που έχουν τα υπόλοιπα μοντέλα.

5.3 Συνολικά Αποτελέσματα

Σε αυτήν την ενότητα παρουσιάζουμε τα συνολικά αποτελέσματα των πειραμάτων μας. Δηλαδή, συγκρίνουμε τα αποτελέσματα των νευρωνικών δικτύων με τα αποτελέσματα των προ-εκπαιδευμένων γλωσσικών μοντέλων. Συγκεκριμένα, συγκρίνουμε το καλύτερο νευρωνικό δίκτυο με το καλύτερο και το χειρότερο προ-εκπαιδευμένο γλωσσικό μοντέλο ανά σύνολο δεδομένων. Σημειώνουμε ότι όταν αναφέρουμε το χειρότερο προ-εκπαιδευμένο μοντέλο εννοούμε το χαμηλότερο accuracy που προέκυψε συνολικά από τα πειράματα με τα προ-εκπαιδευμένα μοντέλα.

Στον ακόλουθο πίνακα βλέπουμε τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις καταιγίδες. Όπως ήταν αναμενόμενο, το καλύτερο νευρωνικό δίκτυο δίνει με διαφορά χαμηλότερο accuracy από το χειρότερο αποτέλεσμα από τα προ-εκπαιδευμένα μοντέλα. Ωστόσο, και το νευρωνικό δίκτυο, συγκεκριμένα το LSTM, δίνει ένα πολύ ικανοποιητικό αποτέλεσμα. Δεδομένου ότι υπάρχει αισθητή διαφορά στο accuracy του καλύτερου νευρωνικού δικτύου και του καλύτερου accuracy από τα προ-εκπαιδευμένα μοντέλα, περίπου 2%, για το σύνολο δεδομένων με τις καταιγίδες θα επιλέγαμε τελικά το DistilBERT (128). Η επιλογή αυτή είναι καλή και με βάση τον υπολογιστικό φόρτο, καθώς η εκπαίδευση του μοντέλου DistilBERT (128) για το συγκεκριμένο σύνολο δεδομένων διήρκεσε 2.95

ώρες, ενώ η αντίστοιχη εκπαίδευση του DeBERTa (32) 6.83 ώρες και η αντίστοιχη εκπαίδευση του LSTM, συνυπολογίζοντας τον χρόνο για τη βελτιστοποίηση των υπερ-παραμέτρων, 2.05 ώρες.

	Μοντέλο	Accuracy(%)
Καλύτερο νευρωνικό δίκτυο	LSTM	94.34
Χειρότερο προ-εκπαιδευμένο μοντέλο	DeBERTa (32)	95.47
Καλύτερο προ-εκπαιδευμένο μοντέλο	**DistilBERT (128)**	**96.39**

Πίνακας 30: Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις καταιγίδες.

Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις πλημμύρες παρατίθενται στον παρακάτω πίνακα. Όπως και στο προηγούμενο σύνολο δεδομένων βλέπουμε ότι υπάρχει μία αύξηση στο accuracy όσο προχωράμε προς την τελευταία γραμμή του πίνακα. Το χειρότερο προ-εκπαιδευμένο μοντέλο έχει ενδιάμεσα αποτελέσματα ανάμεσα στις άλλες δύο περιπτώσεις, ενώ το καλύτερο προ-εκπαιδευμένο γλωσσικό μοντέλο έχει διαφορά με το καλύτερο νευρωνικό δίκτυο γύρω στο 2%, που είναι μία εμφανής διαφορά. Επιπλέον, μπορούμε να πούμε ότι το αποτέλεσμα του μοντέλου XLNet (128) μπορεί να θεωρηθεί αρκετά αξιόλογο. Όσον αφορά τον χρόνο που διήρκεσε η εκπαίδευση στην κάθε περίπτωση έχουμε 2.05 ώρες (συμπεριλαμβανομένου της βελτιστοποίησης των υπερ-παραμέτρων), 3.3 ώρες και 3.87 ώρες αντίστοιχα. Το μοντέλο XLNet (128) έχει όντως μεγάλο υπολογιστικό φόρτο, αλλά δεδομένου ότι ο χρόνος εκπαίδευσής του είναι μόνο ο διπλάσιος του αντίστοιχου χρόνου για το LSTM και ότι υπάρχει αρκετή διαφορά στο accuracy, θα επιλέγαμε τελικά το XLNet (128) για το σύνολο δεδομένων με τις πλημμύρες.

