ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Επεξηγηματική Βαθιά Μάθηση σε Εφαρμογές Ταξινόμησης Κειμένου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γαλάνη Νικόλαου

Επιβλέπων : κ. Ευστάθιος Σταματάτος

Εξεταστική επιτροπή : Καβαλλιεράτου Εργίνα, Κωστούλας Θεόδωρος

Σάμος, [Μάιος 2023]

Αυτή η σελίδα είναι σκόπιμα λευκή.

Πρόλογος

Αρχικά θα ήθελα να ευχαριστήσω τον κ. Ευστάθιο Σταματάτο για την επιλογή του τόσο ενδιαφέροντος θέματος αλλά και για την καθοδήγηση του. Ακόμη, θα ήθελα να ευχαριστήσω κάθε καθηγητή και καθηγήτρια του τμήματος του Μ.Π.Ε.Σ για όλα όσα έχουν προσφέρει.

Αυτή η σελίδα είναι σκόπιμα λευκή.

Πίνακας Περιεχομένων

Λίστα Εικόνων

Λίστα Πινάκων

Συντομογραφίες

AA	Authorship Attribution
AP	Attentive Pooling
BERT	Bidirectional Encoder Representations from Transformers
biLM	Bidirectional Language Modeling
CNN	Convolutional Neural Networks
CoVe	Context Vectors
DAN	Deep Average Network
DL	Deep Learning
DCNN	Dynamic Convolutional Neural Networks
DSCNN	Dependency Sensitive CNN
DMN	Dynamic Memory Network
DNN	Deep Neural Networks
ELMo	Embeddings from Language Models
GAN	Generative Adversarial Networks
GNN	Graph Neural Networks
GCN	Graph Convolutional Networks
LM	Language Modeling
LIME	local interpretable model-agnostic explanations
LSTM	Long-Sort Term Memory
MT-LSTM	Multi-Timescale Long Short-Term Memory Neural Network
MLP	Multi-Layer Perceptron
NLI	Natural language inference
NLP	Natural Language Processing
NLU	Natural Language Understanding
NSE	Neural Semantic Encoder
NSP	Next Sentence Prediction

PLM	Permuted Language Modeling
PLM	Pre-trained Language Models
PTM	Pre-Trained Models
QA	Question Answering
RTE	Recognizing Textual Entailment
RNN	Recurrent Neural Networks
SLP	Single Layer Perceptron
SOM	Self Organizing Maps
SOP	Sentence Order Prediction
SNLI	Stanford Natural Language Inference Corpus
SQuAD	Stanford Question Answering Dataset
SST	Stanford Sentiment Treebank
TC	Text Classification
ΤΝΔ	Τεχνητά Νευρωνικά Δίκτυα

Περίληψη

Η παρούσα εργασία διερευνά την ενσωμάτωση των σύγχρονων τεχνικών του artificial intelligence (AI), συγκεκριμένα της βαθιάς μάθησης και των νευρωνικών δικτύων, με μοντέλα ταξινόμησης κειμένων αλλά και εξηγήσιμα μοντέλα. Το επίκεντρο αυτής της έρευνας είναι η ανάπτυξη ενός μοντέλου που όχι μόνο επιτυγχάνει υψηλή ακρίβεια σε εργασίες ταξινόμησης κειμένου αλλά και παρέχει ερμηνεύσιμες επεξηγήσεις για τις προβλέψεις του. Η προτεινόμενη προσέγγιση κάνει χρήση ενός προ-εκπαιδευμένου μοντέλου βαθιάς μάθησης, το BERT, το οποίο έχει ρυθμιστεί λεπτομερός (fine-tuned) σε ένα μεγάλο σύνολο δεδομένων ταξινόμησης κειμένου. Στη συνέχεια θα ενσωματώσουμε δύο επεξηγήσιμα μοντέλα TN, το LIME και το SHAP, στο λεπτομερώς ρυθμισμένο μοντέλο BERT για να παρέχουν επεξηγήσεις για τις προβλέψεις του. Ακόμη θα διεξαχθούν πειράματα εστιάζοντας στην σύγκριση των δύο αυτών μοντέλων επεξηγηματικότητας και στην εύρεση του καταλληλότερου. Τα αποτελέσματα καταδεικνύουν ότι οι προτεινόμενες προσεγγίσεις παρέχουν ερμηνεύσιμες εξηγήσεις για τις προβλέψεις τους, καθιστώντας τες πολλά υποσχόμενες λύσεις για προβλήματα ταξινόμησης κειμένου στον πραγματικό κόσμο.

Λέξεις κλειδιά : *τεχνητή νοημοσύνη, βαθιά μάθηση, νευρωνικά δίκτυα, επεξηγήσιμη τεχνητή νοημοσύνη, ταξινόμηση κειμένου, υψηλή ακρίβεια, BERT, ερμηνεύσιμες επεξηγήσεις, προ-εκπαιδευμένο μοντέλο, LIME,SHAP.*

Abstract

This paper explores the integration of modern artificial intelligence (AI) techniques, specifically deep learning, and neural networks, with text classification models and explanatory models. The focus of this research is to develop a model that not only achieves high accuracy in text classification tasks but also provides interpretable explanations for its predictions. The proposed approach makes use of a pre-trained deep learning model, BERT, which has been fine-tuned on a large text classification dataset. We then incorporate two explanatory AI models, LIME and SHAP, into the fine-tuned BERT model to provide explanations for its predictions. Furthermore, we will conduct experiments focusing on comparing these two explicability models and finding the most suitable one. The results demonstrate that the proposed approaches provide interpretable explanations for their predictions, making them promising solutions for real-world text classification problems.

Keywords: *artificial intelligence, deep learning, neural networks, explainable artificial intelligence, text classification, high accuracy, BERT, interpretable explanations, pre-trained model, LIME,SHAP.*

1 *Εισαγωγή*

Αποκαλούμε τους εαυτούς μας Homo sapiens (σοφούς ανθρώπους) επειδή η νοημοσύνη είναι το πολυτιμότερο χαρακτηριστικό μας. Εδώ και χιλιάδες χρόνια, προσπαθούμε να εξηγήσουμε τον τρόπο που λειτουργεί ο ανθρώπινος νους, δηλαδή πώς αντιλαμβανόμαστε, κατανοούμε, προβλέπουμε και αλληλεπιδρούμε με έναν κόσμο που είναι πολύ μεγαλύτερος και πολύ πιο περίπλοκος από εμάς. Ο τομέας της τεχνητής νοημοσύνης, πηγαίνει ακόμη πιο πέρα, προσπαθεί όχι μόνο να κατανοήσει αλλά και να κατασκευάσει ευφυείς οντότητες. Η τεχνητή νοημοσύνη είναι ένας από τους νεότερους τομείς της επιστήμης και της μηχανικής. Οι εργασίες ξεκίνησαν σοβαρά αμέσως μετά τον Δεύτερο Παγκόσμιο Πόλεμο, και το ίδιο το όνομα επινοήθηκε το 1956. Η τεχνητή νοημοσύνη περιλαμβάνει σήμερα μια τεράστια ποικιλία από επιμέρους πεδία, που κυμαίνονται από τα γενικά (μάθηση και αντίληψη) έως τα ειδικά, όπως η απόδειξη μαθηματικών θεωρημάτων ,το σκάκι, η συγγραφή ποίησης, η οδήγηση ενός αυτοκινήτου σε έναν πολυσύχναστο δρόμο και η διάγνωση ασθενειών. Ακόμη, μεγάλη τεχνολογική άνθηση παρατηρείται και στον τομέα της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing (NLP)).

Η Επεξεργασία Φυσικής Γλώσσας ασχολείται με το πώς οι υπολογιστές μπορούν να αναγνωρίζουν, να κατανοούν και να παράγουν φυσική γλώσσα. Οι ερευνητές της NLP μελετούν την ανθρώπινη γλώσσα και προσπαθούν να αναπτύξουν τεχνικές και εργαλεία για την επεξεργασία της φυσικής γλώσσας από τους υπολογιστές, προκειμένου να επιτευχθούν επιθυμητά αποτελέσματα. Για την επίτευξη αυτού του στόχου, η NLP χρησιμοποιεί πολλούς διαφορετικούς κλάδους, όπως οι επιστήμες των υπολογιστών, η γλωσσολογία, η ψυχολογία, τα μαθηματικά και η τεχνητή νοημοσύνη. Οι εφαρμογές της NLP περιλαμβάνουν ορισμένα πεδία μελετών, όπως η μηχανική μετάφραση, η επεξεργασία κειμένου σε φυσική γλώσσα και η περίληψη, οι διεπαφές χρήστη (user interfaces), τα αποτελέσματα αναζήτησης (search results), η αναγνώριση συγγραφέα (author attribution), η πρόβλεψη κειμένου (predictive text), η μετάφραση γλώσσας (language translation), η ταξινόμηση κειμένων (text classification), η αναγνώριση ομιλίας, η τεχνητή νοημοσύνη, τα έμπειρα συστήματα, και πολλά άλλα. Στόχος της εργασίας είναι να μελετηθούν τεχνικές που εστιάζουν στην επεξήγηση αλγορίθμων, καθώς και να γίνει χρήση διαφόρων τεχνικών βαθιάς μάθησης.

1.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη αποτελεί έναν πολύ μεγάλο και πολυσύνθετο τομέα, που περιλαμβάνει διάφορες υποκατηγορίες, όπως η μηχανική μάθηση, η επεξεργασία φυσικής γλώσσας, οι αυτόνομοι πράκτορες, η ρομποτική, η αναγνώριση φωνής και εικόνας κ.ά. Σκοπός της τεχνητής νοημοσύνης είναι να αναπτύξει υπολογιστικά συστήματα που μπορούν να επιλύουν προβλήματα και να προσαρμόζονται σε νέες καταστάσεις με τρόπο παρόμοιο με την ανθρώπινη σκέψη και λήψη αποφάσεων.

Η τεχνητή νοημοσύνη είναι ένας τομέας που συνδυάζει πολλαπλές επιστήμες, όπως η πληροφορική, η ψυχολογία, η φιλοσοφία, η νευρολογία, η γλωσσολογία και η επιστήμη μηχανικών, με στόχο να δημιουργήσει ευφυή συμπεριφορά σε μηχανές ή υπολογιστές. Η συμβολική τεχνητή νοημοσύνη χρησιμοποιεί σύμβολα και λογικούς κανόνες υψηλού επιπέδου για να εξομοιώσει την ανθρώπινη νοημοσύνη, ενώ η υποσυμβολική (sub-symbolic) τεχνητή νοημοσύνη χρησιμοποιεί αριθμητικά μοντέλα και αυτοοργανωτικές δομές για να παραμετροποιήσει τη συμπεριφορά των μηχανών. Αυτή η προσέγγιση επιτρέπει την προσομοίωση της εξέλιξης των ειδών και της λειτουργίας του εγκεφάλου και επιτρέπει εφαρμογή στατιστικών μεθοδολογιών σε προβλήματα τεχνητής νοημοσύνης.

Η τεχνητή νοημοσύνη έχει εφαρμογές σε διάφορους τομείς και μπορεί να υλοποιηθεί μέσω διαφόρων τρόπων. Κάποιες από τις εφαρμογές της τεχνητής νοημοσύνης περιλαμβάνουν:

- **Λογισμικά**: Εικονικοί βοηθοί (όπως οι chatbots), λογισμικό ανάλυσης εικόνας (που χρησιμοποιείται για αναγνώριση προτύπων και αυτόματη ταξινόμηση εικόνων), μηχανές αναζήτησης (που χρησιμοποιούν αλγόριθμους μάθησης μηχανής για να παράγουν ακριβέστερα αποτελέσματα), συστήματα αναγνώρισης προσώπου και ομιλίας.

- **Ενσωματωμένη τεχνητή νοημοσύνη**: Αυτονόμων ρομπότ (που μπορούν να εκτελούν διάφορες εργασίες χωρίς την ανθρώπινη παρέμβαση), αυτόνομα αυτοκίνητα (που μπορούν να αναγνωρίζουν τα σήματα κυκλοφορίας και να αποφεύγουν τα εμπόδια), τηλεκατευθυνόμενα αεροσκάφη (drones, που μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, όπως για την επιτήρηση και την αναζήτηση και διάσωση), Internet of Things.

Τα τελευταία χρόνια παρατηρούμε ότι ολοένα και περισσότερο η Τεχνητή Νοημοσύνη εισάγεται στην καθημερινότητα μας, όπως για παράδειγμα

στις διαδικτυακές μας αγορές, στις αυτόματες μεταφράσεις, στον τομέα της αυτοκίνησης και της κυβερνοασφάλειας. Ένα ακόμη αξιοσημείωτο παράδειγμα είναι η χρήση της τεχνητής νοημοσύνης κατά του COVID-19, όπως για παράδειγμα στις συσκευές θερμικής απεικόνισης εξ αποστάσεως.

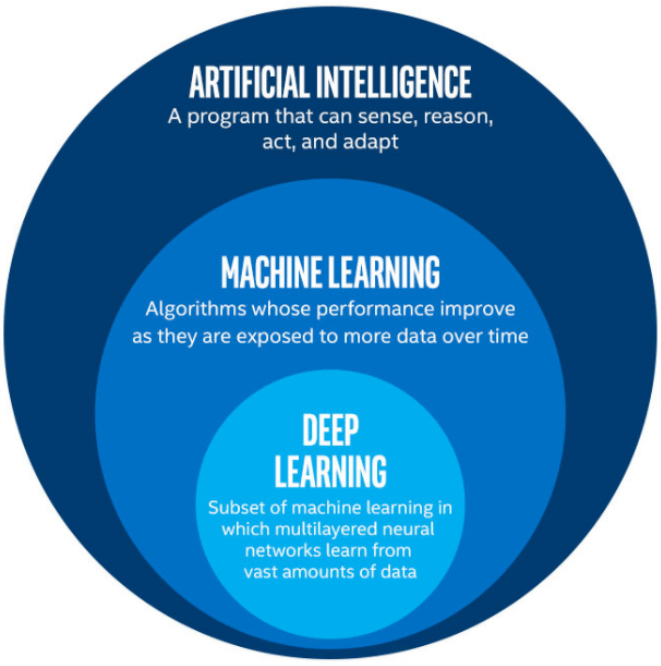
Η παρακάτω εργασία εστιάζει στον τομέα της μηχανικής μάθησης και ειδικότερα στην βαθιά μάθηση, που είναι υποπεδία της τεχνητής νοημοσύνης.

- Μηχανική μάθηση

Η μηχανική μάθηση είναι ένα υποπεδίο της τεχνητής νοημοσύνης, το οποίο επικεντρώνεται στη δημιουργία αλγορίθμων που εκπαιδεύονται από τα δεδομένα που τους δίνονται και λαμβάνουν αποφάσεις με βάση τα μοτίβα που παρατηρούνται στα δεδομένα αυτά. Αυτά τα έξυπνα συστήματα θα απαιτούν ανθρώπινη παρέμβαση όταν η απόφαση που λαμβάνεται είναι λανθασμένη ή ανεπιθύμητη.

- Βαθιά μάθηση

Το Deep Learning είναι ένα περαιτέρω υποσύνολο της μηχανικής μάθησης, που χρησιμοποιεί ένα τεχνητό νευρωνικό δίκτυο για να επεξεργάζεται δεδομένα μέσω διαφόρων επιπέδων αλγορίθμων και να καταλήγει σε μια ακριβή απόφαση χωρίς ανθρώπινη παρέμβαση.



Εικόνα 1: Artificial Intelligence, Machine Learning, Deep Learning

1.2 Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι ένα υποπεδίο της Επιστήμης των Υπολογιστών, το οποίο προέκυψε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Σκοπός της είναι να αναπτύξει αλγόριθμους που μπορούν να μαθαίνουν από δεδομένα και να προβλέπουν σχετικά με αυτά. Αυτοί οι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από δεδομένα και μπορούν να προβλέπουν ή να εξάγουν αποφάσεις βασιζόμενοι στα δεδομένα.

Η Μηχανική Μάθηση συνδέεται στενά με την υπολογιστική στατιστική, η οποία επικεντρώνεται στη χρήση των υπολογιστών για την πρόβλεψη και την ανάλυση δεδομένων. Επιπλέον, η Μηχανική Μάθηση συνδέεται με τη μαθηματική βελτιστοποίηση, παρέχοντας μεθόδους και θεωρία για την αναζήτηση της βέλτιστης λύσης σε προβλήματα πρόβλεψης και αναγνώρισης μοτίβων.

Η Μηχανική Μάθηση χρησιμοποιείται σε πολλές υπολογιστικές εργασίες, όπου ο σχεδιασμός και ο προγραμματισμός των αλγορίθμων είναι αδύνατος. Παραδείγματα εφαρμογών της μηχανικής μάθησης περιλαμβάνουν τα φίλτρα αντι-ανεπιθύμητων μηνυμάτων (spam filtering), την αυτόματη αναγνώριση χαρακτήρων (OCR), τις μηχανές αναζήτησης και την υπολογιστική όραση. Συχνά η μηχανική μάθηση συγχέεται με την εξόρυξη δεδομένων, όπου η τελευταία επικεντρώνεται περισσότερο στην εξερευνητική ανάλυση των δεδομένων, γνωστή και ως μη επιβλεπόμενη μάθηση.

Η μηχανική μάθηση αποτελείται από τρεις βασικές κατηγορίες εργασιών: την επίβλεψη (supervised learning), την ανεπίβλεπτη (unsupervised learning) και την ενισχυτική μάθηση (reinforcement learning).

- **Supervised Learning:** είναι μια κατηγορία μηχανικής μάθησης που χρησιμοποιεί labeled δεδομένα για να εκπαιδεύσει το μοντέλο και να προβλέψει την ετικέτα σε νέα αδιευκρίνιστα δεδομένα.

  _____

- **Unsupervised Learning**: ο αλγόριθμος εκπαιδεύεται χωρίς ετικέτες δεδομένων και έχει ως στόχο την εύρεση μοτίβων και την ομαδοποίηση των δεδομένων.

  _____

- Η **Ενισχυτική μάθηση (reinforcement learning):** είναι μια κατηγορία τεχνητής νοημοσύνης που αφορά τη διαδικασία εκμάθησης ενός αλγορίθμου ή ενός συστήματος να προσαρμόζεται σε ένα περιβάλλον. Σε αυτήν τη μέθοδο μάθησης, το σύστημα μάθησης αναπτύσσεται ώστε να

μπορεί να λαμβάνει αποφάσεις βάσει της επίδοσής του στο περιβάλλον στο οποίο βρίσκεται.

1.2.1 Τεχνητά νευρωνικά δίκτυα

Το **Νευρωνικό Δίκτυο** είναι ένας αλγόριθμος εκμάθησης τεχνητών νευρώνων, που βασίζεται στη δομή και λειτουργία των βιολογικών νευρωνικών δικτύων. Αποτελείται από μια ομάδα εσωτερικά διασυνδεδεμένων νευρώνων, οι οποίοι επεξεργάζονται την πληροφορία και επικοινωνούν μεταξύ τους.

Χρησιμοποιείται συνήθως για τη μοντελοποίηση σύνθετων σχέσεων μεταξύ εισόδου και εξόδου, την ανακάλυψη προτύπων στα δεδομένα, και τον εντοπισμό στατιστικής δομής σε μια άγνωστη κατανομή πιθανοτήτων. Αποτελεί ένα εργαλείο μη γραμμικής στατιστικής μοντελοποίησης δεδομένων.

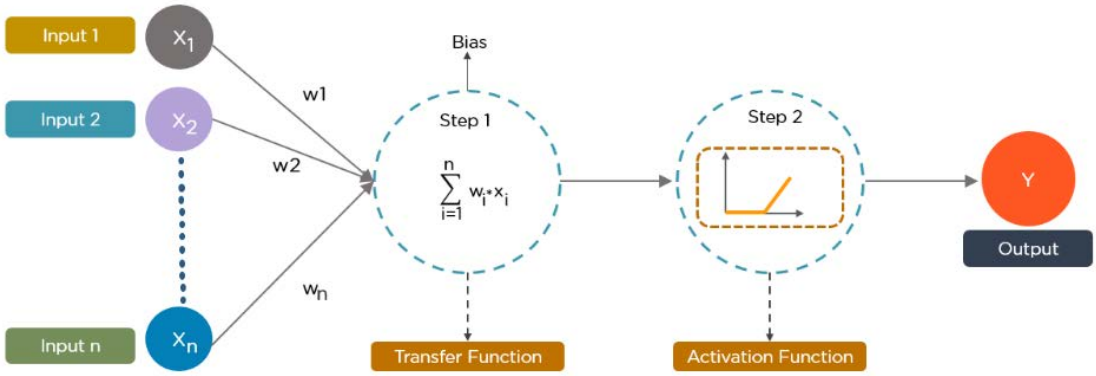
Το νευρωνικό δίκτυο είναι μια τεχνητή νευρωνική δομή που αποτελείται από τεχνητούς νευρώνες, γνωστούς και ως κόμβους, που οργανώνονται σε τρία επίπεδα:

• Το **επίπεδο εισόδου**, όπου οι εισαγόμενες δεδομένες τίθενται σε επαφή με το δίκτυο

• Τα **κρυφά επίπεδα**, όπου οι κόμβοι επεξεργάζονται τις εισαγόμενες πληροφορίες και δημιουργούν νέες παραστάσεις

• Το **επίπεδο εξόδου**, όπου οι παραγόμενες πληροφορίες δίνουν το αποτέλεσμα του δικτύου.

Αυτό το μοντέλο είναι εμπνευσμένο από τη βιολογική δομή του ανθρώπινου εγκεφάλου, όπου οι νευρώνες συνδέονται μεταξύ τους σε διαφορετικά επίπεδα επεξεργασίας πληροφοριών. Η χρήση αυτής της δομής στα τεχνητά νευρωνικά δίκτυα επιτρέπει την ανάλυση και την πρόβλεψη πολύπλοκων σχέσεων μεταξύ δεδομένων, όπως αναγνώριση εικόνων ή φωνητικών εντολών.



Εικόνα 2 : Νευρωνικό δίκτυο

Τα δεδομένα παρέχουν σε κάθε κόμβο πληροφορίες με τη μορφή εισόδων. Ο κόμβος πολλαπλασιάζει τις εισόδους με τυχαία βάρη, τις υπολογίζει και προσθέτει μια bias, ένα φαινόμενο που εμφανίζεται όταν ένας αλγόριθμος παράγει αποτελέσματα που είναι συστηματικά προκατειλημμένα λόγω λανθασμένων παραδοχών στη διαδικασία μηχανικής μάθησης. Οι μη γραμμικές συναρτήσεις, που είναι γνωστές και ως συναρτήσεις ενεργοποίησης, χρησιμοποιούνται για να αποφασίσουν ποιοι νευρώνες θα ανταποκριθούν και θα εκπέμψουν σήματα στο δίκτυο νευρώνων.

1.3 Κίνητρο

Το κίνητρο για τη συγγραφή αυτής της εργασίας προήλθε κυρίως από τη ραγδαία πρόοδο της Τεχνητής Νοημοσύνης και τον αυξανόμενο αντίκτυπό της σε διάφορους κλάδους, την ανάγκη για ερμηνεύσιμα μοντέλα Τεχνητής Νοημοσύνης, τη σημασία της ταξινόμησης κειμένου στην Επεξεργασία Φυσικής Γλώσσας και τα πιθανά οφέλη από το συνδυασμό αυτών των τεχνολογιών και τεχνικών. Η ενσωμάτωση της βαθιάς μάθησης και του Explainable AI στην ταξινόμηση κειμένου μπορεί να οδηγήσει σε πιο ακριβή και ερμηνεύσιμα μοντέλα, τα οποία μπορούν να είναι ιδιαίτερα χρήσιμα σε εφαρμογές του πραγματικού κόσμου, όπου η διαφάνεια και η εμπιστοσύνη στις προβλέψεις του μοντέλου είναι κρίσιμες. Το κίνητρο της εργασίας είναι να προσεγγίσει αυτούς τους ταχέως εξελισσόμενους τομείς και μέσω της εφαρμογής εξηγήσιμων αλγορίθμων να βγάλουμε κάποια συμπεράσματα και να κατανοήσουμε τον τρόπο λειτουργίας τους.

1.4 Στόχος της εργασίας

Βασικός στόχος της εργασίας είναι να αναπτύξουμε και να κατανοήσουμε με παραδείγματα κάποιες σημαντικές έννοιες στον τομέα της μηχανικής μάθησης, όπως είναι η βαθιά μάθηση, τα νευρωνικά δίκτυα, αλλά και διάφοροι αλγόριθμοι που χρησιμοποιούνται για την ταξινόμηση κειμένων.

Ακόμη, θα εμβαθύνουμε στον τομέα της ταξινόμησης κειμένου και, μέσω της χρήσης του αλγορίθμου BERT, θα διαμορφώσουμε κατάλληλα ένα μοντέλο

το οποίο θα εκπαιδευτεί, έτσι ώστε, αφότου του αναθέσουμε ένα κείμενο, να μπορεί να αναγνωρίζει την κατηγορία στην οποία ανήκει μέσω του αλγορίθμου.

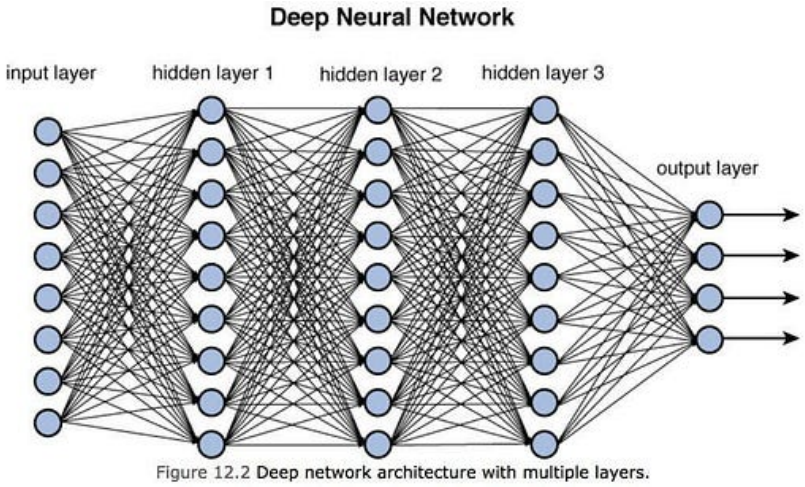
Επιπρόσθετα όσον αφορά την επεξηγηματικότατα του αλγορίθμου, θα τονίσουμε με χρωματική διαφοροποίηση τις λέξεις οι οποίες έπαιξαν θετικό ρόλο στην εύρεση της κατηγορίας με την μεγαλύτερη πιθανότητα, αλλά και το Prediction probability κάθε κατηγορίας, μέσω του LIME, ενός επεξηγηματικού μοντέλου, από τις οποίες ο αλγόριθμος επέφερε το συγκεκριμένο αποτέλεσμα. Επιπρόσθετα θα γίνουν συγκρίσεις με τα αποτελέσματα ενός άλλου επεξηγηματικού μοντέλου, το SHAP, έτσι ώστε να μπορέσουμε να βρούμε την καλύτερη ακρίβεια για το πρόβλημά μας.

2 *Βαθιά Μάθηση (Deep Learning) και προ εκπαιδευμένα μοντέλα (BERT)*

Η βαθιά μάθηση έχει αναδειχθεί ως μια ισχυρή τεχνική στον τομέα της τεχνητής νοημοσύνης, η οποία επιτρέπει στις μηχανές να μαθαίνουν πολύπλοκα μοτίβα και να κάνουν ακριβείς προβλέψεις. Πρόκειται για ένα υποσύνολο της μηχανικής μάθησης που χρησιμοποιεί τεχνητά νευρωνικά δίκτυα, εμπνευσμένα από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου, για την επεξεργασία και την ανάλυση μεγάλου όγκου δεδομένων. Η βαθιά μάθηση έχει επιδείξει αξιοσημείωτη επιτυχία σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένης της αναγνώρισης εικόνας, της αναγνώρισης ομιλίας, της επεξεργασίας φυσικής γλώσσας, ακόμη και του παιχνιδιού.

Ένα από τα κύρια πλεονεκτήματα της βαθιάς μάθησης είναι η ικανότητά της να μαθαίνει αυτόματα ιεραρχικές αναπαραστάσεις δεδομένων, όπου χαρακτηριστικά χαμηλότερου επιπέδου συνδυάζονται για να σχηματίσουν πιο σύνθετα χαρακτηριστικά υψηλότερου επιπέδου. Αυτό επιτρέπει στα μοντέλα βαθιάς μάθησης να συλλαμβάνουν λεπτά μοτίβα στα δεδομένα και να κάνουν ακριβείς προβλέψεις με μεγάλη ακρίβεια. Επιπλέον, τα μοντέλα βαθιάς μάθησης μπορούν να εκπαιδευτούν χρησιμοποιώντας μεγάλες ποσότητες δεδομένων, γεγονός που τα καθιστά ιδιαίτερα κατάλληλα για εφαρμογές όπου τα δεδομένα είναι άφθονα.

Παρά την επιτυχία της, η βαθιά μάθηση εξακολουθεί να είναι ένας τομέας ενεργού έρευνας και πολλές προκλήσεις παραμένουν. Αυτές περιλαμβάνουν τη βελτίωση της αποτελεσματικότητας της εκπαίδευσης βαθιών μοντέλων, τη μείωση της υπερπροσαρμογής (overfitting) και την ανάπτυξη τεχνικών για την ερμηνευσιμότητα και την επεξηγηματικότητα. Παρ' όλα αυτά, η βαθιά μάθηση έχει ήδη επηρεάσει βαθιά πολλούς τομείς της επιστήμης και της τεχνολογίας και είναι έτοιμη να συνεχίσει να διαμορφώνει το μέλλον της τεχνητής νοημοσύνης τα επόμενα χρόνια.



Εικόνα 3 :Deep Neural Network

Πηγή: https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964

Μια από τις βασικότερες χρήσεις της είναι για την ταξινόμηση κειμένου με τη δημιουργία μοντέλων που μπορούν να μάθουν να ταξινομούν αυτόματα κείμενο με βάση το περιεχόμενό του. Στην ταξινόμηση κειμένου, ο στόχος είναι η απόδοση μιας ή περισσότερων ετικετών (label) σε ένα δεδομένο κείμενο εισόδου (input) με βάση το περιεχόμενό του. Για παράδειγμα, μπορεί να θέλουμε να ταξινομήσουμε τις κριτικές πελατών ως θετικές ή αρνητικές ή να ταξινομήσουμε τα άρθρα ειδήσεων σε διάφορες κατηγορίες, όπως αθλητικά, πολιτική ή ψυχαγωγία.

Τα μοντέλα βαθιάς μάθησης, όπως τα νευρωνικά δίκτυα, μπορούν να χρησιμοποιηθούν για να μάθουν πώς να ταξινομούν αυτόματα κείμενο αναλύοντας μοτίβα σε μεγάλες ποσότητες δεδομένων εκπαίδευσης. Μια συνήθης προσέγγιση είναι η χρήση ενός τύπου νευρωνικού δικτύου που ονομάζεται "νευρωνικό δίκτυο συνελίξεων" (CNN) ή "επαναλαμβανόμενο νευρωνικό δίκτυο" (RNN), τα οποία έχουν σχεδιαστεί για να εργάζονται με διαδοχικά δεδομένα όπως το κείμενο.

Σε έναν τυπικό text classification pipeline που χρησιμοποιεί βαθιά μάθηση, το κείμενο εισόδου υφίσταται πρώτα προεπεξεργασία για τη μετατροπή του σε μια αριθμητική αναπαράσταση, όπως ένα διάνυσμα ή ένας πίνακας. Αυτό συνήθως γίνεται με την αναπαράσταση κάθε λέξης ή υπολέξης στο κείμενο ως διάνυσμα χρησιμοποιώντας τεχνικές όπως οι "word embeddings" ή οι "subword embeddings". Τα διανύσματα που προκύπτουν τροφοδοτούνται στη συνέχεια σε ένα μοντέλο βαθιάς μάθησης, το οποίο μαθαίνει να αντιστοιχίζει τα διανύσματα εισόδου στις επιθυμητές ετικέτες εξόδου χρησιμοποιώντας μια σειρά μαθηματικών πράξεων.

Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο βελτιστοποιείται ώστε να ελαχιστοποιεί μια συνάρτηση απωλειών που μετρά τη διαφορά μεταξύ των

προβλεπόμενων ετικετών και των πραγματικών ετικετών. Στη συνέχεια, το μοντέλο αξιολογείται σε ένα κρατημένο σύνολο δοκιμών για τη μέτρηση της ακρίβειας και της απόδοσης της γενίκευσής του.

Συνολικά, η βαθιά μάθηση έχει αποδειχθεί ότι είναι πολύ αποτελεσματική για εργασίες ταξινόμησης κειμένου, επιτυγχάνοντας κορυφαίες επιδόσεις σε πολλά σύνολα δεδομένων αναφοράς. Ωστόσο, απαιτεί μεγάλες ποσότητες σχολιασμένων δεδομένων εκπαίδευσης και μπορεί να είναι υπολογιστικά δαπανηρή η εκπαίδευση και η ανάπτυξή της.

Εκτός από τα οφέλη της χρήσης της βαθιάς μάθησης για την ταξινόμηση κειμένου, τα τελευταία χρόνια έχει σημειωθεί σημαντική πρόοδος στην ανάπτυξη προ-εκπαιδευμένων μοντέλων που μπορούν να ρυθμιστούν λεπτομερώς (fine-tuned) για συγκεκριμένες εργασίες. Αυτά τα μοντέλα συνήθως εκπαιδεύονται σε μεγάλες ποσότητες δεδομένων χωρίς ετικέτες με τεχνικές μάθησης χωρίς επίβλεψη και μπορούν να μάθουν αναπαραστάσεις κειμένου που είναι χρήσιμες για ένα ευρύ φάσμα μεταγενέστερων εργασιών.

Με τη λεπτομερή ρύθμιση αυτών των προ-εκπαιδευμένων μοντέλων σε μια συγκεκριμένη εργασία ταξινόμησης κειμένου, μπορούμε να επιτύχουμε κορυφαίες επιδόσεις με πολύ λιγότερα επισημασμένα δεδομένα και χρόνο εκπαίδευσης από ό,τι θα απαιτούνταν για την εκπαίδευση ενός μοντέλου από την αρχή. Στις επόμενες παραγράφους, θα διερευνήσουμε τους διαφορετικούς τύπους προ-εκπαιδευμένων μοντέλων που είναι διαθέσιμα για την ταξινόμηση κειμένου και τα πλεονεκτήματα και τους περιορισμούς τους.

2.1 Βασικές μέθοδοι Deep Learning

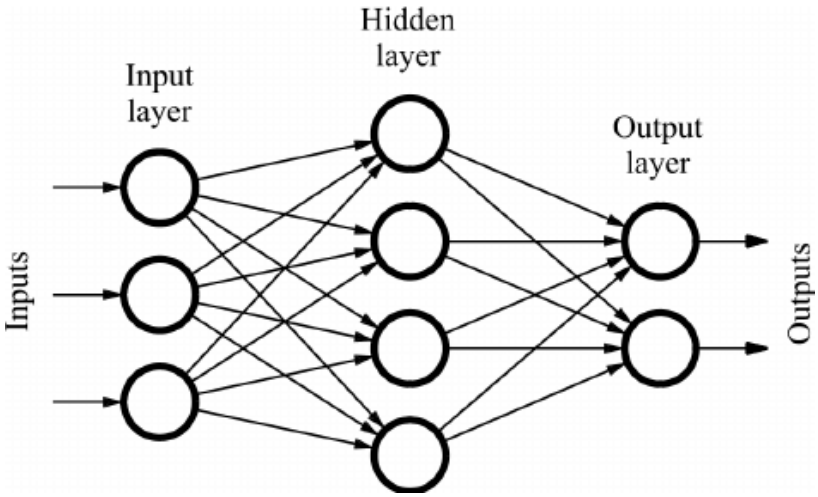
Η βαθιά μάθηση είναι ένα υποπεδίο της μηχανικής μάθησης που χρησιμοποιεί νευρωνικά δίκτυα με πολλά επίπεδα για να μαθαίνει από δεδομένα. Οι βασικές μέθοδοι της βαθιάς μάθησης περιλαμβάνουν:

2.1.1 Feedforward Neural Networks

Τα feedforward νευρωνικά δίκτυά είναι ένας τύπος τεχνητού νευρωνικού δικτύου όπου η πληροφορία ρέει μόνο προς μία κατεύθυνση, από το στρώμα εισόδου στο στρώμα εξόδου. Το δίκτυο αποτελείται από πολλαπλά στρώματα διασυνδεδεμένων κόμβων ή νευρώνων, όπου κάθε νευρώνας λαμβάνει είσοδο από το προηγούμενο στρώμα και μεταβιβάζει την έξοδό του στο επόμενο στρώμα. Η έξοδος του τελευταίου στρώματος αντιπροσωπεύει την πρόβλεψη του δικτύου για μια δεδομένη είσοδο.

Οι νευρώνες σε κάθε στρώμα συνδέονται συνήθως με όλους τους νευρώνες στο προηγούμενο στρώμα, αλλά όχι με άλλους νευρώνες στο ίδιο στρώμα ή με νευρώνες σε μελλοντικά στρώματα. Αυτό βοηθά στην αποφυγή βρόχων ανατροφοδότησης (feedback loops), οι οποίοι μπορεί να προκαλέσουν αστάθεια στο δίκτυο.

Το νευρωνικό δίκτυο τροφοδότησης χρησιμοποιείται συνήθως για εργασίες όπως η ταξινόμηση και η παλινδρόμηση. Το δίκτυο εκπαιδεύεται χρησιμοποιώντας έναν αλγόριθμο που ονομάζεται backpropagation, όπου τα βάρη των συνδέσεων μεταξύ των νευρώνων προσαρμόζονται με βάση τη διαφορά μεταξύ της προβλεπόμενης εξόδου και της επιθυμητής εξόδου. Η διαδικασία αυτή επαναλαμβάνεται για πολλές επαναλήψεις έως ότου η έξοδος του δικτύου είναι αρκετά ακριβής για την επιθυμητή εργασία.



Εικόνα 4 :Feed-Forward Networks

Πηγή: https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network

2.1.1.1 Perceptron

Το Perceptron είναι ένας τύπος τεχνητού νευρωνικού δικτύου που ακολουθεί την αρχιτεκτονική feed-forward. Ουσιαστικά το στρώμα εισόδου δέχεται ένα σύνολο χαρακτηριστικών εισόδου και το στρώμα εξόδου παράγει μια ενιαία έξοδο με βάση ένα σταθμισμένο άθροισμα αυτών των χαρακτηριστικών. Τα χαρακτηριστικά εισόδου πολλαπλασιάζονται με τα αντίστοιχα βάρη τους και οι προκύπτουσες τιμές αθροίζονται. Στη συνέχεια, ένας όρος μεροληψίας (bias term) προστίθεται σε αυτό το άθροισμα για την παραγωγή της τελικής εξόδου. Αυτή η έξοδος περνάει μέσα από μια συνάρτηση ενεργοποίησης, η οποία εισάγει μη γραμμικότητα στην έξοδο.

Η αρχιτεκτονική τροφοδότησης προς τα εμπρός του Perceptron το καθιστά κατάλληλο για την επίλυση προβλημάτων δυαδικής ταξινόμησης, όπου ο στόχος είναι να ταξινομηθούν τα δεδομένα εισόδου σε μία από δύο κλάσεις.

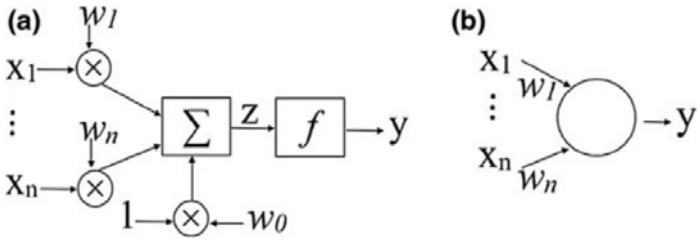
Αναλύοντας περισσότερο την λειτουργία του perceptron μπορούμε να διαπιστώσουμε ότι είναι ένα αρκετά απλό feed-forward δίκτυο. Χαρακτηρίζεται και ως δυαδικός ταξινομητής, διότι σε κάθε είσοδο x αντιστοιχεί μία τιμή εξόδου f(x), η οποία θα είναι είτε 1 είτε 0. Ύστερα, ταξινομείται το αποτέλεσμα είτε ως θετικό είτε ως αρνητικό, με 1 ή 0 αντίστοιχα, ανάλογα με τον αριθμό εξόδου της συνάρτησης. Το 0 ή 1 που έχει ως έξοδο η συνάρτηση f(x) χρησιμοποιείται για να ταξινομηθεί το αποτέλεσμα είτε σαν θετικό (1) είτε σαν αρνητικό (0) στιγμιότυπο.

Το perceptron ενός νευρώνα χαρακτηρίζεται ως το απλούστερο νευρωνικό δίκτυο καθώς έχει μόνο μία έξοδο με την οποία συνδέονται όλες οι είσοδοι. Με δεδομένο i = 0,1, ...,n όπου n ο αριθμός των εισόδων, οι ποσότητες $\{w_i\}$ είναι τα βάρη του νευρώνα. Οι είσοδοι $\{x_i\}$ αντιστοιχούν σε χαρακτηριστικά ή μεταβλητές και η έξοδος y στην προβλεπτική δυαδική κλάση τους [1].

Η Εικόνα 5 περιγράφει τα τρία βήματα που σχηματίζουν το μοντέλο perceptron.

Εικόνα 5: Perceptron steps: from left to right, weighting, sum and transfer steps

Στην Εικόνα 6 παρουσιάζεται η απλοποιημένη αναπαράστασή του. Το βήμα της στάθμισης περιλαμβάνει τον πολλαπλασιασμό της τιμής κάθε χαρακτηριστικού εισόδου με το βάρος του $\{x_i w_i\}$ και στο δεύτερο βήμα αθροίζονται $(x_0 w_0 + x_1 w_1 + + x_n w_n)$.



Εικόνα 6: Perceptron model, from left to right: a steps model. b Simplified model

Το τρίτο είναι το βήμα μεταφοράς, όπου μια συνάρτηση ενεργοποίησης f (που ονομάζεται επίσης συνάρτηση μεταφοράς) εφαρμόζεται στο άθροισμα παράγοντας μια έξοδο y που παρουσιάζεται ως εξής:

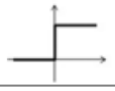
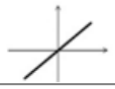
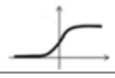
$$y = f(z) \; and \; z = \sum_{i=0}^{n} w_i x_i$$

Εικόνα 7: Output y

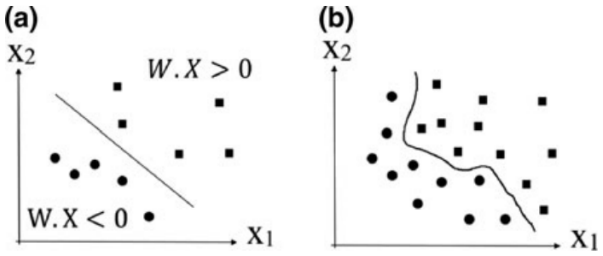
$x_0 = 1$, w_0 το threshold ή το bias και y η έξοδος.

Η συνάρτηση ενεργοποίησης παίρνει διάφορες μορφές. Οι κοινές τους συναρτήσεις παρατίθενται στον παρακάτω πίνακα.