	Μοντέλο	Accuracy(%)
Καλύτερο νευρωνικό δίκτυο	LSTM	91.72
Χειρότερο προ-εκπαιδευμένο μοντέλο	DeBERTa (64)	92.55
Καλύτερο προ-εκπαιδευμένο μοντέλο	**XLNet (128)**	**93.62**

Πίνακας 31: Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις πλημμύρες.

Στον επόμενο πίνακα παρουσιάζουμε τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τους σεισμούς. Αρχικά, παρατηρούμε ότι όπως και στα προηγούμενα σύνολα δεδομένων, όπως ήταν λογικό, το καλύτερο νευρωνικό δίκτυο δίνει χειρότερο αποτέλεσμα από το χαμηλότερο accuracy από τα προ-εκπαιδευμένα μοντέλα. Ωστόσο, η διαφορά τους είναι μικρή. Δηλαδή αν επιλέγαμε ανάμεσα σε αυτά τα δύο μοντέλα, λογικά θα επιλέγαμε το μοντέλο CNN, καθώς το XLNet (128) έχει πολύ μεγάλο υπολογιστικό φόρτο. Όμως, μόνο το αποτέλεσμα του καλύτερου προ-εκπαιδευμένου μοντέλου μπορεί να θεωρηθεί αρκετά αξιόλογο, ενώ το accuracy που δίνει έχει διαφορά περίπου 3% με το αντίστοιχο accuracy του καλύτερου νευρωνικού δικτύου. Όσον αφορά τον χρόνο που διήρκεσε η εκπαίδευση στην κάθε περίπτωση έχουμε 0.87 ώρες (συμπεριλαμβανομένου της βελτιστοποίησης των υπερ-παραμέτρων), 3.46 ώρες και 3.85 ώρες αντίστοιχα. Παρόλο που το υπολογιστικό κόστος είναι πολύ μεγαλύτερο στην περίπτωση του BERT (128), για το συγκεκριμένο σύνολο δεδομένων θα επιλέγαμε αυτό το μοντέλο λόγω της εμφανούς διαφοράς στο accuracy.

	Μοντέλο	Accuracy(%)
Καλύτερο νευρωνικό δίκτυο	CNN	87.96
Χειρότερο προ-εκπαιδευμένο μοντέλο	XLNet (128)	88.21
Καλύτερο προ-εκπαιδευμένο μοντέλο	**BERT (128)**	**90.87**

Πίνακας 32: Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τους σεισμούς.

Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές τα βλέπουμε στον ακόλουθο πίνακα. Σε αντίθεση με τα προηγούμενα σύνολα δεδομένων βλέπουμε ότι το χειρότερο προ-εκπαιδευμένο μοντέλο έχει εμφανώς χαμηλότερο accuracy από το καλύτερο νευρωνικό δίκτυο, ενώ το καλύτερο προ-εκπαιδευμένο γλωσσικό μοντέλο βελτιώνει το accuracy του καλύτερου νευρωνικού δικτύου κατά μόλις 1%. Αξίζει να σημειωθεί επίσης ότι τα αποτελέσματα των δύο καλύτερων μοντέλων ανά κατηγορία έχουν εξαιρετικά αποτελέσματα. Επιπλέον, ο χρόνος που χρειάστηκε για την εκπαίδευση του εκάστοτε μοντέλου είναι αντίστοιχα 0.33 ώρες (συμπεριλαμβανομένου της βελτιστοποίησης των υπερ-παραμέτρων), 2.22 ώρες και 1.77 ώρες. Παρόλο που το DeBERTa είναι ένα πολύ βαρύ μοντέλο, θα μπορούσαμε να το επιλέξουμε για αυτό το σύνολο δεδομένων καθώς είναι αρκετά μικρό και η εκπαίδευση διαρκεί λιγότερο από δυο ώρες, αλλιώς και το CNN έχει πολύ ικανοποιητικά αποτελέσματα.