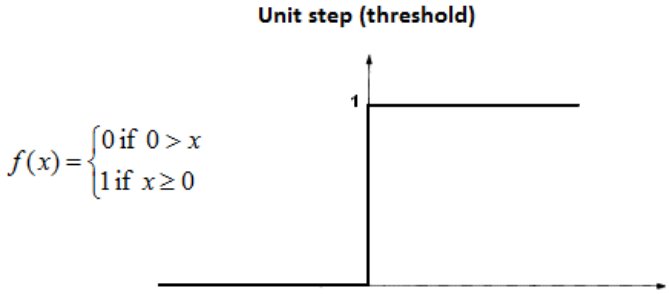
Ένα perceptron μπορεί να μάθει μόνο από γραμμικά διαχωρίσιμες συναρτήσεις. Η Εικόνα 9(α) δείχνει ένα παράδειγμα γραμμικής συνάρτησης που διαχωρίζει τα δεδομένα σε δύο κλάσεις. Σε δύο διαστάσεις με δύο χαρακτηριστικά, η συνάρτηση είναι μια γραμμή. Σε τρεις διαστάσεις με τρία χαρακτηριστικά, είναι ένα επίπεδο. Σε n διαστάσεις, είναι ένα υπερεπίπεδο με εξίσωση:

Activation function	Equation	2D graph
Unit step (Heaviside)	$f(z) = \begin{cases} 1 z \geq 0 \\ 0 z < 0 \end{cases}$	
Linear	$f(z) = z$	
Logistic (sigmoid)	$f(x) = \frac{1}{1+e^{-x}}$	

Εικόνα 8: Unit step, Linear, Logistic



Εικόνα 9: Input patterns, from left to right: a linearly separable, b nonlinearly separable



Εικόνα 10: Unit Step Activation Function

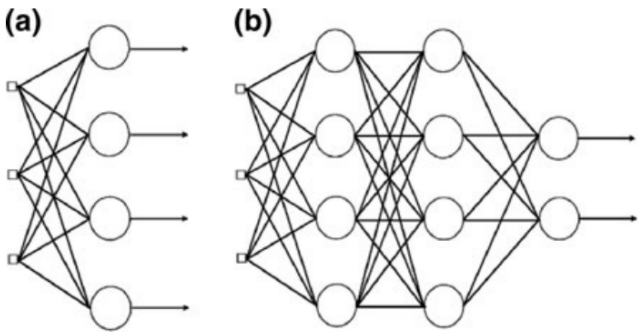
Πηγή : https://www.saedsayad.com/artificial_neural_network.htm

2.1.1.2 Multilayer Perceptron's (MLPs)

Ένα από τα δημοφιλή Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) είναι το Multi-Layer Perceptron (MLP). Πρόκειται για ένα ισχυρό εργαλείο μοντελοποίησης, το οποίο εφαρμόζει μια επιβλεπόμενη διαδικασία εκπαίδευσης χρησιμοποιώντας παραδείγματα δεδομένων με γνωστές εξόδους (Bishop 1995). Η διαδικασία αυτή δημιουργεί ένα μοντέλο μη γραμμικής συνάρτησης που επιτρέπει την πρόβλεψη δεδομένων εξόδου από δεδομένα εισόδου.

Η παράλληλη σύνδεση πολλών perceptrons δημιουργεί μια αρχιτεκτονική single layer perceptron (SLP), η οποία χρησιμοποιείται στην περίπτωση διαφόρων εξόδων. Στην Εικόνα 11α παρουσιάζεται ένα παράδειγμα με ένα στρώμα εισόδου και εξόδου που εξυπηρετεί σε μια περίπτωση γραμμικά διαχωρίσιμης πολλαπλής κατηγορίας. Το MLP και το SLP δεν επιλύουν το μη γραμμικά διαχωρίσιμο πρόβλημα.

Στην περίπτωση αυτή, μπορεί να βρεθεί λύση με την προσθήκη οποιουδήποτε αριθμού στρωμάτων σε διαδοχική διάταξη και τη δημιουργία μιας αρχιτεκτονικής MLP (Εικόνα 11β). Η έξοδος ενός στρώματος γίνεται η είσοδος του επόμενου και ούτω καθεξής. Το πρώτο και το τελευταίο στρώμα ονομάζονται στρώματα εισόδου και εξόδου αντίστοιχα, ενώ τα υπόλοιπα είναι τα κρυφά στρώματα του νευρωνικού δικτύου.



Εικόνα 11: Layer structure: a SLP with three inputs and four outputs. b MLP with three inputs, two hidden layers, and two outputs

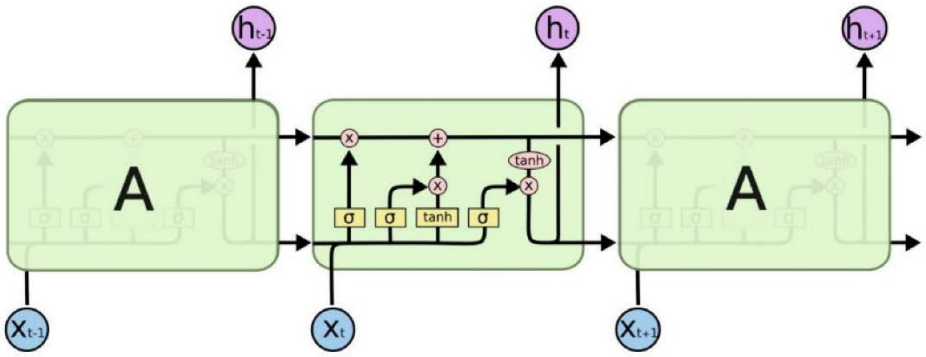
Το MLP είναι ένα πολυεπίπεδο νευρωνικό δίκτυο διάδοσης στο οποίο η πληροφορία ρέει μονόδρομα από το στρώμα εισόδου στο στρώμα εξόδου, περνώντας από τα κρυφά στρώματα (Bishop 1995). Κάθε σύνδεση μεταξύ των νευρώνων έχει το δικό της βάρος. Οι αντιληπτικοί νευρώνες για το ίδιο επίπεδο έχουν την ίδια συνάρτηση ενεργοποίησης. Γενικά, είναι σιγμοειδής για τα κρυφά στρώματα. Ανάλογα με την εφαρμογή, το στρώμα εξόδου μπορεί να είναι επίσης σιγμοειδής ή γραμμική συνάρτηση.

2.1.2 Recurrent Neural Networks (RNNs):

Τα Recurrent Neural Networks (RNN) είναι νευρωνικά δίκτυα που έχουν σχεδιαστεί για να επεξεργάζονται διαδοχικά δεδομένα. Διαθέτουν βρόχους που επιτρέπουν τη διατήρηση των πληροφοριών, πράγμα χρήσιμο για εργασίες όπως η μοντελοποίηση γλωσσών και η μετάφραση μηχανών. Στα RNNs, η έξοδος από ένα time step ανατροφοδοτείται ως είσοδος στο επόμενο time step, γεγονός που επιτρέπει στο δίκτυο να έχει μνήμη των προηγούμενων εισόδων. Αυτό καθιστά τα RNN ιδιαίτερα αποτελεσματικά για την επεξεργασία δεδομένων που έχουν μια χρονική συνιστώσα, όπως η ομιλία, η μουσική και το κείμενο. Τα RNNs έχουν πολλές παραλλαγές, συμπεριλαμβανομένων των δικτύων μακράς βραχυπρόθεσμης μνήμης (LSTM) και των δικτύων Gated Recurrent Unit (GRU), τα οποία αντιμετωπίζουν ορισμένους από τους περιορισμούς των παραδοσιακών RNNs, όπως το πρόβλημα της εξαφανιζόμενης κλίσης (vanishing gradient problem). Τα RNNs έχουν χρησιμοποιηθεί σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένης της επεξεργασίας φυσικής γλώσσας, της αναγνώρισης ομιλίας και της αναγνώρισης γραφής.

2.1.2.1 **Long Short-Term Memory Networks (LSTMs)**

Τα LSTM ή αλλιώς δίκτυα μακράς βραχυπρόθεσμης μνήμης, είναι ένας τύπος Αναδρομικού Νευρωνικού Δικτύου (RNN) που έχει σχεδιαστεί για να αντιμετωπίζει το πρόβλημα vanishing gradients στα παραδοσιακά RNN. Παρουσιάστηκαν το 1997 από τους Hochreiter και Schmidhuber και έκτοτε έχουν γίνει μια από τις πιο ευρέως χρησιμοποιούμενες αρχιτεκτονικές στη βαθιά μάθηση, ιδίως στον τομέα της επεξεργασίας φυσικής γλώσσας (NLP).

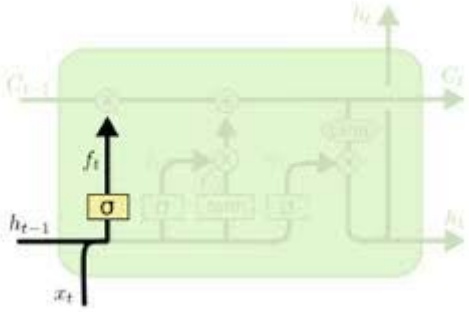


Εικόνα 12 : Εσωτερική δομή LSTM δικτύου

Πηγή: https://www.mdpi.com/2674-1032/1/2/14

Τα LSTM λειτουργούν χρησιμοποιώντας πύλες για τον έλεγχο της ροής των πληροφοριών μέσω του δικτύου, επιτρέποντάς του να μαθαίνει και να θυμάται μακροπρόθεσμες εξαρτήσεις σε διαδοχικά δεδομένα. Τα βήματα που εμπλέκονται στη λειτουργία ενός LSTM είναι τα εξής:

1. Επιλεκτική πύλη (forget gate): Καθορίζει την ποσότητα της νέας πληροφορίας που πρέπει να μεταβιβαστεί στο κελί μνήμης. Αυτό θα γίνει μέσω της χρήσης της σιγμοειδούς στρώσης (Forget Gate Layer). Ανάλογα με τις τιμές των h_{t-1} και x_t μας επιστρέφει ένα νούμερο μεταξύ 0 και 1 για κάθε νούμερο στη κατάσταση κελιού C_t

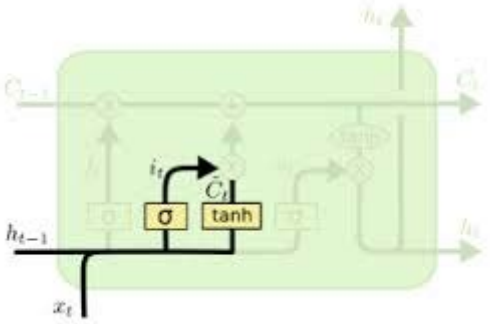


Εικόνα 13 : Επιλεκτική πύλη (forget gate) σε LSTM

Πηγή : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

2. Πύλη εισόδου (Input Gate): Ρυθμίζει πόση από την εισερχόμενη πληροφορία θα αφήσει να επηρεάσει την κατάσταση του κελιού μνήμης. Πιο

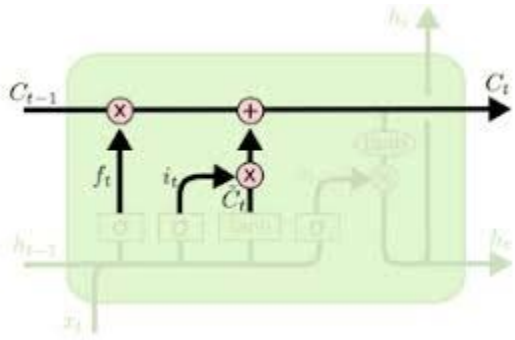
συγκεκριμένα, η πύλη εισόδου χρησιμοποιεί μια σιγμοειδή συνάρτηση ενεργοποίησης για να αποφασίσει ποια στοιχεία της εισόδου θα περάσουν στην κατάσταση του κελιού μνήμης.
Πιο συγκεκριμένα, η πύλη εισόδου λειτουργεί ως ένα φίλτρο που ελέγχει ποια πληροφορία εισέρχεται στο κελί μνήμης. Η σιγμοειδής συνάρτηση ενεργοποίησης της πύλης εισόδου είναι υπεύθυνη για τον υπολογισμό του "στρώματος φιλτραρίσματος", δηλαδή ποια στοιχεία της εισόδου θα περάσουν και ποια θα απορριφθούν. Η συνάρτηση αυτή δέχεται ως είσοδο τη συνολική είσοδο και εξάγει ένα διάνυσμα με τιμές μεταξύ 0 και 1, όπου μικρές τιμές σημαίνουν απόρριψη της πληροφορίας και μεγάλες τιμές σημαίνουν διέλευση της πληροφορίας στο επόμενο στάδιο.



Εικόνα 14 : Σιγμοειδές και εφαπτομενικό επίπεδο για παραγωγή υποψήφιων τιμών κελιού μνήμης

Πηγή: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

3. Κατάσταση κελιού: Η κατάσταση του κελιού C_t αποθηκεύει πληροφορίες για παρατεταμένο χρονικό διάστημα, επιτρέποντας στο δίκτυο να διατηρεί μια μόνιμη μνήμη. Ουσιαστικά από την κατάσταση C_{t-1} με δύο πολλαπλασιασμούς $C_t = f_t * C_{t-1} + i_t * C'_t$, θα έχουμε την νέα κατάσταση C_t

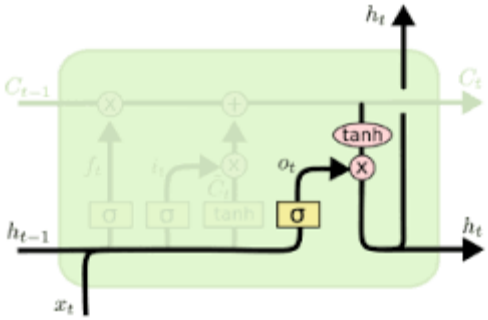


Εικόνα 15: Ενημέρωση νέας κατάστασης Ct

Πηγή : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

4. Πύλη εξόδου (Output Gate): Ελέγχει την ποσότητα της πληροφορίας που επιτρέπεται να περάσει από το κύτταρο μνήμης στην έξοδο. Αντίστοιχα

με τα προηγούμενα βήματα, πρώτα μέσω μιας σιγμοειδούς στρώσης βρίσκουμε τα στοιχεία κατάστασης του κελιού για την έξοδο και ύστερα πολλαπλασιάζεται με την κατάσταση του κελιού C_t αφού έχει περάσει πρώτα από μια συνάρτηση tanh.



Εικόνα 16 : Παραγωγή τιμής εξόδου LSTM

Πηγή : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

5. Κρυμμένη κατάσταση (hidden state): Τέλος έχουμε την έξοδο του LSTM (h_t), η οποία χρησιμοποιείται ως είσοδος για το επόμενο χρονικό βήμα.

Χρησιμοποιώντας αυτές τις πύλες, τα LSTM μπορούν να ξεπεράσουν το πρόβλημα της εξαφάνισης των κλίσεων (vanishing gradients) και να μάθουν αποτελεσματικά τις μακροπρόθεσμες εξαρτήσεις σε διαδοχικά δεδομένα. Έχουν χρησιμοποιηθεί με επιτυχία σε διάφορες εργασίες NLP, όπως η ταξινόμηση κειμένων, η ανάλυση συναισθήματος και η μηχανική μετάφραση.

2.1.3 Convolutional Neural Networks (CNNs):

Μία από τις βασικές διάφορες των RNNs με τα CNNs μοντέλα, είναι ότι τα RNNs εκπαιδεύονται να αναγνωρίζουν μοτίβα στο χρόνο, ενώ τα CNN μαθαίνουν να αναγνωρίζουν μοτίβα στο χώρο. Ακόμη, τα RNNs λειτουργούν καλά για τις εργασίες NLP, όπως POS tagging ή QA, όπου απαιτείται η κατανόηση της σημασιολογίας μεγάλης εμβέλειας, ενώ τα CNNs λειτουργούν καλά όταν είναι σημαντική η ανίχνευση τοπικών και αμετάβλητων ως προς τη θέση μοτίβων. Αυτά τα μοτίβα θα μπορούσαν να είναι φράσεις-κλειδιά που εκφράζουν ένα συγκεκριμένο συναίσθημα όπως "μου αρέσει" ή ένα θέμα όπως "απειλούμενα είδη". Έτσι, τα CNNs έχουν γίνει μια από τις πιο δημοφιλείς αρχιτεκτονικές μοντέλων για την ταξινόμηση κειμένων.

Ένα ακόμη χαρακτηριστικό των CNN είναι ότι αποτελείται 3 επίπεδα:

- επίπεδο συνέλιξης

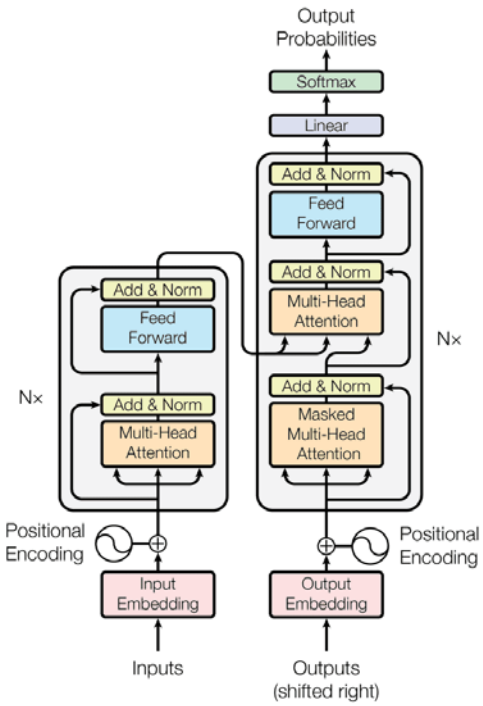
- επίπεδο συγκέντρωσης
- πλήρως συνδεδεμένο επίπεδο

Το επίπεδο συνέλιξης (convolution layer) παίζει βασικό ρόλο στο CNN αφού αποτελείται από μια στοίβα μαθηματικών πράξεων, όπως η συνέλιξη, ένας εξειδικευμένος τύπος γραμμικής λειτουργίας.

2.1.4 Transformers

Ένα από τα υπολογιστικά προβλήματα που αντιμετωπίζουν τα RNN είναι η διαδοχική επεξεργασία κειμένου. Παρόλο που τα CNN είναι λιγότερο διαδοχικά από τα RNN, το υπολογιστικό κόστος για την καταγραφή των σχέσεων μεταξύ των στοιχείων σε μια ακολουθία αυξάνεται επίσης με την αύξηση του μήκους της ακολουθείας, όπως και στα RNN. Οι μετασχηματιστές (transformers) ξεπερνούν αυτόν τον περιορισμό εφαρμόζοντας την self-attention για να υπολογίζουν παράλληλα για κάθε στοιχείο σε μια ακολουθία ή ένα έγγραφο ένα " attention score" για να μοντελοποιήσουν την επιρροή που έχει κάθε λέξη σε μια άλλη. Λόγω αυτού του χαρακτηριστικού, οι Transformers επιτρέπουν πολύ μεγαλύτερο παραλληλισμό από ότι τα CNN και τα RNN, γεγονός που καθιστά δυνατή την αποτελεσματική εκπαίδευση πολύ μεγάλων μοντέλων σε μεγάλες ποσότητες δεδομένων σε GPU.

Μπορούμε να περιγράψουμε μια συνάρτηση προσοχής ως ένα μηχανισμό που επιτρέπει σε μια μηχανή μάθησης να επιλέγει ποιες πληροφορίες πρέπει να επικεντρωθεί κατά την επεξεργασία μιας εισόδου. Αυτή η λειτουργία αντιστοιχεί στον συνδυασμό ενός ερωτήματος με ένα σύνολο ζευγών κλειδιών-τιμών, όπου το ερώτημα και τα κλειδιά είναι διανύσματα, και οι τιμές και η έξοδος είναι επίσης διανύσματα. Η έξοδος παράγεται από ένα σταθμισμένο άθροισμα των τιμών, όπου η συνάρτηση προσοχής υπολογίζει τα βάρη των τιμών με βάση το πόσο καλά ταιριάζουν με το ερώτημα και τα αντίστοιχα κλειδιά. Αυτό δίνει στη μηχανή μάθησης τη δυνατότητα να επικεντρωθεί στα σημαντικά στοιχεία της εισόδου και να αποφασίσει ποιες πληροφορίες πρέπει να χρησιμοποιηθούν για να λύσει μια δεδομένη εργασία. [2].



Εικόνα 17 : Transformer

Πηγή : https://machinelearningmastery.com/the-transformer-model/

Στην παραπάνω εικόνα, εμφανίζονται πλήρως συνδεδεμένα στρώματα τόσο στον κωδικοποιητή όσο και στον αποκωδικοποιητή. Αυτά τα στρώματα είναι συνδεδεμένα μεταξύ τους και κάθε νευρώνας σε ένα στρώμα είναι συνδεδεμένος με όλους τους νευρώνες στο προηγούμενο και στο επόμενο στρώμα. Αυτό επιτρέπει την ανταλλαγή πληροφοριών μεταξύ όλων των νευρώνων και συνεισφέρει στην ακρίβεια του μοντέλου.

Ο **Encoder** αναλαμβάνει τον ρόλο της επεξεργασίας της εισόδου. Πιο συγκεκριμένα, δέχεται μια ακολουθία και τη μετατρέπει σε μια αναπαράσταση που μπορεί να χρησιμοποιηθεί από τον decoder για τη δημιουργία της εξόδου.

Η διαδικασία του encoder αποτελείται από πολλαπλά layers, κάθε ένα από τα οποία περιλαμβάνει δύο βασικές λειτουργίες:

- self-attention
- feedforward.

Στο self-attention, κάθε layer αξιολογεί κάθε στοιχείο μιας ακολουθίας σε σχέση με τα υπόλοιπα στοιχεία και επιλέγει ποιες πληροφορίες είναι σημαντικές για κάθε στοιχείο. Αυτό οδηγεί σε μια νέα αναπαράσταση της ακολουθίας, όπου κάθε στοιχείο συσχετίζεται με σχετικές πληροφορίες από τα υπόλοιπα στοιχεία.

Στη συνέχεια, ένα feedforward layer επεξεργάζεται κάθε στοιχείο στη νέα αναπαράσταση ξεχωριστά και εξάγει χρήσιμες πληροφορίες για κάποιο άλλο task. Αυτά τα βήματα επαναλαμβάνονται πολλές φορές, ανάλογα με το μέγεθος του μοντέλου, για να παραχθεί η τελική αναπαράσταση της αρχικής ακολουθίας.

Κατά τη διάρκεια της κωδικοποίησης, οι γραμμικοί μετασχηματισμοί εφαρμόζονται σε κάθε στοιχείο της ακολουθίας εισόδου από τα ίδια στρώματα, αλλά οι παράμετροι βάρους και απόκλισης των στρωμάτων είναι διαφορετικές για κάθε στρώμα. Αυτό σημαίνει ότι κάθε στοιχείο της ακολουθίας εισόδου περνάει μέσα από μια σειρά από στρώματα που εφαρμόζουν τον ίδιο μετασχηματισμό, αλλά κάθε στρώμα έχει διαφορετικές παραμέτρους. Αυτό βοηθάει στη δημιουργία πολλαπλών αναπαραστάσεων των διαφορετικών πτυχών της εισόδου και στην ανάδειξη των σημαντικών χαρακτηριστικών της, που μετά μπορούν να χρησιμοποιηθούν από τον αποκωδικοποιητή για τη δημιουργία της εξόδου.

Μια σημαντική παρατήρηση που πρέπει να ληφθεί υπόψη, είναι ότι η αρχιτεκτονική του μετασχηματιστή δεν μπορεί εγγενώς να συλλάβει καμία πληροφορία σχετικά με τις σχετικές θέσεις των στοιχείων στην ακολουθία, δεδομένου ότι δεν κάνει χρήση της αναδρομής. Αυτή η πληροφορία πρέπει να εισαχθεί με την εισαγωγή κωδικοποιήσεων θέσης στις ενσωματώσεις εισόδου.

Ο **Decoder** του Transformer αναλαμβάνει να παράγει μία ακολουθία εξόδου από την προηγούμενη επεξεργασία της ακολουθίας εισόδου και το σύνολο των κρυφών προσαρμοστικών παραμέτρων του Encoder. Ο Decoder αποτελείται από παρόμοια συνελικτικά επίπεδα με τον Encoder, αλλά εναλλάσσονται με Multi-Head Attention επίπεδα που είναι υπεύθυνα για την επιλογή ποιόν στοιχείων από την είσοδο θα πρέπει να είναι πιο σημαντικές για την επεξεργασία τους.

Αρχικά, ο Decoder δέχεται μια μάσκα που αποτρέπει τον προσδιορισμό μελλοντικών στοιχείων κατά την πρόβλεψη της κάθε λέξης της ακολουθίας εξόδου. Στη συνέχεια, η ακολουθία εξόδου περνάει από το Embedding Layer του Decoder, όπου κάθε λέξη αντιστοιχίζεται σε ένα διάνυσμα στον ίδιο χώρο εμβέντινγκ όπως και στον Encoder.

Στη συνέχεια, η ακολουθία εξόδου περνάει από τα Multi-Head Attention επίπεδα, όπου η ακολουθία αυτή θεωρείται ως η ακολουθία εισόδου και οι εξόδοι του τελευταίου Encoder layer θεωρούνται ως τα κλειδιά και οι τιμές. Αυτό το βήμα βοηθάει τον Decoder να εστιάσει στα σημαντικότερα στοιχεία της ακολουθίας εισόδου.

Τέλος χρησιμοποιείτε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο τροφοδότησης προς τα εμπρός. Το δίκτυο αυτό λειτουργεί από κάθε επίπεδο του κωδικοποιητή ξεχωριστά και αναλαμβάνει την επεξεργασία της πληροφορίας που έχει εξαχθεί από τα προηγούμενα στρώματα. Η σημασία αυτού του στρώματος είναι ότι δίνει τη δυνατότητα στο Transformer να εκτελεί μη γραμμικές πράξεις και να αναλύει περίπλοκες συσχετίσεις ανάμεσα στα στοιχεία της ακολουθίας εισόδου.

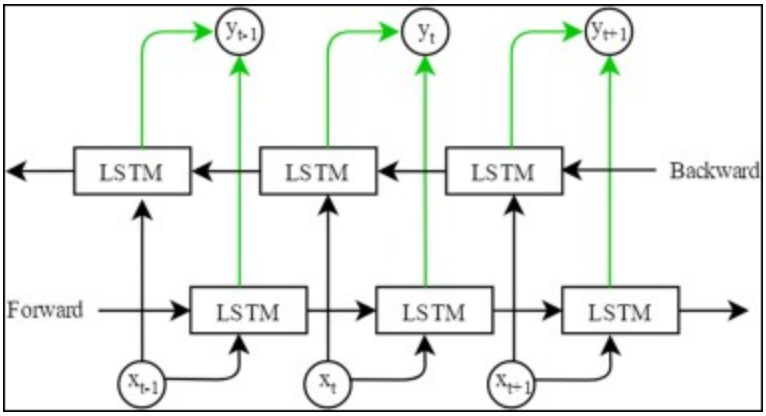
Επιπλέον, ακριβώς όπως και στον κωδικοποιητή, αυτά τα υποστρώματα στην πλευρά του αποκωδικοποιητή έχουν επίσης υπολειμματικές συνδέσεις γύρω τους και τα διαδέχεται ένα στρώμα κανονικοποίησης.

Γενικότερα, έχει υπάρξει αναβάθμιση των προ-εκπαιδευμένων γλωσσικών μοντέλων (PLM) μεγάλης κλίμακας που βασίζονται σε μετασχηματιστές. Σε σύγκριση με τα προηγούμενα μοντέλα ενσωμάτωσης με βάση το περιεχόμενο που βασίζονται σε CNN ή LSTM, τα PLM που βασίζονται σε Transformer χρησιμοποιούν πολύ βαθύτερες αρχιτεκτονικές δικτύων και προ-εκπαιδεύονται σε πολύ μεγαλύτερες ποσότητες σωμάτων κειμένου για να μάθουν αναπαραστάσεις κειμένου με βάση το περιεχόμενο προβλέποντας λέξεις που εξαρτώνται από το περιεχόμενό τους. Αυτά τα PLM είναι λεπτομερώς ρυθμισμένα με τη χρήση ειδικών ετικετών για κάθε εργασία και έχουν δημιουργήσει νέα δεδομένα σε πολλές μεταγενέστερες εργασίες NLP, συμπεριλαμβανομένου του TC. Αν και η προ-εκπαίδευση είναι unsupervised (ή self-supervised), το fine-tuning είναι supervised μάθηση.

2.1.4.1 Διαφορές Μεταξύ LSTM και Transformer

Η αρχιτεκτονική των transformer νευρωνικών δικτύων αρχικά δημιουργήθηκε για την επίλυση του προβλήματος της μετάφρασης (language translation), στην οποία μπορεί να ανταπεξέλθει με μεγάλη επιτυχία, σε αντίθεση με τα LSTM Networks.

Πριν την δημιουργία των transformers, χρησιμοποιούνταν τα LSTM νευρωνικά δίκτυα, όμως είχαν πολλά μειονεκτήματα. Αρχικά ήταν πολύ αργά στην εκμάθηση, διότι οι λέξεις επιλέγονταν σειριακά και παράγονταν σειριακά, με αποτέλεσμα το νευρωνικό δίκτυο να χρειάζεται παραπάνω χρόνο. Ακόμη, δεν είναι η κατάλληλη μέθοδος για να αποσπάσουμε με ακρίβεια την σημασιολογία των λέξεων, διότι μαθαίνουν από τα δεξιά προς τα αριστερά και από τα αριστερά προς τα δεξιά ξεχωριστά και ύστερα τα συνδέουν (όπως θα δείτε στο παρακάτω σχήμα),



Εικόνα 18: LSTM for language translation

σε αντίθεση με τον transformer ο οποίος μπορεί να μάθει λέξεις και από τις δύο κατευθύνσεις παράλληλα. Με αυτήν την μέθοδο καταφέρνει να είναι πιο γρήγορος, αλλά και πιο ακριβής στην σημασιολογική μετάφραση των λέξεων**.**

2.1.5 Capsule Neural Networks

Ο βασικός σκοπός των Capsule Neural networks είναι η αντιμετώπιση του προβλήματος της απώλειας πληροφορίας που προκαλείται από τις λειτουργίες συγκέντρωσης των CNNs. Τα CNNs ταξινομούν εικόνες ή κείμενα χρησιμοποιώντας διαδοχικά επίπεδα συνελίξεων και συνένωσης. Παρόλο που οι λειτουργίες συγκέντρωσης εντοπίζουν τα σημαντικά χαρακτηριστικά και μειώνουν την υπολογιστική πολυπλοκότητα των λειτουργιών συνέλιξης, χάνουν πληροφορίες σχετικά με τις χωρικές σχέσεις και είναι πιθανό να ταξινομήσουν εσφαλμένα τις οντότητες με βάση τον προσανατολισμό ή την αναλογία τους. Για να αντιμετωπιστούν τα προβλήματα της συγκέντρωσης, προτείνεται μια νέα προσέγγιση από τον Hinton, η οποία ονομάζεται «δίκτυα κάψουλας» (CapsNets). Μια κάψουλα είναι μια ομάδα νευρώνων των οποίων το διάνυσμα δραστηριότητας (activity vector) αντιπροσωπεύει διαφορετικά χαρακτηριστικά ενός συγκεκριμένου τύπου οντότητας, όπως ένα αντικείμενο ή ένα μέρος αντικειμένου.

Σε αντίθεση με το max pooling των CNNs, το οποίο επιλέγει κάποιες πληροφορίες και απορρίπτει τις υπόλοιπες, οι κάψουλες "δρομολογούν" κάθε κάψουλα από το κατώτερο στρώμα στην καλύτερη parent capsule στο ανώτερο στρώμα, χρησιμοποιώντας όλες τις πληροφορίες που είναι διαθέσιμες στο δίκτυο μέχρι το τελικό στρώμα για την ταξινόμηση. Η δρομολόγηση μπορεί να υλοποιηθεί με τη χρήση διαφορετικών αλγορίθμων, όπως η δυναμική routing-by-agreement.

2.1.6 *The attention mechanism*

Ο «μηχανισμός προσοχής» έχει ως βάση τον τρόπο με τον οποίο δίνουμε οπτική προσοχή στα διάφορα μέρη μιας εικόνας ή συσχετίζουμε λέξεις σε μια πρόταση. Ο μηχανισμός προσοχής γίνεται μια όλο και πιο δημοφιλής έννοια και ένα χρήσιμο εργαλείο στην ανάπτυξη μοντέλων DL για NLP. Με λίγα λόγια, η προσοχή στα γλωσσικά μοντέλα μπορεί να ερμηνευτεί ως ένα σημαντικό διάνυσμα. Προκειμένου να προβλέψουμε μια λέξη σε μια πρόταση, εκτιμούμε χρησιμοποιώντας το διάνυσμα προσοχής πόσο έντονα συσχετίζεται με άλλες λέξεις και παίρνουμε το άθροισμα των τιμών τους σταθμισμένο με το διάνυσμα προσοχής ως προσέγγιση του στόχου.

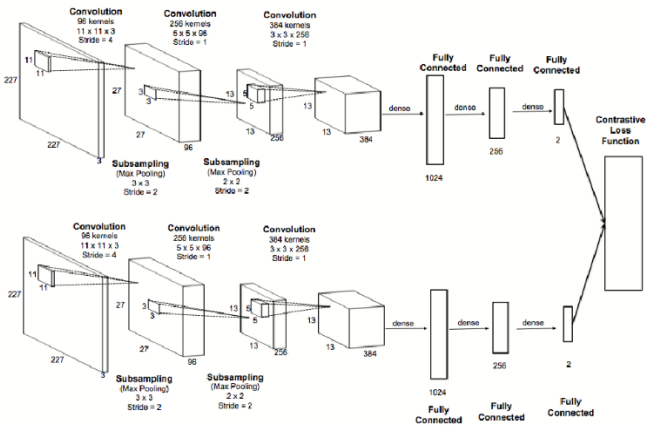
2.1.7 *The Memory-augmented networks*

Ενώ τα κρυφά διανύσματα που αποθηκεύονται από ένα μοντέλο προσοχής κατά την κωδικοποίηση μπορούν να θεωρηθούν ως καταχωρήσεις της εσωτερικής μνήμης του μοντέλου, τα δίκτυα με ενισχυμένη μνήμη συνδυάζουν τα νευρωνικά δίκτυα με μια μορφή εξωτερικής μνήμης, από την οποία το μοντέλο μπορεί να διαβάζει και να γράφει.

2.1.8 *Siamese Neural Networks*

Τα σιαμαία νευρωνικά δίκτυα (S2Nets) και οι παραλλαγές τους DNN, γνωστά ως Βαθιά Δομημένα Σημασιολογικά Μοντέλα (Deep Structured Semantic Models - DSSMs), έχουν σχεδιαστεί για την αντιστοίχιση κειμένου. Η εργασία αυτή είναι θεμελιώδης για πολλές εφαρμογές NLP, όπως η κατάταξη ερωτημάτων-κειμένων και η επιλογή απαντήσεων στην εξαγωγή QA. Αυτές οι εργασίες μπορούν να θεωρηθούν ως ειδικές περιπτώσεις του TC.

Στην Εικόνα 19 παρουσιάζεται η αρχιτεκτονική του σιαμαίου μοντέλου, όπου τα δύο δίκτυα χρησιμοποιούν το ίδιο μοντέλο LSTM.



Εικόνα 19: Siamese Neural Networks

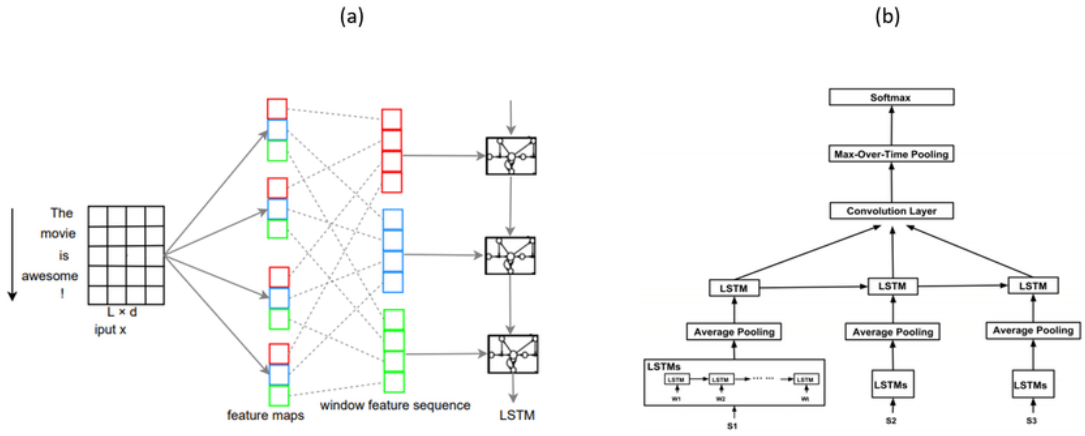
Πηγή: https://becominghuman.ai/siamese-networks-algorithm-applications-and-pytorch-implementation-4ffa3304c18

Είναι σημαντικό όχι μόνο η αρχιτεκτονική των υποδικτύων να είναι πανομοιότυπη, αλλά και οι εργασίες να είναι χωρισμένες ισάξια, έτσι ώστε το δίκτυο να αποκαλείται "σιαμαίο". Η βασική έννοια πίσω από τα σιαμαία δίκτυα είναι ότι μπορούν να μάθουν χρήσιμες πληροφορίες δεδομένων που μπορούν να χρησιμοποιηθούν περαιτέρω για τη σύγκριση μεταξύ των εισόδων των αντίστοιχων υποδικτύων. Με τον τρόπο αυτό, οι είσοδοι μπορούν να είναι οτιδήποτε, από αριθμητικά δεδομένα (σε αυτή την περίπτωση τα υποδίκτυα σχηματίζονται συνήθως από πλήρως συνδεδεμένα layers), δεδομένα εικόνας (με τα CNN ως υποδίκτυα), ή ακόμη και διαδοχικά δεδομένα όπως προτάσεις ή χρονικά σήματα (με τα RNN ως υποδίκτυα).

2.1.9 Hybrid models

Έχουν αναπτυχθεί πολλά υβριδικά μοντέλα που συνδυάζουν τις αρχιτεκτονικές LSTM και CNN για την καταγραφή των τοπικών και παγκόσμιων χαρακτηριστικών των προτάσεων και των εγγράφων.

Όπως απεικονίζεται στην Εικόνα 20 (α), το Convolutional LSTM (C-LSTM) χρησιμοποιεί ένα CNN για την εξαγωγή μιας ακολουθίας αναπαραστάσεων φράσεων υψηλότερου επιπέδου (n-gram), οι οποίες τροφοδοτούνται σε ένα δίκτυο LSTM για να ληφθεί η αναπαράσταση της πρότασης. Ύστερα, για την μοντελοποίηση του εγγράφου όπως απεικονίζεται στην Εικόνα 20 (β), το DSCNN είναι ένα ιεραρχικό μοντέλο, όπου το LSTM μαθαίνει τα διανύσματα προτάσεων, τα οποία τροφοδοτούνται στα στρώματα συνέλιξης και max pooling για τη δημιουργία της αναπαράστασης εγγράφου.



Εικόνα 20 : Αρχιτεκτονική του C-LSTM και DSCNN

Πηγή: https://www.researchgate.net/figure/a-The-architecture-of-C-LSTM-121-b-The-architecture-of-DSCNN-for-document-modeling_fig18_340523298

2.2 Προεκπαιδευμένα μοντέλα (Pre-trained models)

Με την ανάπτυξη της βαθιάς μάθησης, ο αριθμός των παραμέτρων των μοντέλων αυξήθηκε ραγδαία. Έχει γίνει απαραίτητη η χρήση μεγαλύτερου συνόλου δεδομένων για την πλήρη εκπαίδευση των παραμέτρων του μοντέλου και την αποφυγή της υπερπροσαρμογής (overfitting). Ωστόσο, η δημιουργία μεγάλης κλίμακας επισημασμένων συνόλων δεδομένων αποτελεί μεγάλη πρόκληση για τις περισσότερες εργασίες NLP λόγω του εξαιρετικά δαπανηρού κόστους σχολιασμού, ειδικά για εργασίες που σχετίζονται με το συντακτικό και τη σημασιολογία. Αντίθετα, τα μεγάλης κλίμακας μη επισημασμένα δεδομένα (non-labeled data) είναι σχετικά εύκολο να κατασκευαστούν. Για να αξιοποιήσουμε τα τεράστια μη επισημασμένα δεδομένα κειμένου, μπορούμε πρώτα να μάθουμε μια καλή αναπαράσταση από αυτά και στη συνέχεια να χρησιμοποιήσουμε αυτές τις αναπαραστάσεις για άλλες εργασίες. Πρόσφατες μελέτες έχουν καταδείξει σημαντική αύξηση της απόδοσης σε πολλές εργασίες NLP με τη βοήθεια της αναπαράστασης που εξάγεται από τα PTMs στα μεγάλης κλίμακας μη επισημασμένα δεδομένα (non-labeled data). Τα πλεονεκτήματα της προ-εκπαίδευσης μπορούν να συνοψιστούν ως εξής:

1. Η προ-εκπαίδευση στο τεράστιο σώμα κειμένων μπορεί να μάθει γενικές γλωσσικές αναπαραστάσεις και να βοηθήσει στις επόμενες εργασίες.
2. Η προ-εκπαίδευση παρέχει καλύτερη αρχικοποίηση του μοντέλου, η οποία συνήθως οδηγεί σε καλύτερη απόδοση γενίκευσης και επιταχύνει τη σύγκλιση στην εργασία-στόχο.
3. Η προ-εκπαίδευση μπορεί να θεωρηθεί ως ένα είδος κανονικοποίησης για την αποφυγή υπερβολικής προσαρμογής σε μικρής κλίμακας δεδομένα.

Πρόσφατα, σημαντική εργασία έδειξε ότι τα προ-εκπαιδευμένα μοντέλα (PTM), στο μεγάλο σώμα κειμένων μπορούν να μάθουν καθολικές γλωσσικές αναπαραστάσεις, οι οποίες είναι επωφελείς για μεταγενέστερες εργασίες NLP και μπορούν να αποφύγουν την ανάγκη εκπαίδευσης ενός νέου μοντέλου από την αρχή. Με την ανάπτυξη της υπολογιστικής ισχύος, την εμφάνιση των “deep” μοντέλων (π.χ. Transformer) και τη συνεχή βελτίωση των εκπαιδευτικών δεξιοτήτων, η αρχιτεκτονική των PTMs έχει εξελιχθεί από «απλή» σε «βαθιά».

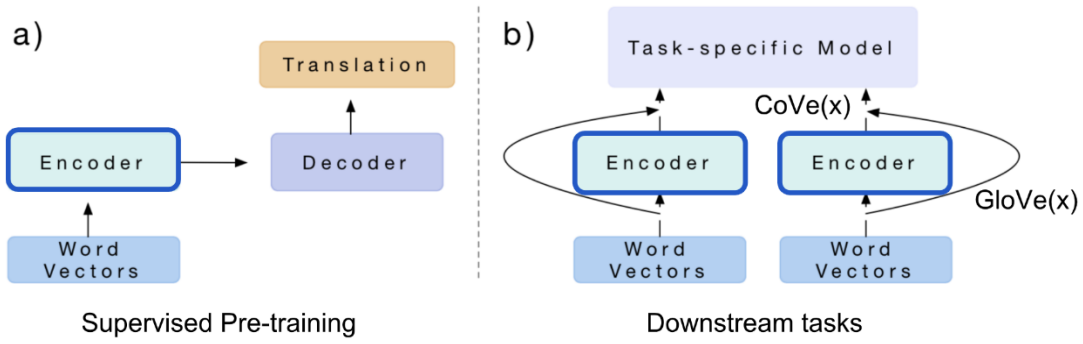
2.2.1 Pre-trained models (PTMs)(1st and 2nd Generation)

Τα PTM πρώτης γενιάς στοχεύουν στην εκμάθηση καλής ενσωμάτωσης λέξεων. Δεδομένου ότι τα ίδια αυτά μοντέλα δεν χρειάζονται πλέον από τις επόμενες εργασίες, είναι συνήθως πολύ απλά για λόγους υπολογιστικής αποτελεσματικότητας, όπως το Skip-Gram και το GloVe. Παρόλο που αυτές οι προ-εκπαιδευμένες ενσωματώσεις μπορούν να συλλάβουν τις σημασιολογικές

έννοιες των λέξεων, είναι χωρίς συμφραζόμενα και αποτυγχάνουν να συλλάβουν έννοιες υψηλότερου επιπέδου στα συμφραζόμενα, όπως η πολυσημασιολογική αποσαφήνιση, οι συντακτικές δομές, οι σημασιολογικοί ρόλοι, οι αναφορές. Τα PTM δεύτερης γενιάς επικεντρώνονται στην εκμάθηση ενσωμάτωσης λέξεων με βάση υπάρχοντα πλαίσια, όπως τα **CoVe** , **ELMo**, **OpenAI GPT** και **BERT**. Αυτοί οι εκπαιδευμένοι κωδικοποιητές εξακολουθούν να είναι απαραίτητοι για την αναπαράσταση λέξεων σε συμφραζόμενα από μεταγενέστερες εργασίες. Εκτός αυτού, προτείνονται επίσης διάφορες εργασίες προ-εκπαίδευσης για την εκμάθηση των PTMs για διαφορετικούς σκοπούς.

Η προ-εκπαίδευση ήταν πάντα μια αποτελεσματική στρατηγική για την εκμάθηση των παραμέτρων των deep νευρωνικών δικτύων, τα οποία στη συνέχεια ρυθμίζονται λεπτομερώς (fine-tuned) σε μεταγενέστερες εργασίες (downstream tasks).