	Μοντέλο	Accuracy(%)
Καλύτερο νευρωνικό δίκτυο	CNN	96.05
Χειρότερο προ-εκπαιδευμένο μοντέλο	XLNet (32)	95.43
Καλύτερο προ-εκπαιδευμένο μοντέλο	**DeBERTa (128)**	**97.02**

Πίνακας 33: Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις βιομηχανικές καταστροφές.

Στον παρακάτω πίνακα παρουσιάζουμε τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις κοινωνικές καταστροφές. Όπως και στο σύνολο δεδομένων με τις βιομηχανικές καταστροφές το χειρότερο προ-εκπαιδευμένο μοντέλο δίνει χαμηλότερο accuracy από το καλύτερο νευρωνικό δίκτυο, συγκεκριμένα το LSTM. Από την άλλη, το καλύτερο προ-εκπαιδευμένο μοντέλο έχει καλύτερο accuracy από το LSTM κατά 1.5% περίπου. Η διαφορά δεν είναι πολύ μεγαλή αλλά είναι αρκετή για να θεωρηθεί το αποτέλεσμα του καλύτερου προ-εκπαιδευμένο μοντέλο αρκετά ικανοποιητικό. Όσον αφορά το χρόνο εκπαίδευσης αυτών των μοντέλων έχουμε αντίστοιχα 0.75 ώρες (συμπεριλαμβανομένου της βελτιστοποίησης των υπερ-παραμέτρων), 0.86 ώρες και 0.56 ώρες. Καθώς το υπολογιστικό κόστος του μοντέλου DistilBERT (128) είναι μικρό και δίνει και το καλύτερο accuracy, το μοντέλο αυτό αποτελεί την καλύτερη επιλογή για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

	Μοντέλο	Accuracy(%)
Καλύτερο νευρωνικό δίκτυο	LSTM	89.91
Χειρότερο προ-εκπαιδευμένο μοντέλο	BERT (128)	88.79
Καλύτερο προ-εκπαιδευμένο μοντέλο	**DistilBERT (128)**	**91.36**

Πίνακας 34: Τα συνολικά αποτελέσματα για το σύνολο δεδομένων με τις κοινωνικές καταστροφές.

Ως παρατήρηση στα παραπάνω αποτελέσματα αξίζει να σημειωθεί ότι το καλύτερο προ-εκπαιδευμένο γλωσσικό μοντέλο σε όλα τα σύνολα δεδομένων εκπαιδεύτηκε με μέγεθος του batch ίσο με 128. Σημειώνουμε ότι καθώς όλα τα πειράματα έγιναν για τον ίδιο ρυθμό μάθησης στο αντίστοιχο μοντέλο, τότε πιθανώς για μικρότερο μέγεθος του batch να χρειαζόταν και ένας μικρότερος ρυθμός μάθησης, ώστε να βγει ένα αντίστοιχα καλό αποτέλεσμα με το καλύτερο accuracy που έδωσε το κάθε μοντέλο για το κάθε σύνολο δεδομένων. Συνεπώς, ο έλεγχος της επίδρασης του μεγέθους του batch, έγινε κυρίως ως μία προσπάθεια βελτιστοποίησης αυτής της υπερ-παραμέτρου, ενώ άμα ελέγχαμε συνδυασμούς τιμών για τον ρυθμό μάθησης και το μέγεθος του batch πιθανώς να είχαμε άλλα αποτελέσματα. Επιπλέον, όπως ήταν αναμενόμενο στις περισσότερες περιπτώσεις το χειρότερο προ-εκπαιδευμένο μοντέλο είχε καλύτερο αποτέλεσμα από το καλύτερο νευρωνικό δίκτυο. Αυτό από μόνο του δείχνει ότι το BERT και οι παραλλαγές του αποτελούν μία πολύ σημαντική ανακάλυψη για τον τομέα της Επεξεργασίας Φυσικής Γλώσσας. Τέλος, σε όλα τα σύνολα δεδομένων θεωρήσαμε ότι είναι καλύτερη η επιλογή του

καλύτερου προ-εκπαιδευμένου μοντέλου, ενώ σε όλα τα σύνολα δεδομένων καταφέραμε να έχουμε ένα αξιόλογο αποτέλεσμα.