2.2.1.1 CoVe Contextualized Word Vectors



Εικόνα 21 : CoVe

Πηγή : https://lilianweng.github.io/posts/2019-01-31-lm/

Στην παραπάνω εικόνα μπορούμε να παρατηρήσουμε ότι χωρίζεται σε δύο τμήματα. Στο πρώτο τμήμα (τμήμα a) εκπαιδεύουμε ένα αμφίδρομο LSTM δύο επιπέδων ως κωδικοποιητή ενός μοντέλου προσοχής ακολουθίας-σειράς για μηχανική μετάφραση και στο δεύτερο (τμήμα β) το χρησιμοποιούμε για να παρέχουμε πλαίσιο για άλλα μοντέλα NLP.

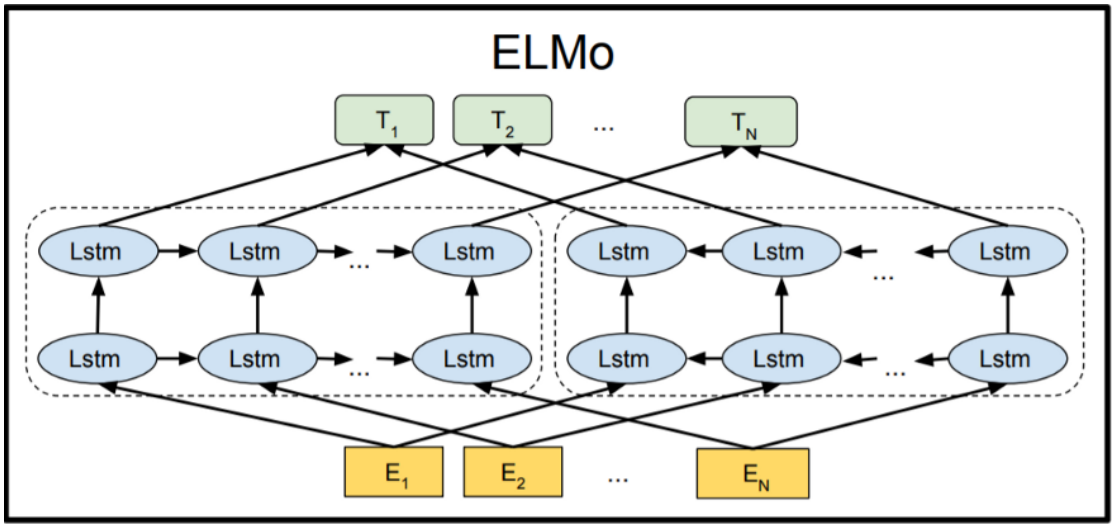
Προκειμένου να ελέγξουμε τη δυνατότητα μεταφοράς αυτών των encoders, αναπτύσσουμε μια κοινή αρχιτεκτονική για μια ποικιλία εργασιών ταξινόμησης και τροποποιούμε το Dynamic Coattention Network για την απάντηση ερωτήσεων. Προσθέτουμε τις εξόδους των MT-LSTM, τις οποίες ονομάζουμε context vectors (CoVe), στα διανύσματα λέξεων που χρησιμοποιούνται συνήθως ως είσοδοι σε αυτά τα μοντέλα. Αυτή η προσέγγιση βελτίωσε την απόδοση των μοντέλων για μεταγενέστερες εργασίες σε σχέση με

εκείνη των βασικών μοντέλων που χρησιμοποιούν μόνο προ-εκπαιδευμένα διανύσματα λέξεων. Για το Stanford Sentiment Treebank (SST) και το Stanford Natural Language Inference Corpus (SNLI), το CoVe ωθεί τις επιδόσεις του βασικού μας μοντέλου στο πιο υψηλό επίπεδο [3].

2.2.1.2 ELMo (Embeddings from Language Models)

Σε αντίθεση με τις περισσότερες ευρέως χρησιμοποιούμενες ενσωματώσεις λέξεων, οι αναπαραστάσεις λέξεων ELMo είναι συναρτήσεις ολόκληρης της πρότασης εισόδου. Υπολογίζονται πάνω σε biLM δύο επιπέδων με συμπτύξεις χαρακτήρων, ως γραμμική συνάρτηση των εσωτερικών καταστάσεων του δικτύου. Αυτή η ρύθμιση μας επιτρέπει να κάνουμε μάθηση με ημι-επίβλεψη, όπου το biLM προ-εκπαιδεύεται σε μεγάλη κλίμακα και ενσωματώνεται εύκολα σε ένα ευρύ φάσμα υφιστάμενων νευρωνικών αρχιτεκτονικών NLP [4].

Εκτεταμένα πειράματα αποδεικνύουν ότι οι αναπαραστάσεις ELMo λειτουργούν πολύ καλά στην πράξη. Το συγκεκριμένο task μπορεί εύκολα να προστεθεί σε υπάρχοντα μοντέλα για διάφορα προβλήματα γλωσσικής κατανόησης, όπως η συνεπαγωγή κειμένου, η απάντηση ερωτήσεων και η ανάλυση συναισθήματος. Η προσθήκη των αναπαραστάσεων ELMo από μόνη της βελτιώνει σημαντικά την κατάσταση της τεχνολογίας σε κάθε περίπτωση, συμπεριλαμβανομένης της μείωσης των σχετικών σφαλμάτων έως και 20%. Για εργασίες όπου είναι δυνατές οι άμεσες συγκρίσεις, το ELMo υπερτερεί έναντι του CoVe, το οποίο υπολογίζει αναπαραστάσεις με βάση το περιεχόμενο χρησιμοποιώντας έναν νευρωνικό κωδικοποιητή μηχανικής μετάφρασης. Τέλος, μια ανάλυση τόσο του ELMo όσο και του CoVe αποκαλύπτει ότι οι βαθιές αναπαραστάσεις υπερτερούν εκείνων που προέρχονται μόνο από το ανώτερο στρώμα ενός LSTM.



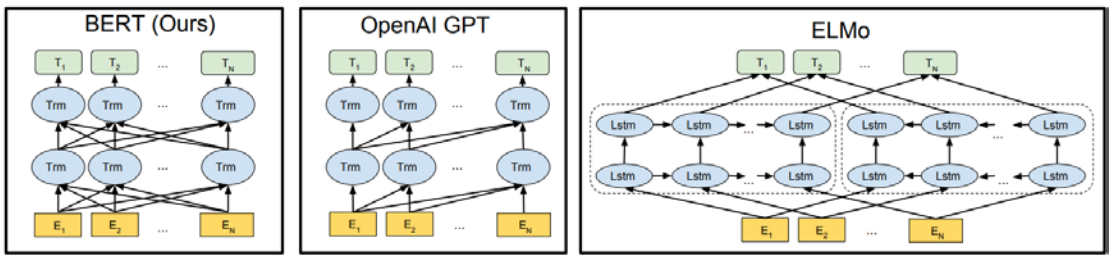
Εικόνα 22: ELMo

Πηγή: https://medium.com/saarthi-ai/elmo-for-contextual-word-embedding-for-text-classification-24c9693b0045

2.2.1.3 OpenAI GPT

Το GPT είναι ένα μοντέλο ενσωμάτωσης με βάση την θέση του, οπότε συνήθως συνιστάται να συμπληρώνει τις εισόδους στα δεξιά παρά στα αριστερά. Ακόμη έχει εκπαιδευτεί με στόχο το causal language modeling (CLM) και επομένως είναι ισχυρό στην πρόβλεψη του επόμενου token σε μια ακολουθία. Η αξιοποίηση αυτού του χαρακτηριστικού επιτρέπει στο GPT να παράγει ένα συντακτικά συνεκτικό κείμενο.

Συγκρίνοντας λοιπόν τα μοντέλα προ-εκπαίδευσης, διαπιστώνουμε ότι υπάρχουν αρκετές διαφορές στις αρχιτεκτονικές τους. Ο BERT χρησιμοποιεί έναν αμφίδρομο μετασχηματιστή. Το OpenAI GPT χρησιμοποιεί έναν μετασχηματιστή από αριστερά προς τα δεξιά. Το ELMo χρησιμοποιεί τη συνένωση ανεξάρτητα εκπαιδευμένων από τα αριστερά προς τα δεξιά, και από τα δεξιά προς τα αριστερά LSTM για τη δημιουργία χαρακτηριστικών για μεταγενέστερες εργασίες. Μεταξύ των τριών, μόνο οι αναπαραστάσεις BERT εξαρτώνται από κοινού από το αριστερό και το δεξί πλαίσιο σε όλα τα επίπεδα. Εκτός από τις διαφορές στην αρχιτεκτονική, η BERT και η OpenAI GPT είναι προσεγγίσεις fine tuning, ενώ η ELMo είναι μια προσέγγιση βασισμένη σε χαρακτηριστικά.



Εικόνα 23 : Διαφορές μεταξύ BERT, OpenAI GPT και ELMo

Πηγή: https://www.researchgate.net/figure/Comparison-of-BERT-OpenAI-GPT-and-ELMo-model-architectures-5_fig1_338931711

2.2.2 Pre-training Tasks

Οι εργασίες προ-εκπαίδευσης είναι ζωτικής σημασίας για την εκμάθηση της καθολικής αναπαράστασης της γλώσσας. Συνήθως, αυτές οι εργασίες προ-εκπαίδευσης πρέπει να είναι απαιτητικές και να έχουν σημαντικά δεδομένα εκπαίδευσης. Παρακάτω, συνοψίζουμε τις εργασίες προ-εκπαίδευσης σε τρεις κατηγορίες: supervised learning, unsupervised learning, and self-supervised learning [5].

1. **Supervised learning (SL)** είναι η εκμάθηση μιας συνάρτησης που αντιστοιχίζει μια είσοδο σε μια έξοδο με βάση δεδομένα εκπαίδευσης που αποτελούνται από ζεύγη εισόδου-εξόδου.
2. **Unsupervised learning (UL)** είναι η εύρεση κάποιας εγγενούς γνώσης από μη επισημασμένα δεδομένα, όπως ομάδες, πυκνότητες, λανθάνουσες αναπαραστάσεις.
3. **Self-Supervised learning (SSL)** είναι ένα μείγμα μάθησης με επίβλεψη και μάθησης χωρίς επίβλεψη. Το παράδειγμα μάθησης της SSL είναι εξ ολοκλήρου το ίδιο με αυτό της επιβλεπόμενης μάθησης, αλλά οι ετικέτες των δεδομένων εκπαίδευσης παράγονται αυτόματα. Η βασική ιδέα του SSL είναι να προβλέψει οποιοδήποτε μέρος της εισόδου από άλλα μέρη σε κάποια μορφή. Για παράδειγμα, το μοντέλο καλυμμένης γλώσσας (MLM) είναι μια αυτοεπιβλεπόμενη εργασία που προσπαθεί να προβλέψει τις καλυμμένες λέξεις σε μία πρόταση δοθέντων των υπόλοιπων λέξεων.

Σύμφωνα με τις παραπάνω τρείς κατηγορίες, έχουν δημιουργηθεί πολλά pre-training tasks τα οποία χρησιμοποιούνται ανάλογα με τις ανάγκες του κάθε μοντέλου. Παρακάτω αναλύονται τα πιο βασικά και ευρέως διαδεδομένα tasks για pre-training.

2.2.2.1 **Language Modeling(LM)**

Η πιο συνηθισμένη unsupervised εργασία στο NLP είναι η probabilistic language modeling (LM), η οποία είναι ένα κλασικό πρόβλημα πιθανολογικής

εκτίμησης πυκνότητας. Παρόλο που η LM είναι μια γενική έννοια, στην πράξη, η LM συχνά αναφέρεται ειδικότερα στην αυτόματη LM ή στη μονόδρομη LM.

2.2.2.2 **Masked Language Modeling (MLM)**

Βασικός στόχος του (MLM) ήταν να ξεπεραστεί το μειονέκτημα της τυπικής μονόδρομης LM. Πληροφοριακά, η MLM πρώτα αποκρύπτει ορισμένα tokens από τις προτάσεις εισόδου και στη συνέχεια εκπαιδεύει το μοντέλο έτσι ώστε να μπορεί να προβλέπει τα κρυπτογραφημένα tokens από τα υπόλοιπα tokens. Ωστόσο, αυτή η μέθοδος προ-εκπαίδευσης δημιουργεί μια αναντιστοιχία μεταξύ της φάσης προ-εκπαίδευσης και της φάσης τελειοποίησης, επειδή το εικονίδιο μάσκας δεν εμφανίζεται κατά τη διάρκεια της φάσης τελειοποίησης. Η επίλυση αυτού του ζητήματος και περαιτέρω πληροφορίες αναφέρονται στο κεφάλαιο 3.3.3.1.

2.2.2.3 **Permuted Language Modeling (PLM)**

Παρά την ευρεία χρήση της εργασίας MLM στην προ-εκπαίδευση, πολλοί ισχυρίστηκαν ότι ορισμένα ειδικά tokens που χρησιμοποιούνται στην προ-εκπαίδευση της MLM, όπως το [MASK], απουσιάζουν όταν το μοντέλο εφαρμόζεται σε επόμενες εργασίες, οδηγώντας σε ένα κενό μεταξύ προ-εκπαίδευσης και τελειοποίησης (fine-tuning). Για να ξεπεραστεί αυτό το ζήτημα, η Permuted Language Modeling (PLM) είναι ένας στόχος προ-εκπαίδευσης που αντικαθιστά το MLM. Εν ολίγοις, η PLM είναι ένα task γλωσσικής μοντελοποίησης σε μια τυχαία μετάθεση ακολουθιών εισόδου. Μια μετάθεση επιλέγεται τυχαία από όλες τις πιθανές μεταθέσεις. Στη συνέχεια, επιλέγονται ορισμένα από τα tokens της μετασχηματιζόμενης ακολουθίας ως στόχος και το μοντέλο εκπαιδεύεται για να προβλέψει αυτούς τους στόχους, ανάλογα με τα υπόλοιπα tokens και τις φυσικές θέσεις των στόχων. Αξίζει να σημειωθεί ότι αυτή η μετάλλαξη δεν επηρεάζει τις φυσικές θέσεις των ακολουθιών και καθορίζει μόνο τη σειρά των προβλέψεων των συμβόλων. Στην πράξη, μόνο τα τελευταία σημεία στις μετασχηματιζόμενες ακολουθίες προβλέπονται, λόγω της αργής σύγκλισης, και εισάγεται μια ειδική αυτοπροσοχή δύο ροών για αναπαραστάσεις με επίγνωση στόχων.

2.2.2.4 **Contrastive Learning (CTL)**

Η μέθοδος του Contrastive Learning βασίζεται ουσιαστικά στην υπόθεση ότι κάποια ζεύγη κειμένων είναι περισσότερο όμοια σημασιολογικά από ό,τι τα τυχαία επιλεγμένα κείμενα. Παρακάτω θα αναφέρουμε τρία από τα βασικότερα tasks που είναι βασισμένα σε CTL:

- **Deep InfoMax (DIM)** Το Deep InfoMax (DIM) προτάθηκε αρχικά για εικόνες, διότι βελτιώνει την ποιότητα της αναπαράστασης μεγιστοποιώντας την αμοιβαία πληροφορία μεταξύ μιας αναπαράστασης εικόνας και τοπικών περιοχών της εικόνας.

- **Next Sentence Prediction (NSP)** Τα σημεία στίξης είναι οι φυσικοί διαχωριστές των δεδομένων κειμένου. Έτσι, είναι λογικό να κατασκευαστούν μέθοδοι προ-εκπαίδευσης χρησιμοποιώντας τα. Η πρόβλεψη επόμενης πρότασης (NSP), όπως υποδηλώνει και το όνομά της είναι ένα εξαιρετικό παράδειγμα, διότι εκπαιδεύει το μοντέλο να διακρίνει αν δύο προτάσεις εισόδου είναι συνεχή τμήματα από το σώμα εκπαίδευσης. Ο τρόπος λειτουργίας αυτού του task εξηγείται πιο αναλυτικά στο κεφάλαιο 3.3.3.2

- **Sentence Order Prediction (SOP)** Για να μοντελοποιήσει καλύτερα τη συνοχή μεταξύ των προτάσεων, ο αλγόριθμος ALBERT αντικαθιστά την απώλεια NSP με μια απώλεια «πρόβλεψης της σειράς των προτάσεων» (SOP). Με λίγα λόγια, η NSP συνδυάζει την πρόβλεψη θέματος και την πρόβλεψη συνοχής σε μια ενιαία εργασία. Έτσι, το μοντέλο έχει τη δυνατότητα να κάνει προβλέψεις που βασίζονται απλώς στην ευκολότερη εργασία, την πρόβλεψη θέματος. Σε αντίθεση με το NSP, το SOP χρησιμοποιεί δύο διαδοχικά τμήματα από το ίδιο έγγραφο ως θετικά παραδείγματα, και τα ίδια δύο διαδοχικά τμήματα, αλλά με τη σειρά τους αλλαγμένη, ως αρνητικά παραδείγματα. Ως αποτέλεσμα, ο ALBERT υπερτερεί σταθερά έναντι του BERT σε διάφορες μεταγενέστερες εργασίες. Οι StructBERT και BERTje λαμβάνουν επίσης το SOP ως self-supervised learning task.

2.3 BERT model

Ένα από τα πιο ευρέως χρησιμοποιούμενα συστήματα αυτόματης κωδικοποίησης PLM είναι το BERT. Σε αντίθεση με το OpenGPT, το οποίο προβλέπει λέξεις με βάση προηγούμενες προβλέψεις, το BERT εκπαιδεύεται χρησιμοποιώντας την εργασία masked γλωσσικής μοντελοποίησης (MLM), η οποία καλύπτει τυχαία ορισμένα σημεία σε μια ακολουθία κειμένου και στη συνέχεια ανακτά ανεξάρτητα τα masked σημεία με βάση τα διανύσματα κωδικοποίησης που λαμβάνονται από έναν αμφίδρομο μετασχηματιστή. Υπήρξαν πολυάριθμες εργασίες για τη βελτίωση του BERT.

- Το RoBERTa είναι πιο ανθεκτικό από το BERT και εκπαιδεύεται χρησιμοποιώντας πολύ περισσότερα δεδομένα εκπαίδευσης.
- Το ALBERT μειώνει την κατανάλωση μνήμης και αυξάνει την εκπαίδευση του BERT.
- Το DistillBERT χρησιμοποιεί την απόσταξη γνώσης κατά την προ-εκπαίδευση για να μειώσει το μέγεθος του BERT κατά 40%, διατηρώντας το 99% των αρχικών δυνατοτήτων του και κάνοντας την εξαγωγή συμπερασμάτων 60% ταχύτερη.
- Το SpanBERT επεκτείνει το BERT για την καλύτερη αναπαράσταση και πρόβλεψη των διαστημάτων κειμένου.
- Το ERNIE ενσωματώνει γνώση τομέα από εξωτερικές βάσεις γνώσης, όπως ονομαστικές οντότητες, για την προ-εκπαίδευση του μοντέλου.
- Το ALUM εισάγει μια αντίπαλη απώλεια για την προ-εκπαίδευση του μοντέλου που βελτιώνει τη γενίκευση του μοντέλου σε νέες εργασίες και την ανθεκτικότητα σε αντίπαλες επιθέσεις.

2.3.1 Τι είναι το μοντέλο BERT

Ο αλγόριθμος BERT (Bidirectional Encoder Representations from Transformers) είναι μια τεχνική μηχανικής εκμάθησης που χρησιμοποιεί τους μετασχηματιστές (transformers) για την προεκπαίδευση ενός μοντέλου επεξεργασίας φυσικής γλώσσας (NLP). Έχει προκαλέσει αίσθηση στην κοινότητα της μηχανικής μάθησης, παρουσιάζοντας εντυπωσιακά αποτελέσματα σε μια ευρεία ποικιλία εργασιών NLP, όπως η Απάντηση Ερωτήσεων (SQuAD v1.1) και η Συμπερασματολογία Φυσικής Γλώσσας (MNLI), μεταξύ άλλων.

Η καινοτομία του BERT βασίζεται στην αμφίδρομη εκπαίδευση του Transformer, ένα δημοφιλές μοντέλο προσοχής (attention model), για τη μοντελοποίηση των γλωσσών. Αυτό αποτελεί αναβάθμιση σε σχέση με προηγούμενες προσεγγίσεις που εξέταζαν μια ακολουθία κειμένου είτε από αριστερά προς τα δεξιά, είτε συνδυαστικά από αριστερά προς τα δεξιά και από

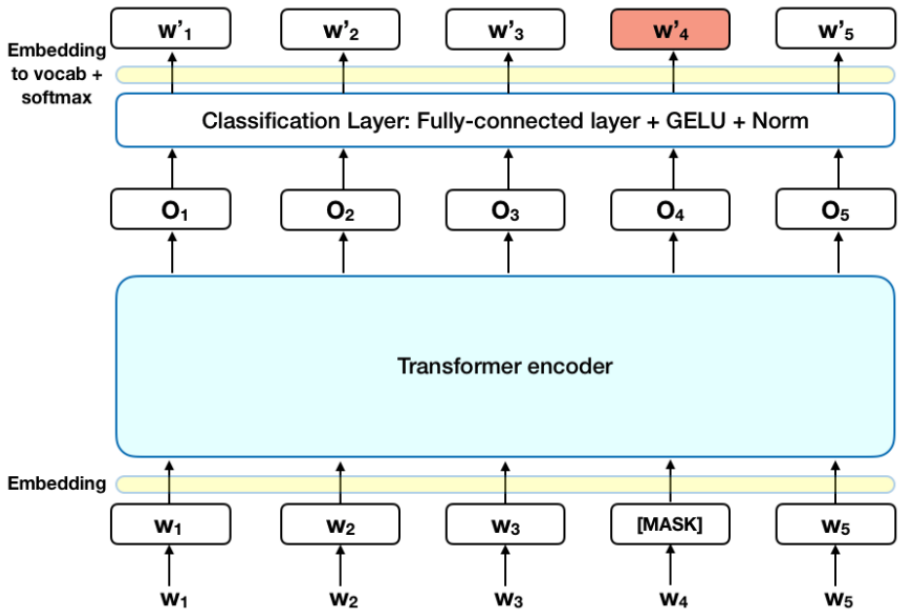
δεξιά προς τα αριστερά. Τα αποτελέσματα δείχνουν ότι ένα γλωσσικό μοντέλο που εκπαιδεύεται αμφίδρομα μπορεί να έχει βαθύτερη κατανόηση του γλωσσικού πλαισίου

2.3.2 Τρόπος λειτουργίας του μοντέλου BERT και ανάλυση προτάσεων

Ένας Transformer όπως αναφέραμε και στο κεφάλαιο 2.1.4, ακολουθεί μια βασική δομή, γνωστή ως "Vanilla", και αποτελείται από δύο διαφορετικούς μηχανισμούς: έναν κωδικοποιητή, ο οποίος επεξεργάζεται την είσοδο του κειμένου, και έναν αποκωδικοποιητή, ο οποίος παράγει μια πρόβλεψη για την εργασία που πρέπει να εκτελεστεί. Ωστόσο, στην περίπτωση του BERT, που στοχεύει στη δημιουργία ενός γλωσσικού μοντέλου, είναι απαραίτητο μόνο ο μηχανισμός κωδικοποιητή για την επεξεργασία της εισόδου του κειμένου.

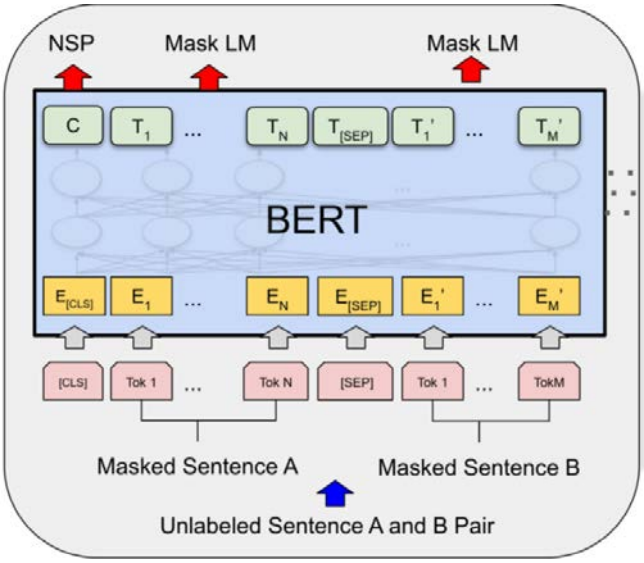
Ο Transformer είναι ένα μοντέλο νευρωνικών δικτύων που λειτουργεί διαφορετικά από τα μοντέλα κατεύθυνσης καθώς διαβάζει όλες τις λέξεις του κειμένου ταυτόχρονα, αντί για το διαδοχικό διάβασμα από αριστερά προς τα δεξιά ή αντίστροφα. Αυτό το χαρακτηριστικό καθιστά το μοντέλο μη κατευθυντικό και επιτρέπει στο μοντέλο να κατανοήσει το συνολικό πλαίσιο της κάθε λέξης, λαμβάνοντας υπόψη τους όρους που προηγούνται και ακολουθούν την κάθε λέξη. Αυτό έχει ως αποτέλεσμα μια πιο πλήρη κατανόηση του κειμένου και μια πιο ακριβή μετάφραση αυτού σε άλλη γλώσσα.

Στο παρακάτω διάγραμμα απεικονίζεται ο κωδικοποιητής Transformer, ένα νευρωνικό δίκτυο που επεξεργάζεται μια ακολουθία από tokens. Η ακολουθία αυτή περνάει αρχικά μέσα από έναν μετασχηματιστή, όπου τα tokens μετατρέπονται σε διανύσματα και επεξεργάζονται αντίστοιχα. Στη συνέχεια, η ακολουθία από τα διανύσματα εξέρχεται ως έξοδος του μοντέλου, όπου κάθε διάνυσμα αντιστοιχεί σε ένα token εισόδου με τον ίδιο δείκτη.



Εικόνα 24: Transformer encoder

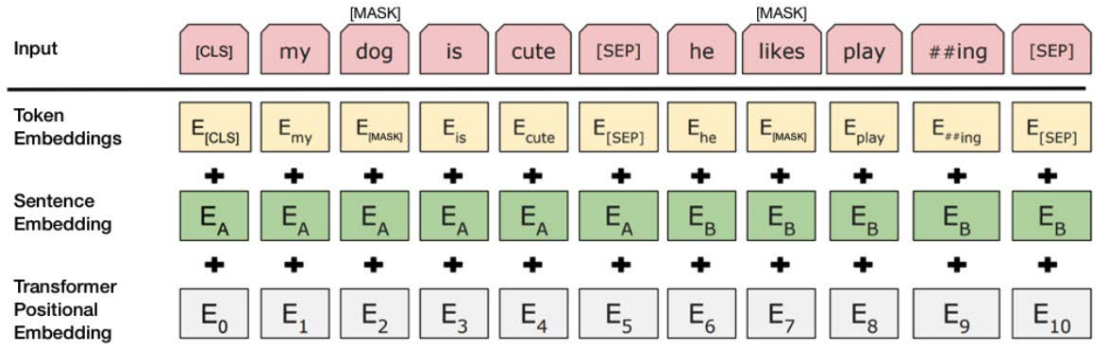
Πιο συγκεκριμένα, ο τρόπος με τον οποίο τα δεδομένα εισόδου γίνονται αντικείμενο επεξεργασίας είναι πολύ ξεχωριστός, επειδή είναι σχεδιασμένος να λειτουργεί σε ζεύγη προτάσεων. Αυτός ο τρόπος πραγματοποιείται προπαρασκευάζοντας ένα ειδικό σύμβολο (CLS) στην αρχή της εισόδου, έτσι ώστε να αναγνωρίζουμε ότι αυτό είναι το σημείο έναρξης της πρώτης πρότασης, και ύστερα ακολουθεί το σύμβολο διαχωρισμού (SEP), το οποίο υποδηλώνει την αρχή της δεύτερης πρότασης του ζεύγους. Αυτή η μέθοδος είναι πολύ χρήσιμη για τις εργασίες οι οποίες περιέχουν πολλές προτάσεις, όπως για παράδειγμα κείμενα ερωτήσεων/απαντήσεων. Ακόμη, η χρήση των συμβόλων διευκολύνει τον αλγόριθμο, διότι μπορούμε να επεξεργαστούμε με μεγαλύτερη ευκολία και πιο ολοκληρωτικά την ακολουθία εισόδου.



Εικόνα 25: BERT

Πηγή : https://pytorch.org/tutorials/intermediate/dynamic_quantization_bert_tutorial.html

Ύστερα προσθέτουμε ένα learned embedding σε κάθε token που υποδεικνύει αν ανήκει στην πρόταση A ή πρόταση B. Όπως βλέπουμε στην Εικόνα 25, συμβολίζουμε το input embedding ως E, το τελικό κρυφό διάνυσμα του ειδικού token [CLS] ως C∈R H, και το τελικό κρυφό διάνυσμα για το i-οστό token εισόδου ως Ti ∈R H. Για ένα δεδομένο token, η αναπαράσταση εισόδου του κατασκευάζεται αθροίζοντας τα αντίστοιχα token, sentence και position embeddings. Μια απεικόνιση αυτής της κατασκευής φαίνεται στην Εικόνα 26.



Εικόνα 26: BERT input (token embeddings, segmentation embeddings and position embeddings)

Πηγή : https://www.kdnuggets.com/2018/12/bert-sota-nlp-model-explained.html

2.3.3 Μοντέλα Εκπαίδευσης BERT

Στην εκπαίδευση γλωσσικών μοντέλων, ένα από τα μεγαλύτερα προβλήματα είναι η ανάγκη να καθοριστεί ο στόχος πρόβλεψης του μοντέλου. Πολλές προσεγγίσεις προβλέπουν απλά την επόμενη λέξη σε μια ακολουθία, περιορίζοντας έτσι την ικανότητα του μοντέλου να εκμεταλλευτεί την σημασιολογική συνέχεια των λέξεων στην ακολουθία. Για να αντιμετωπιστεί αυτό το πρόβλημα, ο BERT χρησιμοποιεί δύο στρατηγικές εκπαίδευσης που του επιτρέπουν να κατανοήσει τις λέξεις σε σχέση με το συνολικό περιβάλλον τους. Αυτές οι στρατηγικές είναι :

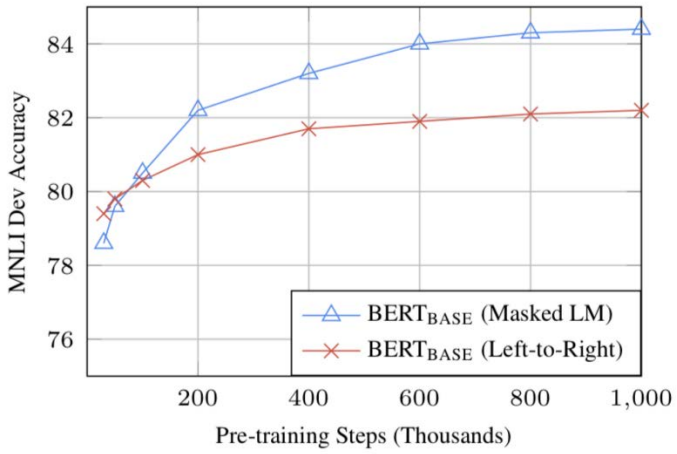
2.3.3.1 Masked LM (MLM)

Η Προσαρμοστική Μάθηση Μάσκας (Masked Language Modeling - MLM) είναι μια διαδικασία στην επεξεργασία φυσικής γλώσσας (Natural Language Processing - NLP), όπου ένα μοντέλο εκπαιδεύεται να προβλέπει μία λέξη που λείπει ή ένα token σε ένα δεδομένο κείμενο. Η εργασία απαιτεί τη προσθήκη ενός σύμβολου [MASK] συγκεκριμένου ποσοστού (15%) από τα tokens στην είσοδο και ως αποτέλεσμα απαιτεί από το μοντέλο να προβλέψει το σωστό token με βάση το περιβάλλοντα περιεχόμενο.

Κατά τη διάρκεια της εκπαίδευσης, ένα συγκεκριμένο ποσοστό των tokens στην είσοδο τυχαία γίνεται masked και το μοντέλο εκπαιδεύεται να προβλέπει τα masked tokens βάσει των υπόλοιπων unmasked tokens στην είσοδο.

Η MLM χρησιμοποιείται ευρέως στην προεκπαίδευση μοντέλων γλώσσας που βασίζονται σε μετασχηματιστές (transformer-based language models) όπως το BERT. Προ-εκπαιδεύοντας το μοντέλο στην εργασία MLM, το μοντέλο μπορεί να μάθει να κατανοεί το περιβάλλοντα περιεχόμενο και τη σημασία του κειμένου στη φυσική γλώσσα, το οποίο μπορεί να εφαρμοστεί σε εφαρμογές NLP όπως ανάλυση απόχρωσης, αναγνώριση ονοματολογιών και μηχανική μετάφραση.

Η συνάρτηση απώλειας του BERT αξιοποιεί μόνο τις προβλέψεις που αναφέρονται στις λέξεις που περιλαμβάνονται στην πρόταση και αγνοεί τις προβλέψεις για λέξεις που δεν περιλαμβάνονται σε αυτήν. Κατά συνέπεια, η αμφίδρομη προσέγγιση του BERT (MLM) συγκλίνει πιο αργά από τις προσεγγίσεις από αριστερά προς τα δεξιά (επειδή μόνο το 15% των λέξεων προβλέπεται σε κάθε παρτίδα), αλλά η αμφίδρομη εκπαίδευση εξακολουθεί να υπερτερεί έναντι της εκπαίδευσης από αριστερά προς τα δεξιά μετά από ένα μικρό αριθμό βημάτων προ-εκπαίδευσης.



Εικόνα 27: Masked LM

Πρακτικά, η υλοποίηση του BERT είναι ελαφρώς πιο περίπλοκη και δεν αντικαθιστά όλες τις λέξεις που έχουν αποκρυφτεί κατά 15%.

Η εκπαίδευση του γλωσσικού μοντέλου στο BERT γίνεται με την πρόβλεψη του 15% των tokens της εισόδου, τα οποία επιλέχθηκαν τυχαία.

Αυτά τα tokens προ-επεξεργάζονται ως εξής - το 80% αντικαθίσταται με ένα token "[MASK]", το 10% με μια τυχαία λέξη και το 10% χρησιμοποιεί την αρχική λέξη. Οι λόγοι που επιλέγουμε αυτήν την προσέγγιση είναι η εξής :

- Εάν χρησιμοποιούσαμε το [MASK] 100% του χρόνου, το μοντέλο δεν θα παρήγαγε απαραίτητα καλές αναπαραστάσεις συμβόλων για unmasked λέξεις. Τα unmasked tokens εξακολουθούσαν να χρησιμοποιούνται για το πλαίσιο, αλλά το μοντέλο βελτιστοποιήθηκε για την πρόβλεψη των masked λέξεων.
- Αν χρησιμοποιούσαμε [MASK] 90% του χρόνου και τυχαίες λέξεις 10% του χρόνου, αυτό θα μάθαινε στο μοντέλο ότι η παρατηρούμενη λέξη δεν είναι ποτέ σωστή.
- Αν χρησιμοποιούσαμε [MASK] το 90% του χρόνου και διατηρούσαμε την ίδια λέξη το 10% του χρόνου, τότε το μοντέλο θα μπορούσε απλά να αντιγράψει τυχαία ένα non-contextual embedding.

2.3.3.2 Next Sentence Prediction (NSP)

Η τεχνική Next Sentence Prediction (NSP) χρησιμοποιείται στο προεπεξεργαστικό στάδιο του BERT για να βοηθήσει το μοντέλο να αντιληφθεί τη συσχέτιση μεταξύ δύο προτάσεων σε ένα κείμενο.

Συγκεκριμένα, για κάθε ζεύγος προτάσεων που παρέχεται στο BERT ως είσοδος, το μοντέλο καλείται να αποφανθεί εάν η δεύτερη πρόταση ακολουθεί αμέσως την πρώτη ή όχι. Έτσι, το NSP βοηθάει το BERT να κατανοήσει το πλαίσιο και τον συνολικό σκοπό του κειμένου και να διατηρήσει μια κατανόηση του νοήματος του κειμένου σε μεγαλύτερη κλίμακα.

Η διαδικασία που ακολουθεί η τεχνική Next Sentence Prediction (NSP) στο προεπεξεργαστικό στάδιο του BERT είναι η εξής:

1. Αρχικά, το κείμενο διαχωρίζεται σε προτάσεις.
2. Έπειτα, δημιουργούνται ζεύγη προτάσεων, το κάθε ένα από τα οποία αποτελείται από δύο συνεχόμενες προτάσεις.
3. Για κάθε ζεύγος προτάσεων, το BERT λαμβάνει τις δύο προτάσεις ως είσοδο και καλείται να αποφανθεί εάν η δεύτερη πρόταση ακολουθεί αμέσως την πρώτη ή όχι.
4. Η επιλογή του BERT για κάθε ζεύγος προτάσεων (δηλαδή αν η δεύτερη πρόταση ακολουθεί αμέσως την πρώτη ή όχι) χρησιμοποιείται από το μοντέλο κατά τη διαδικασία εκπαίδευσης του, προκειμένου να βελτιωθεί η ικανότητά του να κατανοήσει το πλαίσιο και τον συνολικό σκοπό του κειμένου.

Εικόνα 28: NSP, Next Sentence Prediction

Κατά την εκπαίδευση του μοντέλου BERT, τα Masked LM και Next Sentence Prediction (NSP) εκπαιδεύονται ταυτόχρονα με στόχο την ελαχιστοποίηση της συνδυασμένης συνάρτησης απώλειας των δύο μεθόδων.

2.3.4 How to use BERT (Fine-tuning)

Η χρήση του BERT για μια συγκεκριμένη εργασία είναι σχετικά απλή: Η BERT μπορεί να χρησιμοποιηθεί για μια μεγάλη ποικιλία γλωσσικών εργασιών, προσθέτοντας μόνο ένα μικρό επίπεδο στο βασικό μοντέλο [6]:

1. Εργασίες ταξινόμησης, όπως η ανάλυση συναισθήματος (sentiment analysis), γίνονται παρόμοια με την ταξινόμηση Επόμενης πρότασης (NSP), προσθέτοντας ένα επίπεδο ταξινόμησης πάνω στην έξοδο του μετασχηματιστή για το διακριτικό [CLS].
2. Στις εργασίες απάντησης ερωτήσεων (π.χ. SQuAD v1.1), το λογισμικό λαμβάνει μια ερώτηση σχετικά με μια ακολουθία κειμένου και καλείται να επισημάνει την απάντηση στην ακολουθία. Με τη χρήση του BERT, μπορεί να εκπαιδευτεί ένα μοντέλο ερωτήσεων και απαντήσεων με την προσθήκη δύο επιπλέον διανυσμάτων που υποδηλώνουν την αρχή και το τέλος της απάντησης.
3. Στην αναγνώριση ονομαστικών οντοτήτων (NER), το λογισμικό λαμβάνει μια ακολουθία κειμένου και καλείται να επισημάνει τους διάφορους τύπους οντοτήτων (πρόσωπο, οργανισμός, ημερομηνία κ.λπ.) που εμφανίζονται στο κείμενο. Χρησιμοποιώντας το BERT, μπορεί να εκπαιδευτεί ένα μοντέλο NER τροφοδοτώντας το διάνυσμα εξόδου κάθε συμβόλου σε ένα επίπεδο ταξινόμησης που προβλέπει την ετικέτα NER.

Κατά την εκπαίδευση λεπτομερούς ρύθμισης (fine-tuning training), οι περισσότερες υπερ-παράμετροι παραμένουν οι ίδιες όπως και στην εκπαίδευση BERT.

2.3.5 Συμπέρασμα

Το BERT αποτελεί ένα από τα πιο προηγμένα μοντέλα μηχανικής μάθησης που χρησιμοποιούνται σήμερα για την επεξεργασία φυσικής γλώσσας. Το μοντέλο BERT βασίζεται στο Transformer και εκπαιδεύεται σε μεγάλα σώματα κειμένων με τη χρήση δύο κύριων καθηκόντων: το Masked Language Modeling (MLM) και το Next Sentence Prediction (NSP).

Το MLM απαιτεί από το μοντέλο να προβλέψει λέξεις που έχουν γινει masked στο κείμενο, ενώ το NSP αξιοποιείται για να προβλέψει αν δύο προτάσεις είναι συνδεδεμένες ή όχι. Οι δύο αυτοί στόχοι συμβάλλουν στην κατανόηση του μοντέλου της γλώσσας και στην αναπαράστασή του στον αλγόριθμο μάθησης.

3 *Ταξινόμηση Κειμένου*

3.1 Σε τι είδους εφαρμογές χρησιμοποιείται

Η ταξινόμηση κειμένου (TC) είναι η διαδικασία κατηγοριοποίησης κειμένων (π.χ. tweets, άρθρα ειδήσεων, κριτικές πελατών) σε οργανωμένες ομάδες. Συνήθεις εργασίες TC περιλαμβάνουν την ανάλυση συναισθήματος, την κατηγοριοποίηση ειδήσεων και την ταξινόμηση θεμάτων. Πρόσφατα, οι ερευνητές έδειξαν ότι είναι αποτελεσματικό να αποδίδονται πολλές εργασίες κατανόησης φυσικής γλώσσας (NLU) (π.χ. εξαγωγή ερωτήσεων, εξαγωγή συμπερασμάτων φυσικής γλώσσας) ως TC, επιτρέποντας σε ταξινομητές κειμένου βασισμένους σε DL να λαμβάνουν ένα ζεύγος κειμένων ως είσοδο.

Παρακάτω θα αναλύσουμε πέντε είδη εφαρμογών, στα οποία χρησιμοποιούμε ταξινόμηση κειμένου, συμπεριλαμβανομένων τριών τυπικών εργασιών TC και δύο εργασιών NLU, οι οποίες συνήθως εκτίθενται ως TC σε πολλές πρόσφατες μελέτες DL[7].

Ανάλυση συναισθήματος (Sentiment Analysis). Πρόκειται για την ανάλυση των απόψεων των ανθρώπων σε δεδομένα κειμένου (π.χ. κριτικές προϊόντων, κριτικές ταινιών ή tweets) και της εξαγωγής της άποψής τους. Η εργασία μπορεί να θεωρηθεί είτε ως δυαδικό είτε ως πρόβλημα πολλαπλών κλάσεων. Η δυαδική ανάλυση συναισθήματος ταξινομεί τα κείμενα σε θετικές και αρνητικές κλάσεις, ενώ η ανάλυση συναισθήματος πολλαπλών κλάσεων ταξινομεί τα κείμενα με fine-grained labels ή multi-level intensities.

Κατηγοριοποίηση ειδήσεων (News Categorization). Το περιεχόμενο των ειδήσεων είναι από τις σημαντικότερες πηγές πληροφοριών. Ένα σύστημα κατηγοριοποίησης ειδήσεων βοηθά τους χρήστες να λαμβάνουν πληροφορίες που τους ενδιαφέρουν σε πραγματικό χρόνο, π.χ. εντοπίζοντας αναδυόμενα θέματα ειδήσεων ή συνιστώντας σχετικές ειδήσεις με βάση τα ενδιαφέροντα του χρήστη.

Ανάλυση θεμάτων (Topic Analysis). Η εργασία (task), γνωστή και ως θεματική ταξινόμηση, αποσκοπεί στον εντοπισμό του θέματος ή των θεμάτων ενός κειμένου (π.χ. αν μια κριτική προϊόντος αφορά την "υποστήριξη πελατών" ή την "ευκολία χρήσης").

Αναγνώριση συγγραφικής Πατρότητας. Αφορά την αναγνώριση του συγγραφέα ενός κειμένου βασιζόμενη σε στοιχεία όπως ο στυλ γραφής, η χρήση λέξεων και η γλωσσική δομή. Αυτή η τεχνολογία μπορεί να χρησιμοποιηθεί για την ταξινόμηση κειμένου, όπως για παράδειγμα για τη διάκριση ανάμεσα σε διαφορετικούς συγγραφείς σε ένα σύνολο κειμένων. Με τη χρήση αλγορίθμων μηχανικής μάθησης, μπορούν να εξαχθούν χαρακτηριστικά από το κείμενο που μπορούν να βοηθήσουν στην αναγνώριση του συγγραφέα.

Αναγνώριση του υφολογικού είδους ενός κειμένου. Μια σημαντική μέθοδος για την αυτόματη ταξινόμηση του κειμένου. Με τη χρήση αλγορίθμων μηχανικής μάθησης, μπορούμε να αναγνωρίζουμε το υφολογικό είδος του κειμένου βάσει των γλωσσικών χαρακτηριστικών του, όπως η συντακτική δομή, το λεξιλόγιο και η χρήση των γλωσσικών στοιχείων. Η ταξινόμηση κειμένου βασιζόμενη στο υφολογικό είδος μπορεί να χρησιμοποιηθεί σε πολλά πεδία, όπως στη διαχείριση περιεχομένου, στην ανάλυση κοινωνικών μέσων και στην ανίχνευση παραπλανητικού περιεχομένου.