6 Σύνοψη και Μελλοντική Εργασία

6.1 Σύνοψη

Πολλές έρευνες έχουν δείξει ότι τα τελευταία χρόνια πολλοί άνθρωποι χρησιμοποιούν τα μέσα κοινωνικής δικτύωσης για την ενημέρωση των γεγονότων μιας καταστροφής ή για την κλήση βοήθειας, με αποτέλεσμα να αυξάνεται η ανάγκη για την αυτόματη ανάλυση του περιεχομένου των μηνυμάτων που στέλνουν οι χρήστες. Το πρώτο βήμα για αυτήν την ανάλυση κειμένου είναι η εύρεση των μηνυμάτων που σχετίζονται με το είδος της καταστροφής που μελετάται κάθε φορά, ενώ μπορεί να ακολουθηθεί από άλλες αναλύσεις, όπως είναι η ανάλυση συναισθημάτων ή η ταξινόμηση των μηνυμάτων σε κλάσεις πληροφορίας. Μέχρι στιγμής έχουν πραγματοποιηθεί πολλές έρευνες για καθένα από τα παραπάνω προβλήματα, ενώ στις περισσότερες χρησιμοποιούνται δεδομένα από το Twitter, καθώς το συγκεκριμένο μέσο κοινωνικής δικτύωσης προσφέρει δυνατότητα πρόσβασης σε μεγάλη ποσότητα δεδομένων. Η βιβλιογραφική ανασκόπηση που έγινε στο κεφάλαιο δύο έδειξε ότι σπάνια συγκρίνονται αρχιτεκτονικές νευρωνικών δικτύων με τα σύγχρονα προ-εκπαιδευμένα γλωσσικά μοντέλα, όπως είναι το BERT, ενώ κυριώς συγκρίνονται παραδοσιακές τεχνικές μηχανικής μάθησης με τεχνικές βαθιάς μηχανικής μάθησης, όπου τα νευρωνικά δίκτυα υπερίσχυαν σχεδόν πάντα. Σε αυτή λοιπόν την εργασία, χρησιμοποιήσαμε πολλές μεθόδους βαθιάς μηχανικής μάθησης για την εύρεση δεδομένων από το Twitter που σχετίζονται με καταστροφές.

Για το πρόβλημα που μόλις διατυπώσαμε, χρησιμοποιήσαμε δεδομένα από πολλαπλές πηγές, που όλες περιλαμβάνουν tweets που έχουν ταξινομηθεί από ανθρώπους με βάση τη σχετικότητά τους με την αντίστοιχη καταστροφή. Από αυτά τα σύνολα δεδομένων επιλέξαμε πέντε καταστροφές, τις καταιγίδες, τις πλημμύρες, τους σεισμούς, τις κοινωνικές και τις βιομηχανικές καταστροφές, ενώ συνδυάσαμε όλα τα επιμέρους δεδομένα από αυτούς τους τύπους καταστροφών σε μεγαλύτερα σύνολα δεδομένων. Επίσης, τα νευρωνικά δίκτυα που επιλέξαμε για το πρόβλημά μας ήταν τα MLP Mean, MLP MMM, LSTM, LSTM Attention, BiLSTM, BiLSTM Attention και CNN, ενώ σε όλα αυτά πραγματοποιήσαμε βελτιστοποίηση των αντίστοιχων υπερ-παραμέτρων του κάθε μοντέλου. Από την άλλη πλευρά, τα σύγχρονα προ-εκπαιδευμένα γλωσσικά μοντέλα που επιλέξαμε να εφαρμόσουμε ήταν τα BERT, DistilBERT, DeBERTa, XLNet και ELECTRA, λόγω των σημαντικών διαφορών τους. Σε αυτά τα μοντέλα θέσαμε τις αντίστοιχες τιμές στην κάθε υπερ-παράμετρο που προτείνουν οι συγγραφείς του κάθε μοντέλου.