3.2 Μέθοδοι Ταξινόμησης Κειμένου

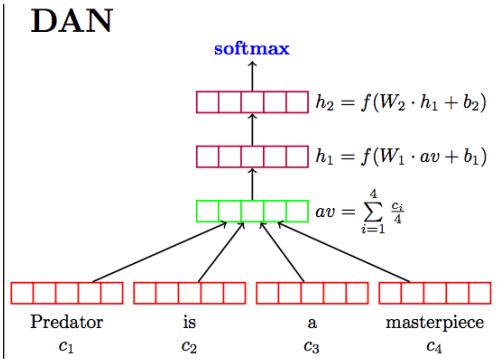
Λόγω της πληθώρας διαφορετικών προβλημάτων τα οποία έχουν δημιουργηθεί στον τομέα της ταξινόμησης κειμένων, έχουν αντίστοιχα κατασκευαστεί κατάλληλα μοντέλα, έτσι ώστε η επίλυση αυτού του προβλήματος να γίνεται με μεγαλύτερη αποτελεσματικότητα. Παρακάτω, θα αναλύσουμε λεπτομερώς κάποιες από τις εφαρμογές ταξινόμησης κειμένων από κάθε κατηγορία.

3.2.1 Feed-Forward Neural Networks για Ταξινόμηση Κειμένων

Είναι από τα απλούστερα μοντέλα DL για την αναπαράσταση κειμένου. Ωστόσο, έχουν επιτύχει υψηλή ακρίβεια σε πολλούς δείκτες αναφοράς TC. Αυτά τα μοντέλα θεωρούν το κείμενο ως μια «σακούλα» λέξεων (bag of words). Για κάθε λέξη, μαθαίνουν μια διανυσματική αναπαράσταση χρησιμοποιώντας ένα μοντέλο ενσωμάτωσης (embedding model), όπως το word2vec, λαμβάνουν το άθροισμα των διανυσμάτων ή το μέσο όρο των ενσωματώσεων ως αναπαράσταση του κειμένου, το περνούν από ένα ή περισσότερα στρώματα προώθησης, γνωστά ως Multi-Layer Perceptrons (MLPs) ή αλλιώς δίκτυο “Vanilla”, και στη συνέχεια εκτελούν ταξινόμηση στην αναπαράσταση του τελευταίου στρώματος χρησιμοποιώντας έναν ταξινομητή.

Ένα παράδειγμα αυτών των μοντέλων είναι το **Deep Average Network (DAN)**, η αρχιτεκτονική του οποίου παρουσιάζεται στο παρακάτω σχήμα. Παρά

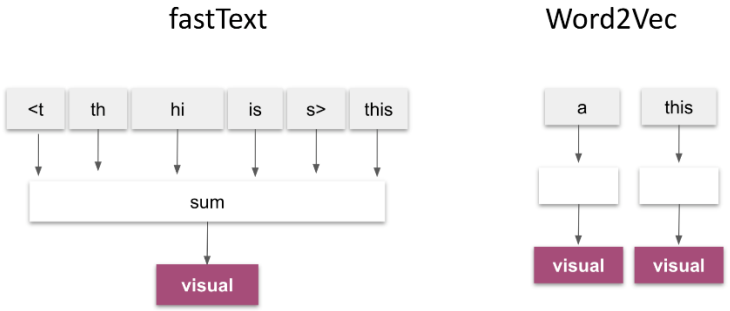
την απλότητά του, το DAN ξεπερνά άλλα πιο σύνθετα μοντέλα που σχεδιάστηκαν ειδικά για να μάθουν εκ των προτέρων τον συντακτικό συνδυασμό των κειμένων. Για παράδειγμα, το DAN υπερτερεί των συντακτικών μοντέλων σε σύνολα δεδομένων με υψηλή συντακτική διακύμανση.



Εικόνα 29: Deep Average Network (DAN)

Πηγή: https://www.researchgate.net/figure/The-architecture-of-the-Deep-Average-Network-DAN-10_fig1_340523298

Ακόμη, πολλοί προτείνουν έναν απλό και αποτελεσματικό ταξινομητή κειμένου που ονομάζεται **fastText**. Όπως το DAN, το fastText θεωρεί το κείμενο ως bag of words. Σε αντίθεση με το DAN, το fastText χρησιμοποιεί n-grams ως πρόσθετα χαρακτηριστικά, για να καταγράψει τοπικές πληροφορίες για τη σειρά των λέξεων. Αυτό αποδεικνύεται πολύ αποτελεσματικό στην πράξη, επιτυγχάνοντας συγκρίσιμα αποτελέσματα με τις μεθόδους που χρησιμοποιούν ρητά τη σειρά των λέξεων.



Εικόνα 30 : fastText vs Word2Vec

Πηγή: https://kavita-ganesan.com/fasttext-vs-word2vec/

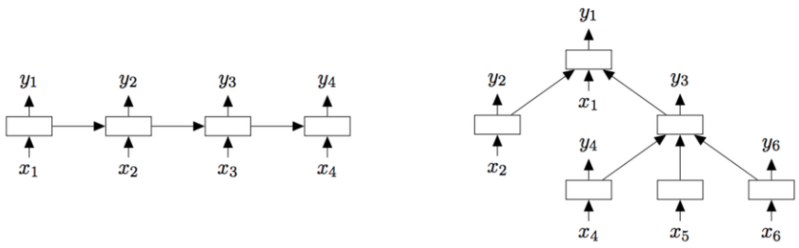
3.2.2 *RNN-Based models για Ταξινόμηση Κειμένων*

Ο Tai κ.α.[8] ανέπτυξαν ένα μοντέλο Tree-LSTM, μια γενίκευση του LSTM σε τυπολογίες δικτύων με δενδροειδή δομή (tree-structured), για να μαθαίνουν πλούσιες σημασιολογικές αναπαραστάσεις. Γενικότερα, το Tree-LSTM είναι καλύτερο μοντέλο από το αλυσιδωτά δομημένο (chain-structured) LSTM για εργασίες NLP, επειδή η φυσική γλώσσα παρουσιάζει συντακτικές ιδιότητες που με φυσικό τρόπο συνδυάζει λέξεις σε φράσεις.

Επαληθεύουν την αποτελεσματικότητα του Tree-LSTM σε δύο tasks: μέσω της ταξινόμησης με βάση το συναίσθημα και της πρόβλεψης της σημασιολογικής συγγένειας δύο προτάσεων. Οι αρχιτεκτονικές αυτών των μοντέλων παρουσιάζονται στην Εικόνα 31.

Ο Zhu κ.α.[9] επίσης επέκτειναν την αλυσιδωτή δομή LSTM (chain-structured) σε δομές δέντρου, χρησιμοποιώντας ένα κελί μνήμης για την αποθήκευση του ιστορικού πολλαπλών 'απογόνων' κελιών σε μια αναδρομική διαδικασία. Υποστηρίζουν ότι το νέο μοντέλο παρέχει πιο σωστό τρόπο εξέτασης των long-distance αλληλεπιδράσεων σε ιεραρχίες, π.χ. δομές ανάλυσης γλώσσας ή εικόνας.

Παρακάτω παρουσιάζονται οι αρχιτεκτονικές αυτών των δυο μοντέλων.



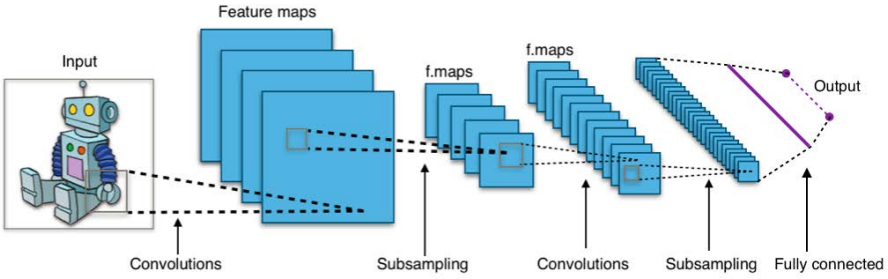
Εικόνα 31: Tree-LSTM

Πηγή: https://www.researchgate.net/figure/Left-A-chain-structured-LSTM-network-and-right-a-tree-structured-LSTM-network-with_fig3_340523298

3.2.3 *CNN-Based models για Ταξινόμηση Κειμένων*

Ένα από τα πρώτα μοντέλα βασισμένα σε CNN για TC ονομάζεται Dynamic CNN (DCNN) και χρησιμοποιεί δυναμική *k*-max-pooling. Όπως απεικονίζεται στην Εικόνα 32, το πρώτο επίπεδο του DCNN κατασκευάζει έναν πίνακα προτάσεων χρησιμοποιώντας την ενσωμάτωση για κάθε λέξη της πρότασης. Στη συνέχεια, μια αρχιτεκτονική συνέλιξης που εναλλάσσει ευρεία στρώματα συνέλιξης με δυναμικά στρώματα συγκέντρωσης που δίνονται από τη

δυναμική k-max συγκέντρωση χρησιμοποιείται για τη δημιουργία ενός χάρτη χαρακτηριστικών πάνω στην πρόταση, ο οποίος είναι ικανός να αποτυπώνει ρητά τις σχέσεις μικρής και μεγάλης εμβέλειας των λέξεων και των φράσεων. Η παράμετρος pooling k μπορεί να επιλέγεται δυναμικά ανάλογα με το μέγεθος της πρότασης και το επίπεδο στην ιεραρχία συνέλιξης.



Εικόνα 32: CNN-Based models

Πηγή: https://subscription.packtpub.com/book/big-data-and-business-intelligence/9781787128422/3/ch03lvl1sec24/deep-convolutional-neural-network-dcnn

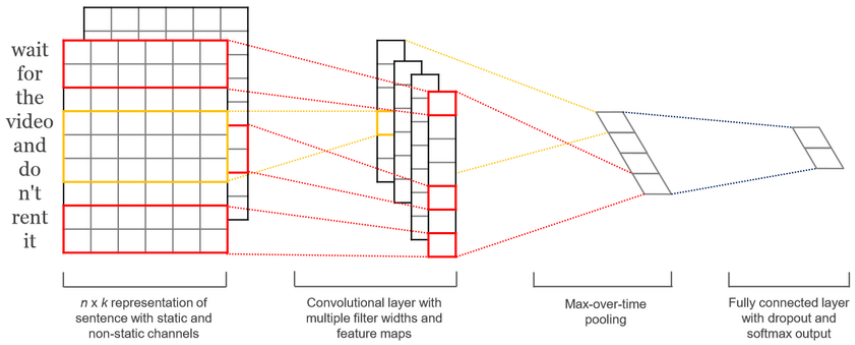
Αργότερα, από τον Yoon Kim [10] προτάθηκε ένα πολύ απλούστερο μοντέλο με βάση το CNN από το DCNN για το TC. Όπως φαίνεται στην Εικόνα 33, το μοντέλο του Kim χρησιμοποιεί μόνο ένα επίπεδο συνέλιξης πάνω από τα διανύσματα λέξεων που λαμβάνονται από ένα μη επιβλεπόμενο νευρωνικό γλωσσικό μοντέλο, δηλαδή το **word2vec**. Ο Kim συγκρίνει επίσης τέσσερις διαφορετικές προσεγγίσεις για την εκμάθηση της ενσωμάτωσης λέξεων:

1. CNN-rand, όπου όλες οι ενσωματώσεις λέξεων αρχικοποιούνται τυχαία και στη συνέχεια τροποποιούνται κατά τη διάρκεια της εκπαίδευσης

2. CNN-static, όπου χρησιμοποιούνται οι προεκπαιδευμένες ενσωματώσεις word2vec και παραμένουν σταθερές κατά τη διάρκεια της εκπαίδευσης του μοντέλου

3. CNN-non-static, όπου οι ενσωματώσεις word2vec ρυθμίζονται λεπτομερώς κατά τη διάρκεια της εκπαίδευσης για κάθε εργασία

4. CNN-multi-channel, χρησιμοποιεί δύο σετ διανυσμάτων ενσωμάτωσης λέξεων, τα οποία προέρχονται από τη μέθοδο word2vec. Και τα δύο σετ αρχικοποιούνται με τη μέθοδο αυτή, αλλά ένα εκ των δύο ενημερώνεται κατά τη διάρκεια της εκπαίδευσης του μοντέλου, ενώ το άλλο παραμένει σταθερό καθ' όλη τη διαδικασία. Αναφέρεται ότι αυτά τα μοντέλα που βασίζονται σε CNN βελτιώνουν την κατάσταση της τεχνολογίας στην ανάλυση συναισθήματος και την ταξινόμηση ερωτήσεων μιας και έχουν γίνει προσπάθειες βελτίωσης των αρχιτεκτονικών των μοντέλων που βασίζονται σε CNN.

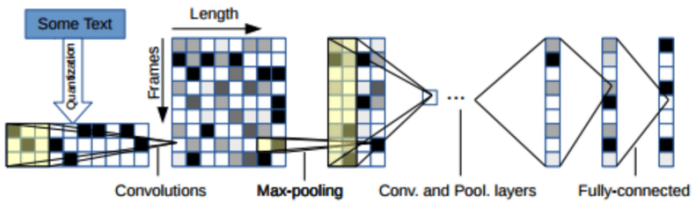


Εικόνα 33: The architecture of CNN model for text classification

Πηγή: https://www.researchgate.net/figure/The-architecture-of-a-sample-CNN-model-for-text-classification-courtesy-of-Yoon-Kim-27_fig5_340523298

Τα CNNs σε επίπεδο χαρακτήρων έχουν επίσης διερευνηθεί για το TC. Ένα από τα πρώτα τέτοια μοντέλα προτείνεται από τους Zhang κ.α.[11]. Όπως απεικονίζεται στην Εικόνα 34, το μοντέλο δέχεται ως είσοδο τους χαρακτήρες σε fixed-sized, κωδικοποιημένους ως διανύσματα μιας στιγμής, τους περνάει μέσα από ένα βαθιάς μάθησης μοντέλο CNN που αποτελείται από έξι επίπεδα συνέλιξης με λειτουργίες συγκέντρωσης και τρία πλήρως συνδεδεμένα επίπεδα.

Ο Prusa κ.α. [12] ακόμη παρουσίασαν μια προσέγγιση για την κωδικοποίηση κειμένου με χρήση CNN που μειώνει σημαντικά την κατανάλωση μνήμης και το χρόνο εκπαίδευσης που απαιτείται για την εκμάθηση αναπαραστάσεων κειμένου σε επίπεδο χαρακτήρων. Αυτή η προσέγγιση κλιμακώνεται καλά με το μέγεθος του αλφαβήτου, επιτρέποντας τη διατήρηση περισσότερων πληροφοριών από το αρχικό κείμενο για την ενίσχυση της απόδοσης ταξινόμησης.



Εικόνα 34 : Character-level CNN

Πηγή:https://www.researchgate.net/figure/Diagram-of-character-level-CNN-model-architecture-Figure-reproduced-from-Zhang-et-al_fig1_333418860

3.2.4 Transformers για Ταξινόμηση Κειμένων

Τα μοντέλα Transformer έχουν χρησιμοποιηθεί εκτενώς σε εργασίες ταξινόμησης κειμένου, λόγω της ικανότητάς τους να μαθαίνουν αναπαραστάσεις κειμένου με βάση τα συμφραζόμενα.

Παρακάτω παρατίθενται ορισμένες εφαρμογές των μοντέλων μετασχηματιστών στην ταξινόμηση κειμένου:

- **Ανάλυση συναισθήματος**: Μοντέλα Transformer όπως τα BERT και RoBERTa έχουν επιτύχει κορυφαία αποτελέσματα σε εργασίες ανάλυσης συναισθήματος. Αυτά τα μοντέλα μπορούν να μάθουν να ταξινομούν το συναίσθημα ενός κειμένου ως θετικό, αρνητικό ή ουδέτερο, χρησιμοποιώντας τις πληροφορίες πλαισίου του κειμένου.
- **Ταξινόμηση θεμάτων**: Τα Transformers μπορούν επίσης να χρησιμοποιηθούν για εργασίες θεματικής ταξινόμησης, όπου το μοντέλο πρέπει να αποδώσει μια ετικέτα θέματος σε ένα κομμάτι κειμένου. Αυτό μπορεί να είναι χρήσιμο σε εφαρμογές όπως η κατηγοριοποίηση ειδήσεων ή η ανάλυση μέσων κοινωνικής δικτύωσης.
- **Named Entity Recognition:** Τα μοντέλα Transformer μπορούν να χρησιμοποιηθούν για εργασίες αναγνώρισης ονομαστικών οντοτήτων, όπου το μοντέλο πρέπει να εντοπίσει και να ταξινομήσει οντότητες όπως άτομα, οργανισμούς και τοποθεσίες σε ένα κείμενο.

3.2.5 The attention mechanism για Ταξινόμηση Κειμένων

Παρακάτω εξετάζουμε ορισμένα από τα πιο σημαντικά μοντέλα προσοχής που δημιουργούν νέα δεδομένα στις ΤΣ εργασίες.

Οι Yang κ.α. [13] προτείνουν ένα ιεραρχικό δίκτυο προσοχής για την ταξινόμηση κειμένων. Το μοντέλο αυτό έχει δύο διακριτά χαρακτηριστικά:

1. μια ιεραρχική δομή που αντικατοπτρίζει την ιεραρχική δομή των εγγράφων και
2. δύο επίπεδα μηχανισμών προσοχής που εφαρμόζονται σε επίπεδο λέξης και πρότασης, επιτρέποντάς του να παρακολουθεί διαφορετικά το περισσότερο και το λιγότερο σημαντικό περιεχόμενο κατά την κατασκευή της αναπαράστασης του εγγράφου.

Αυτό το μοντέλο υπερτερεί των προηγούμενων μεθόδων με σημαντική διαφορά σε έξι εργασίες ΤΣ.

Οι Zhou κ.α. [14] επεκτείνουν το μοντέλο ιεραρχικής προσοχής σε διαγλωσσική ταξινόμηση συναισθήματος. Σε κάθε γλώσσα, χρησιμοποιείται ένα

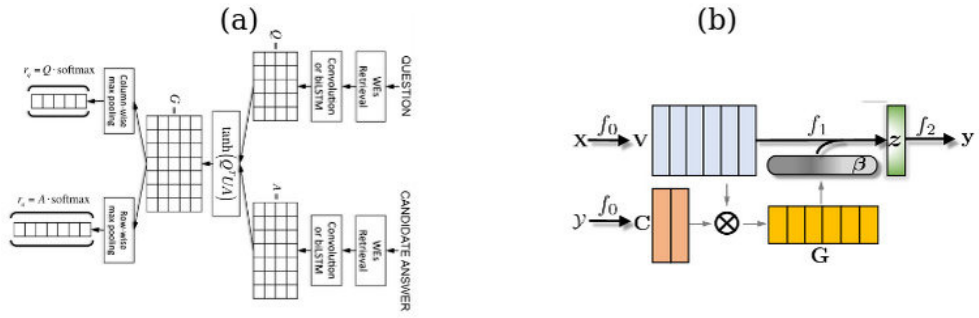
δίκτυο LSTM για τη μοντελοποίηση των εγγράφων. Στη συνέχεια, η ταξινόμηση επιτυγχάνεται με τη χρήση ενός μηχανισμού ιεραρχικής προσοχής, όπου το μοντέλο προσοχής σε επίπεδο πρότασης μαθαίνει ποιες προτάσεις ενός εγγράφου είναι πιο σημαντικές για τον καθορισμό του συνολικού συναισθήματος, ενώ το μοντέλο προσοχής σε επίπεδο λέξης μαθαίνει ποιες λέξεις σε κάθε πρόταση είναι καθοριστικές.

Οι Shen κ.α. [15] παρουσιάζουν ένα δίκτυο κατευθυνόμενης αυτοπροσοχής (self-attention) για γλωσσική κατανόηση χωρίς RNN/CNN, όπου η προσοχή μεταξύ στοιχείων από ακολουθία(ες) εισόδου είναι κατευθυνόμενη και πολυδιάστατη. Ένα ελαφρύ νευρωνικό δίκτυο χρησιμοποιείται για την εκμάθηση της ενσωμάτωσης προτάσεων, αποκλειστικά με βάση την προτεινόμενη προσοχή χωρίς καμία δομή RNN/CNN.

Οι Liu κ.α. [16] παρουσιάζουν ένα μοντέλο LSTM με εσωτερική προσοχή για NLI. Αυτό το μοντέλο χρησιμοποιεί μια διαδικασία δύο σταδίων για την κωδικοποίηση μιας πρότασης. Πρώτον, χρησιμοποιείται η συγκέντρωση του μέσου όρου σε επίπεδο λέξεων Bi-LSTMs για τη δημιουργία μιας αναπαράστασης πρότασης σε πρώτο στάδιο. Δεύτερον, ο μηχανισμός προσοχής χρησιμοποιείται για να αντικαταστήσει τη μέση συγκέντρωση στην ίδια πρόταση για καλύτερες αναπαραστάσεις. Η αναπαράσταση του πρώτου σταδίου της πρότασης χρησιμοποιείται για να παρακολουθήσει τις λέξεις που εμφανίζονται στην ίδια. Τα μοντέλα προσοχής εφαρμόζονται ευρέως και σε εργασίες κατάταξης κατά ζεύγη ή αντιστοίχισης κειμένου.

Οι Santos κ.α. [17] προτείνουν έναν μηχανισμό αμφίδρομης προσοχής, γνωστό ως Attentive Pooling (AP), για την κατάταξη ανά ζεύγη. Το AP επιτρέπει στο στρώμα συγκέντρωσης να γνωρίζει το τρέχον ζεύγος εισόδου (π.χ. ένα ζεύγος ερώτησης-απάντησης), με τρόπο ώστε οι πληροφορίες από τα δύο στοιχεία εισόδου να μπορούν να επηρεάσουν άμεσα τον υπολογισμό των αναπαραστάσεων του άλλου. Εκτός από την εκμάθηση των αναπαραστάσεων του ζεύγους εισόδου, το AP μαθαίνει από κοινού ένα μέτρο ομοιότητας σε προβαλλόμενα τμήματα του ζεύγους και στη συνέχεια εξάγει το αντίστοιχο διάνυσμα προσοχής για κάθε είσοδο για να καθοδηγήσει τη συγκέντρωση. Το AP είναι ένα γενικό πλαίσιο ανεξάρτητο από την υποκείμενη εκμάθηση αναπαράστασης και μπορεί να εφαρμοστεί τόσο σε CNN όσο και σε RNN, όπως απεικονίζεται στην Εικόνα 35 (α).