Τα αποτελέσματα των πειραμάτων για τα νευρωνικά δίκτυα έδειξαν ότι το υψηλότερο accuracy επιτεύχθηκε από το LSTM ή το CNN σε όλα τα σύνολα δεδομένων, ενώ το χαμηλότερο accuracy, όπως ήταν αναμενόμενο, από τα MLPs μοντέλα. Ύστερα, δοκιμάσαμε την τεχνική fine-tuning με τη χρήση του καλύτερου μοντέλου του μεγαλύτερου συνόλου δεδομένων, δηλαδή του μοντέλου LSTM του συνόλου δεδομένων με τις καταιγίδες, στο σύνολο δεδομένων με τις κοινωνικές καταστροφές, που είναι αρκετά μικρό. Το αποτέλεσμα ήταν αρκετά ικανοποιητικό και κοντά στο αποτέλεσμα που έδωσε το καλύτερο μοντέλο για το συγκεκριμένο σύνολο δεδομένων (LSTM). Όσον αφορά τα προ-εκπαιδευμένα μοντέλα, παρατηρήσαμε ότι για τις προτεινόμενες τιμές στις υπερ-παραμέτρους, το καλύτερο αποτέλεσμα σε όλα τα σύνολα δεδομένων το έδωσε το DistilBERT, με την εξαίρεση ενός συνόλου δεδομένων όπου το υψηλότερο accuracy δόθηκε από το DeBERTa. Στη συνέχεια, εκτιμήσαμε την επίδραση του μεγέθους του batch στα προ-εκπαιδευμένα μοντέλα, χρησιμοποιώντας μεγαλύτερες τιμές από αυτές που πρότειναν κυρίως οι συγγραφείς. Τα αποτελέσματα έδειξαν ότι τα περισσότερα μοντέλα έδωσαν το υψηλότερο accuracy για μέγεθος του batch ίσο με 128 και το χαμηλότερο για μέγεθος του batch ίσο με 64, ενώ πλέον διαφορετικά μοντέλα έδωσαν το καλύτερο αποτέλεσμα ανά σύνολο δεδομένων. Τέλος, η σύγκριση των αποτελεσμάτων των νευρωνικών δικτύων με τα αποτελέσματα των προ-εκπαιδευμένων μοντέλων έδειξε ότι στις περισσότερες περιπτώσεις το χειρότερο προ-εκπαιδευμένο γλωσσικό μοντέλο είχε καλύτερο αποτέλεσμα από το καλύτερο νευρωνικό δίκτυο, ενώ τα καλύτερα προ-εκπαιδευμένα

μοντέλα σε όλα τα σύνολα δεδομένων εκπαιδεύτηκαν με μέγεθος του batch ίσο με 128.

Συμπεραίνοντας λοιπόν, τα πειράματά μας έδειξαν ότι καταφέραμε να πετύχουμε ένα ικανοποιητικό αποτέλεσμα σε όλα τα σύνολα δεδομένων, ενώ τα σύγχρονα προ-εκπαιδευμένα γλωσσικά μοντέλα έδωσαν σε όλες τις περιπτώσεις εμφανώς καλύτερα αποτελέσματα από τα νευρωνικά δίκτυα. Συνεπώς, θεωρήσαμε ότι σε όλα τα σύνολα δεδομένων είναι καλύτερη η επιλογή του αντίστοιχου βέλτιστου μοντέλου. Ωστόσο, σε μερικές περιπτώσεις θα μπορούσαμε να θυσιάσουμε λίγη ακρίβεια, ώστε να μειώσουμε πολύ το υπολογιστικό κόστος.

6.2 Μελλοντική Εργασία

Για μελλοντική εργασία, αρχικά θα μπορούσαμε να κάνουμε περισσότερα πειράματα με τα προ-εκπαιδευμένα γλωσσικά μοντέλα. Δηλαδή, αντί να προσθέσουμε ένα πλήρως συνδεδεμένο επίπεδο εξόδου, θα μπορούσαμε να προσθέσουμε ένα MLP ή ένα πιο πολύπλοκο μοντέλο, όπως είναι ένα BiLSTM. Επιπλέον, θα μπορούσαμε να μελετήσουμε την επίδραση άλλων υπερ-παραμέτρων σε αυτά τα μοντέλα. Έπειτα από την ταξινόμηση των tweets με βάση τη σχετικότητά τους με τις καταστροφές, θα μπορούσαμε να αναπτύξουμε ένα μοντέλο για την ανάλυση των συναισθημάτων των tweets που είναι σχετικά με καταστροφές. Ένας τρόπος για να γίνει αυτό θα ήταν μέσω της τεχνικής word embedding. Μία ιδέα θα ήταν τα συναισθήματα να ταξινομηθούν σε θετικά ή αρνητικά και να γίνει μία ανάλυση των κυριότερων θεμάτων στα οποία αναφέρονται τα tweets που εκδηλώνουν αρνητικά συναισθήματα. Τέλος, μία άλλη πρόταση είναι να εφαρμοστούν τα μοντέλα σε μη ταξινομημένα δεδομένα που θα συλλεχθούν από το Twitter για μία πρόσφατη καταστροφή και να αξιολογηθούν τα αποτελέσματα.

Βιβλιογραφία

[1] A. Kruspe, J. Kersten, and F. Klan. "Review article: Detection of informative tweets in crisis events". In: *Natural Hazards and Earth System Sciences Discussions* 2020 (2020), pp. 1–18. DOI: 10.5194/nhess-2020-214. URL: https://nhess.copernicus.org/preprints/nhess-2020-214/.

[2] Matti Wiegmann et al. "Analysis of Detection Models for Disaster-Related Tweets". In: May 2020. DOI: 10.5281/zenodo.3713920.

[3] Andrew McMinn, Yashar Moshfeghi, and Joemon Jose. "Building a large-scale corpus for evaluating event detection on twitter". In: Oct. 2013, pp. 409–418. DOI: 10.1145/2505515.2505695.

[4] Alexandra Olteanu et al. "CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises". In: *ICWSM*. 2014.

[5] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages". In: (May 2016).

[6] Firoj Alam, Ferda Ofli, and Muhammad Imran. "CrisisMMD: Multimodal Twitter Datasets from Natural Disasters". In: *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*. USA, June 2018.

[7] Kevin Stowe et al. "Developing and Evaluating Annotation Procedures for Twitter Data during Hazard Events". In: *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 133–143. URL: https://www.aclweb.org/anthology/W18-4915.

[8] Jens Kersten et al. "Robust Filtering of Crisis-related Tweets". In: *International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. May 2019.

[9] Matti Wiegmann et al. "Disaster Tweet Corpus 2020". In: (Jan. 2020).

[10] Richard Mccreadie, Cody Buntain, and Ian Soboroff. *TREC Incident Streams: Finding Actionable Information on Social Media*. Ed. by Zeno Franco, José González, and José Canós. Sept. 2019. URL: http://eprints.gla.ac.uk/183409/.

[11] Appen Ltd. *Multilingual Disaster Response Messages*. 2020. URL: https://appen.com/datasets/combined-disaster-response-data/.

[12] S. Smith. *Coronavirus (covid19) Tweets*. 2020. URL: https://www.kaggle.com/smid80/coronavirus-covid19-tweets.

[13] Juan M. Banda et al. *A large-scale COVID-19 Twitter chatter dataset for open scientific research - an international collaboration*. Version 59. Zenodo, Apr. 2021. DOI: 10.5281/zenodo.4726268. URL: https://doi.org/10.5281/zenodo.4726268.

[14] Umair Qazi, Muhammad Imran, and Ferda Ofli. *GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information*. 2020. arXiv: 2005.11177 [cs.SI].

[15] J. Rexiline Ragini, P.M. Rubesh Anand, and Vidhyacharan Bhaskar. "Big data analytics for disaster response and recovery through sentiment analysis". In: *International Journal of Information Management* 42 (2018), pp. 13–24. ISSN: 0268-4012. DOI: https://doi.org/10.1016/j.ijinfomgt.2018.05.004. URL: https://www.sciencedirect.com/science/article/pii/S0268401217307843.

[16] Gonzalo Ruz, Pablo A. Henríquez, and Aldo Mascareño. "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers". In: *Future Generation Computer Systems* 106 (May 2020). DOI: 10.1016/j.future.2020.01.005.

[17] Jyoti Singh et al. "Event Classification and Location Prediction from Tweets during Disasters." In: *Annals of Operations Research* 283 (Dec. 2019). DOI: 10.1007/s10479-017-2522-3.

[18] Sukanya Manna and Haruto Nakai. "Comparative Analysis of Different Classifiers on Crisis-Related Tweets: An Elaborate Study". In: Sept. 2019, pp. 77–94. ISBN: 978-3-030-28552-4. DOI: 10.1007/978-3-030-28553-1_4.

[19] Beverly Parilla-Ferrer, Proceso Fernandez, and Jaime IV. "Automatic Classification of Disaster-Related Tweets". In: *International conference on Innovative Engineering Technologies (ICIET 2014)*. Dec. 2014.

[20] Abhinav Kumar et al. "A deep multi-modal neural network for informative Twitter content classification during emergencies". In: *Annals of Operations Research* (Jan. 2020). DOI: 10.1007/s10479-020-03514-x.

[21] Simon O'Keefe and Mohammed Alrashdi. "Deep Learning and Word Embeddings for Tweet Classification for Crisis Response". In: *National Computing Colleges Conference (NC3)*. Abha, Saudi Arabia, Oct. 2018.

[22] Pradip Bhere et al. "Classifying Informatory Tweets during Disaster Using Deep Learning". In: *ITM Web of Conferences* 32 (Jan. 2020), p. 03025. DOI: 10.1051/itmconf/20203203025.

[23] Sreenivasulu Madichetty and Sridevi Muthukumarasamy. "Detection of situational information from Twitter during disaster using deep learning models". In: *Sādhanā* 45 (Dec. 2020). DOI: 10.1007/s12046-020-01504-0.

[24] Sreenivasulu Madichetty. "A stacked convolutional neural network for detecting the resource tweets during a disaster". In: *Multimedia Tools and Applications* (Sept. 2020).

[25] Venkata Kishore Neppalli, Cornelia Caragea, and Doina Caragea. "Deep Neural Networks versus Naive Bayes Classifiers for Identifying Informative Tweets during Disasters". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management, Rochester, NY, USA, May 20-23, 2018*. Ed. by Kees Boersma and Brian M. Tomaszewski. ISCRAM Association, 2018. URL: http://idl.iscram.org/files/venkatakishoreneppalli/2018/1589%5C_VenkataKishoreNeppalli%5C_etal2018.pdf.

[26] Md. Yasin Kabir and Sanjay Madria. "A Deep Learning Approach for Tweet Classification and Rescue Scheduling for Effective Disaster Management". In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '19. Chicago, IL, USA: Association for Computing Machinery, 2019, pp. 269–278. ISBN: 9781450369091. DOI: 10.1145/3347146.3359097. URL: https://doi.org/10.1145/3347146.3359097.

[27] Abhinav Kumar, Jyoti Prakash Singh, and Sunil Saumya. "A Comparative Analysis of Machine Learning Techniques for Disaster-Related Tweet Classification". In: *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)*. 2019, pp. 222–227. DOI: 10.1109/R10-HTC47129.2019.9042443.

[28] Dat Nguyen et al. "Rapid Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks". In: (Aug. 2016).

[29] Dat Tien Nguyen et al. "Robust classification of crisis-related data on social networks using convolutional neural networks". English (US). In: *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*. Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017. AAAI press, 2017, pp. 632–635.

[30] Sreenivasulu Madichetty and M Sridevi. "Detecting Informative Tweets during Disaster using Deep Neural Networks". In: *2019 11th International Conference on Communication Systems Networks (COMSNETS)*. 2019, pp. 709–713. DOI: 10.1109/COMSNETS.2019.8711095.

[31] Ashwin Devaraj, Dhiraj Murthy, and Aman Dontula. "Machine-learning methods for identifying social media-based requests for urgent help during hurricanes". In: *International Journal of Disaster Risk Reduction* 51 (2020), p. 101757. ISSN: 2212-4209. DOI: https://doi.org/10.1016/j.ijdrr.2020.101757. URL: https://www.sciencedirect.com/science/article/pii/S2212420920312590.

[32] Hamada M. Zahera et al. "Fine-tuned BERT Model for Multi-Label Tweets Classification". In: *TREC*. 2019.

[33] Warih Maharani. "Sentiment Analysis during Jakarta Flood for Emergency Responses and Situational Awareness in Disaster Management using BERT". In: *2020 8th International Conference on Information and Communication Technology (ICoICT)*. 2020, pp. 1–5. DOI: 10.1109/ICoICT49345.2020.9166407.

[34] Pallavi Jain, Robert Ross, and Bianca Schoen-Phelan. "Estimating Distributed Representation Performance in Disaster-Related Social Media Classification". In: *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2019, pp. 723–727. DOI: 10.1145/3341161.3343680.

[35] Mohamed Barbouch, Frank Takes, and Suzan Verberne. "Combining Language Models and Network Features for Relevance-Based Tweet Classification". In: Oct. 2020, pp. 15–27. ISBN: 978-3-030-60974-0. DOI: 10.1007/978-3-030-60975-7_2.

[36] Abdul Hameed Azeemi and Adeel Waheed. "COVID-19 Tweets Analysis through Transformer Language Models". In: (Feb. 2021). arXiv: 2103.00199 [cs.CL].

[37] Hansi Hettiarachchi and Tharindu Ranasinghe. "InfoMiner at WNUT-2020 Task 2: Transformer-based Covid-19 Informative Tweet Extraction". In: *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 359–365. DOI: 10.18653/v1/2020.wnut-1.49. URL: https://www.aclweb.org/anthology/2020.wnut-1.49.

[38] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. ISBN: 9780262035613. URL: http://www.deeplearningbook.org.

[39] Aston Zhang et al. *Dive into Deep Learning*. 2020. URL: https://d2l.ai.

[40] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 2020. URL: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.

[41] Dat Tien Nguyen et al. *Rapid Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks*. Tech. rep. Qatar Computing Research Institute, Aug. 2016. URL: https://arxiv.org/pdf/1608.03902.pdf.

[42] Zichao Yang et al. "Hierarchical Attention Networks for Document Classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1480–1489. DOI: 10.18653/v1/N16-1174. URL: https://www.aclweb.org/anthology/N16-1174.

[43] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

[44] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[45] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].

[46] Pengcheng He et al. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. arXiv: 2006.03654 [cs.CL].

[47] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[48] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

[49] Kevin Clark et al. "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=r1xMH1BtvB.

[50] Firoj Alam, Shafiq Joty, and Muhammad Imran. "Domain Adaptation with Adversarial Training and Graph Embeddings". In: 2018.

[51] Hongmin Li. "Comparison of Word Embeddings and Sentence Encodings as Generalized Representations for Crisis Tweet Classification Tasks". In: *Proceedings of ISCRAM*. 2018.

[52] Abhinav Kumar and Jyoti Prakash Singh. "Location reference identification from tweets during emergencies: A deep learning approach". In: *International Journal of Disaster Risk Reduction* 33 (2019), pp. 365–375. ISSN: 2212-4209. DOI: https://doi.org/10.1016/j.ijdrr.2018.10.021. URL: https://www.sciencedirect.com/science/article/pii/S2212420918307799.

[53] Xukun Li and Doina Caragea. "Improving Disaster-related Tweet Classification with a Multimodal Approach". In: *17th International Conference on Information Systems for Crisis Response and Management*. Blacksburg, VA (USA): Virginia Tech, 2020, pp. 893–902. ISBN: 2411-3465.

[54] Guandan Chen, Qingchao Kong, and Wenji Mao. "An attention-based neural popularity prediction model for social media events". In: *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2017, pp. 161–163. DOI: 10.1109/ISI.2017.8004898.