Οι Wang κ.α. [18] θεωρούν το TC ως ένα πρόβλημα αντιστοίχισης ετικέτας-λέξης: κάθε ετικέτα ενσωματώνεται στον ίδιο χώρο με το διάνυσμα λέξης. Εισάγουν ένα πλαίσιο προσοχής που μετρά τη συμβατότητα των ενσωματώσεων μεταξύ ακολουθιών κειμένου και ετικετών μέσω της ομοιότητας συνημίτονου, όπως φαίνεται στην Εικόνα 35 (β).



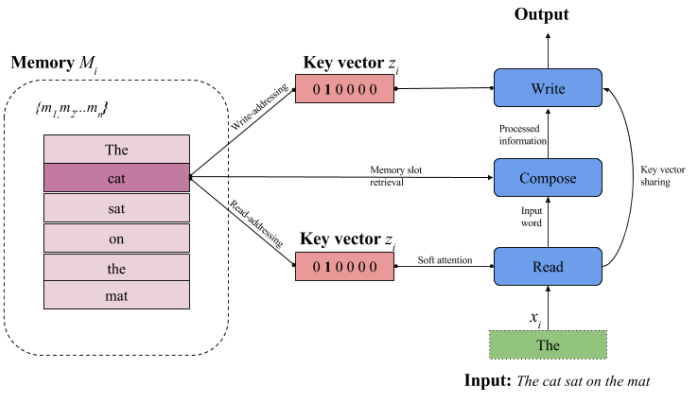
Εικόνα 35 : (a) The architecture of attentive pooling networks. (b) The architecture of lthe label-text matching model

Πηγή: https://www.researchgate.net/figure/a-The-architecture-of-attentive-pooling-networks-64-b-The-architecture-of_fig8_340523298

Οι Kim κ.α. [19] προτείνουν μια προσέγγιση σημασιολογικής αντιστοίχισης προτάσεων με τη χρήση ενός πυκνά συνδεδεμένου αναδρομικού και συνυπολογιστικού δικτύου. Κάθε στρώμα αυτού του μοντέλου χρησιμοποιεί συνδεδεμένες πληροφορίες από προσεκτικά χαρακτηριστικά καθώς και κρυφά χαρακτηριστικά όλων των προηγούμενων αναδρομικών στρωμάτων. Αυτό επιτρέπει τη διατήρηση των αρχικών και τις πληροφορίες συν-προσεγγιστικών χαρακτηριστικών από το πιο κάτω στρώμα ενσωμάτωσης λέξεων στο πιο πάνω επαναλαμβανόμενο στρώμα.

3.2.6 Memory-augmented networks για Ταξινόμηση Κειμένων

Ο Munkhdalai και ο Yu παρουσίασαν ένα νευρωνικό δίκτυο με επαυξημένη μνήμη, το οποίο ονομάζεται Νευρωνικός Σημασιολογικός Κωδικοποιητής (Neural Semantic Encoder, NSE), για ΤΣ. Το NSE διαθέτει μια επεκτάσιμη μνήμη κωδικοποίησης, η οποία εξελίσσεται στη διάρκεια του χρόνου και διατηρεί μια κατανόηση των ακολουθιών εισόδου μέσω διαφόρων λειτουργιών όπως η ανάγνωση, η σύνθεση και η εγγραφή, όπως φαίνεται στην Εικόνα 36.



Εικόνα 36 : NSE

Πηγή : https://paperswithcode.com/paper/neural-semantic-encoders

Ύστερα οι Weston κ.ά.[20] σχεδίασαν ένα δίκτυο μνήμης για μια συνθετική εργασία QA, όπου μια σειρά δηλώσεων (καταχωρήσεις μνήμης) παρέχονται στο μοντέλο ως υποστηρικτικά γεγονότα στην ερώτηση. Το μοντέλο μαθαίνει να ανακτά μία καταχώρηση κάθε φορά από τη μνήμη με βάση την ερώτηση και την προηγούμενη ανάκτηση μνήμης.

Οι Sukhbaatar κ.ά.[21] επεκτείναν αυτή την εργασία και προτείναν δίκτυα μνήμης από άκρο σε άκρο, όπου οι καταχωρήσεις μνήμης ανακτώνται με ήπιο τρόπο με μηχανισμό προσοχής, επιτρέποντας έτσι την εκπαίδευση τους. Δείχνουν ότι με πολλαπλούς γύρους (hops), το μοντέλο είναι σε θέση να ανακτήσει και να αιτιολογήσει πολλά υποστηρικτικά γεγονότα για να απαντήσει σε μια συγκεκριμένη ερώτηση.

Οι Kumar κ.ά. [22] προτείναν το Dynamic Memory Network (DMN), το οποίο επεξεργάζεται ακολουθίες εισόδου και ερωτήσεις, σχηματίζει επεισοδιακές μνήμες και παράγει σχετικές απαντήσεις. Οι ερωτήσεις ενεργοποιούν μια επαναληπτική διαδικασία προσοχής, η οποία επιτρέπει στο μοντέλο να εξαρτά την προσοχή του από τις εισόδους και το αποτέλεσμα προηγούμενων επαναλήψεων. Αυτά τα αποτελέσματα στη συνέχεια αιτιολογούνται σε ένα ιεραρχικό μοντέλο επαναλαμβανόμενων ακολουθιών για τη δημιουργία απαντήσεων. Το DMN εκπαιδεύεται από άκρη σε άκρη και επιτυγχάνει αποτελέσματα τελευταίας τεχνολογίας για την ποιοτική αξιολόγηση και την επισήμανση POS (POS tagging).

3.3 Τρόποι αξιολόγησης αυτών των εφαρμογών

Η απάντηση στο ερώτημα "ποια είναι η καλύτερη αρχιτεκτονική νευρωνικού δικτύου για την ΤΚ;" ποικίλλει σε μεγάλο βαθμό ανάλογα με τη φύση της εργασίας-στόχου και του τομέα, τη διαθεσιμότητα των ετικετών εντός του τομέα, τους περιορισμούς καθυστέρησης και χωρητικότητας των εφαρμογών κ.ο.κ. Αν και δεν υπάρχει αμφιβολία ότι η ανάπτυξη ενός ταξινομητή κειμένου είναι μια διαδικασία δοκιμής και λάθους, αναλύοντας πρόσφατα αποτελέσματα σε δημόσια benchmarks, προτείνουμε την ακόλουθη μέθοδο για να κάνουμε τη διαδικασία ευκολότερη. Η μέθοδος αποτελείται από πέντε βήματα:

1. **Επιλογή PLM**. Η χρήση PLM οδηγεί σε σημαντικές βελτιώσεις σε όλες τις δημοφιλείς εργασίες ταξινόμησης κειμένου και τα PLM με αυτόματη κωδικοποίηση (π.χ. BERT ή RoBERTa) συχνά λειτουργούν καλύτερα από τα PLM με αυτόματη παλινδρόμηση (π.χ. OpenAI GPT). Το Hugging Face3 διατηρεί ένα πλούσιο αποθετήριο PLMs που αναπτύχθηκαν για διάφορες εργασίες και ρυθμίσεις.
2. **Προσαρμογή στον τομέα**. Οι περισσότερες PLMs εκπαιδεύονται σε σώματα κειμένου γενικού τομέα (π.χ. Web). Εάν ο τομέας-στόχος διαφέρει δραματικά από τον γενικό τομέα, θα μπορούσαμε να εξετάσουμε το ενδεχόμενο να προσαρμόσουμε το PLM χρησιμοποιώντας δεδομένα εντός του τομέα, με συνεχή προ-εκπαίδευση του επιλεγμένου PLM γενικού τομέα. Για τομείς με άφθονο μη επισημασμένο κείμενο, όπως η βιοϊατρική, η προ-εκπαίδευση γλωσσικών μοντέλων από το μηδέν είναι μια καλή επιλογή.
3. **Σχεδιασμός μοντέλων για συγκεκριμένες εργασίες.** Αφού δοθεί το κείμενο εισόδου, το PLM παράγει μια ακολουθία διανυσμάτων στην αναπαράσταση συμφραζομένων. Στη συνέχεια, στην κορυφή προστίθενται ένα ή περισσότερα επίπεδα ειδικά για την εργασία, για να παραχθεί η τελική έξοδος για την εργασία-στόχο. Η επιλογή της αρχιτεκτονικής των ειδικών για την εργασία στρωμάτων εξαρτάται από τη φύση της εργασίας, π.χ. πρέπει να αποτυπωθεί η γλωσσική δομή του κειμένου. Όπως περιεγράφηκε στην ενότητα 2, τα νευρωνικά δίκτυα feed-forward βλέπουν το κείμενο ως μια σακούλα λέξεων, τα RNN μπορούν να συλλάβουν τη σειρά των λέξεων, τα CNN είναι καλά στην αναγνώριση μοτίβων όπως οι φράσεις-κλειδιά, οι μηχανισμοί προσοχής είναι αποτελεσματικοί για τον εντοπισμό συσχετιζόμενων λέξεων στο κείμενο, τα Siamese NN χρησιμοποιούνται για εργασίες αντιστοίχισης κειμένου και τα GNN μπορεί να είναι μια καλή επιλογή εάν οι δομές γράφων της φυσικής γλώσσας (π.χ. δέντρα ανάλυσης) είναι χρήσιμες για την εργασία-στόχο.
4. **Λεπτομερής ρύθμιση (fine-tuning) για συγκεκριμένη εργασία**. Ανάλογα με τη διαθεσιμότητα των ετικετών εντός του τομέα, τα ειδικά για την εργασία

επίπεδα μπορούν είτε να εκπαιδευτούν μόνα τους με σταθερό το PLM είτε να εκπαιδευτούν μαζί με το PLM. Εάν πρέπει να κατασκευαστούν πολλαπλοί παρόμοιοι ταξινομητές κειμένου (π.χ. ταξινομητές ειδήσεων για διαφορετικούς τομείς), η λεπτομερής ρύθμιση πολλαπλών εργασιών [23] είναι μια καλή επιλογή για την αξιοποίηση επισημασμένων δεδομένων παρόμοιων τομέων.

5. **Συμπίεση μοντέλου**. Η εξυπηρέτηση των PLM είναι δαπανηρή. Συχνά πρέπει να συμπιέζονται μέσω π.χ. της απόσταξης γνώσης για να ανταποκρίνονται στους περιορισμούς καθυστέρησης και χωρητικότητας στις εφαρμογές του πραγματικού κόσμου.

Παρακάτω περιγράφουμε ένα σύνολο από μετρήσεις που χρησιμοποιούνται συνήθως για την αξιολόγηση των επιδόσεων των μοντέλων TC και στη συνέχεια παρουσιάζουμε μια ποσοτική ανάλυση των επιδόσεων ενός συνόλου μοντέλων TC που βασίζονται σε DL σε δημοφιλείς δείκτες αναφοράς.

3.3.1 Confusion Matrix

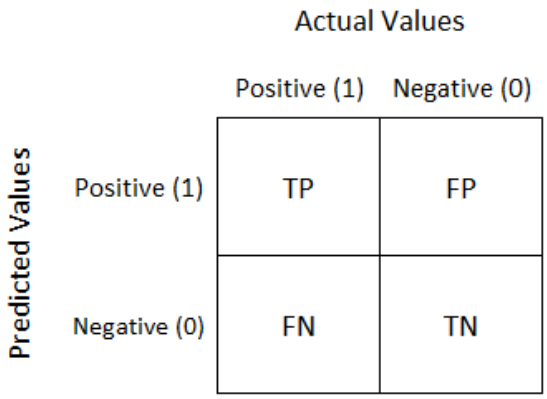
Για την επίτευξη των μετρήσεων της αποτελεσματικότητας ενός μοντέλου, χρησιμοποιούνται διάφορες τεχνικές. Μια από αυτές είναι ο πίνακας σύγχυσης (Confusion Matrix). Ο Πίνακας Σύγχυσης μετράει την αποτελεσματικότητα σε ένα πρόβλημα ταξινόμησης (Classification Problem) Μηχανικής Μάθησης όπου το αποτέλεσμα μπορεί να είναι δύο ή περισσότερες κλάσεις (labels). Κάθε γραμμή του πίνακα αναπαριστά τις περιπτώσεις σε μια πραγματική κλάση, ενώ κάθε στήλη αναπαριστά τις περιπτώσεις σε μια προβλεπόμενη κλάση, ή το αντίστροφο. Ο πίνακας σύγχυσης για ένα πρόβλημα με δύο κλάσεις αποτυπώνεται στην παρακάτω Εικόνα 37.

• TP ή αλλιώς True Positive: Το μοντέλο πρόβλεψε θετική τιμή και είναι θετική η πραγματική του τιμή.

• TN ή αλλιώς True Negative: Το μοντέλο πρόβλεψε αρνητική τιμή και είναι αρνητική η πραγματική του τιμή.

• FP ή αλλιώς False Positive: Το μοντέλο πρόβλεψε θετική τιμή, αλλά η πραγματική τιμή του είναι αρνητική. Το παραπάνω σφάλμα ονομάζεται και Τύπου 1 (Type 1 Error).

• FN ή αλλιώς False Negative: Το μοντέλο πρόβλεψε αρνητική τιμή, και η πραγματική τιμή του είναι θετική. Το παραπάνω σφάλμα ονομάζεται και Τύπου 2 (Type 2 Error).



Εικόνα 37: Confusion Matrix

Πηγή: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

3.3.2 Accuracy and Error Rate.

Αυτοί είναι οι κύριοι δείκτες μέτρησης για την αξιολόγηση της ποιότητας ενός μοντέλου ταξινόμησης. Έστω TP, FP, TN, FN που υποδηλώνουν το αληθώς θετικό, το ψευδώς θετικό, το αληθώς αρνητικό και το ψευδώς αρνητικό, αντίστοιχα. Η ακρίβεια ταξινόμησης και το ποσοστό σφάλματος ορίζονται στην παρακάτω εξίσωση (Εικόνα 38).

$$\textbf{Accuracy} = \frac{(TP + TN)}{N}, \quad \textbf{Error rate} = \frac{(FP + FN)}{N},$$

Εικόνα 38 : Accuracy and Error rate

όπου N είναι ο συνολικός αριθμός δειγμάτων.

Προφανώς έχουμε, Error Rate = 1 - Accuracy

3.3.3 Precision / Recall / F1 score.

Αυτές είναι επίσης κύριες μετρήσεις και χρησιμοποιούνται συχνότερα από την ακρίβεια ή το ποσοστό σφάλματος για μη ισορροπημένα σύνολα δοκιμών, όταν, π.χ., η πλειοψηφία των δειγμάτων δοκιμής έχει μία ετικέτα κλάσης. Σκοπός της ανάκλησης, είναι να βρίσκει από το σύνολο των θετικών, τι ποσοστό προβλέπεται ως θετικό. Το συγκεκριμένο μέτρο αξιολόγησης, χρησιμοποιείται όταν υπάρχει μεγάλο κόστος για Τύπου 2 σφάλματα (Type 2 Error). Η ειδοποιός διαφορά της ακρίβειας με την ανάκληση, είναι ότι με το Precision, υπολογίζουμε πόσο ακριβές είναι το μοντέλο, δηλαδή από το σύνολο των θετικών προβλέψεων, ποιο ποσοστό είναι πραγματικά θετικό. Χρησιμοποιείται συνήθως όταν το κόστος ενός Τύπου 1 Λάθους (Type 1 Error),

είναι υψηλό. Η ακρίβεια και η ανάκληση για δυαδική ταξινόμηση ορίζονται στην Εικόνα 39.

Η βαθμολογία F1 είναι ο αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης, χρησιμοποιείται όταν χρειάζεται μια ισορροπία μεταξύ Precision και Recall ή όταν οι κλάσεις δεν είναι ομοιόμορφα κατανεμημένες. Ο τύπος υπολογισμού F1, αποδίδεται στην Εικόνα 39.

$$\textbf{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \textbf{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \textbf{F1-score} = \frac{2\ \text{Prec}\ \text{Rec}}{\text{Prec} + \text{Rec}}$$

Εικόνα 39: Precision, Recall ,and F1-score

Η βαθμολογία F1 επιτυγχάνει την καλύτερη τιμή της στο 1 υποδεικνύοντας τέλεια ακρίβεια και ανάκληση, και τη χειρότερη στο 0, εάν είτε η ακρίβεια είτε η ανάκληση είναι μηδενικές.

3.3.4 Exact Match (EM).

Η μετρική της ακριβούς αντιστοίχισης είναι μια δημοφιλής μετρική για συστήματα απάντησης ερωτήσεων, η οποία μετρά το ποσοστό των προβλέψεων που ταιριάζουν ακριβώς με οποιαδήποτε από τις απαντήσεις της βασικής αλήθειας. Η "EM" είναι μία από τις βασικές μετρικές που χρησιμοποιούνται για το SQuAD, ένα σύνολο δεδομένων που περιλαμβάνει ερωτήσεις που τίθενται από crowdworkers σχετικά με άρθρα της Wikipedia. Η "EM" αναφέρεται στο ποσοστό των περιπτώσεων στις οποίες η απάντηση που δίνεται από το σύστημα είναι ακριβώς ίδια με την απάντηση που δίνεται στο σύνολο των δεδομένων.

3.3.5 Mean Reciprocal Rank (MRR).

Το MRR χρησιμοποιείται συχνά για την αξιολόγηση της απόδοσης των αλγορίθμων κατάταξης σε εργασίες NLP, όπως η κατάταξη ερωτημάτων-εργαλείων και QA. Το MRR ορίζεται στην Εικόνα 40, όπου Q είναι ένα σύνολο όλων των πιθανών απαντήσεων, και $rank_i$ είναι η θέση κατάταξης της βασικής αληθινής απάντησης.

$$\textbf{MRR} = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{rank_i}.$$

Εικόνα 40: Mean Reciprocal Rank (MRR)

Άλλες ευρέως χρησιμοποιούμενες μετρικές περιλαμβάνουν τη Mean Average Precision (MAP), Area Under Curve (AUC), False Discovery Rate, False Omission Rate.

4 *Επεξηγηματικά μοντέλα*

4.1 Explainability

Η βαθιά μάθηση έχει αποτελέσει σημαντική πρόοδο στην τεχνητή νοημοσύνη τα τελευταία χρόνια. Αντίθετα με παραδοσιακές μεθόδους μηχανικής μάθησης, όπως οι αλγόριθμοι απόφασης και οι μηχανές διανυσμάτων υποστήριξης (support vector), οι μέθοδοι βαθιάς μάθησης έχουν επιτύχει σημαντική βελτίωση σε διάφορες εργασίες πρόβλεψης, όπως στην επεξεργασία εικόνας και στην επεξεργασία φυσικής γλώσσας.

Ωστόσο, τα βαθιά νευρωνικά δίκτυα (DNN) είναι συγκριτικά αδύναμα στην εξήγηση των διαδικασιών εξαγωγής συμπερασμάτων και των τελικών αποτελεσμάτων τους. Οι τεχνολογίες που χρησιμοποιούνται συχνά θεωρούνται αδιαπραγμάτευτες και δεν είναι εύκολο να κατανοηθούν από τους προγραμματιστές ή τους χρήστες, και επομένως θεωρούνται ως "μαύρα κουτιά". Ορισμένοι μάλιστα θεωρούν τα DNN (βαθιά νευρωνικά δίκτυα) στο σημερινό στάδιο μάλλον ως αλχημεία, παρά ως πραγματική επιστήμη. Σε πολλές εφαρμογές του πραγματικού κόσμου, όπως η βελτιστοποίηση διαδικασιών, η ιατρική διάγνωση και η εισήγηση επενδύσεων, η επεξηγηματικότητα και η διαφάνεια των συστημάτων τεχνητής νοημοσύνης καθίστανται ιδιαίτερα σημαντικές για τους χρήστες τους, για τους ανθρώπους που επηρεάζονται από τις αποφάσεις της τεχνητής νοημοσύνης και επιπλέον για τους ερευνητές και τους προγραμματιστές που δημιουργούν τις λύσεις τεχνητής νοημοσύνης. Τα τελευταία χρόνια, η επεξηγησιμότητα και η επεξηγήσιμη ΤΝ έχουν λάβει όλο και μεγαλύτερη προσοχή τόσο από την ερευνητική κοινότητα όσο και από τη βιομηχανία.

Οι διαφορετικές κατηγορίες δείχνουν διαφορετικές διαστάσεις στην έρευνα για την ερμηνευσιμότητα, από προσεγγίσεις που παρέχουν "προφανώς" ερμηνεύσιμες πληροφορίες μέχρι τις μελέτες περίπλοκων προτύπων. Εφαρμόζοντας την ίδια κατηγοριοποίηση στην ερμηνευσιμότητα στην ιατρική έρευνα, ελπίζουμε ότι :

1) οι κλινικοί ιατροί και οι επαγγελματίες μπορούν στη συνέχεια να προσεγγίσουν αυτές τις μεθόδους με προσοχή

2) η γνώση στην ερμηνευσιμότητα να γεννηθεί με περισσότερες εκτιμήσεις για τις ιατρικές πρακτικές

3) να ενθαρρύνονται πρωτοβουλίες στην ιατρική εκπαίδευση βασισμένες σε δεδομένα μαθηματικά και τεχνικά ορθές.

Ο κύριος λόγος για τον οποίο χρειαζόμαστε τα επεξηγηματικά μοντέλα στον τομέα του AI είναι για να εξηγήσουμε το μοντέλο το οποίο χρησιμοποιούμε, αλλά και να κάνουμε κατανοητό το αποτέλεσμα του αλγόριθμου στο χρήστη.

Υπάρχουν δύο βασικές κατηγορίες επεξηγηματικών μοντέλων, αυτά που χρησιμοποιούνται σε ήδη εκπαιδευμένα μοντέλα (Post-hoc) και τα Εγγενή.

Post-hoc: Ονομάζουμε την επεξηγηματική μας μέθοδο post-hoc, όταν εξηγούμε ένα μοντέλο αφού έχουν γίνει οι προβλέψεις ή αφού έχει εκπαιδευτεί το μοντέλο. Αυτό είναι σημαντικό, επειδή αυτές οι μέθοδοι μας επιτρέπουν να εξηγήσουμε ιδιαίτερα περίπλοκα μοντέλα. Αποτελεί μια δημοφιλή προσέγγιση για την αντιμετώπιση του προβλήματος της αδιαφάνειας των μοντέλων μηχανικής μάθησης "μαύρου κουτιού". Ωστόσο, αυτές οι μέθοδοι μπορούν να ξεγελαστούν, είναι ελαττωματικές υπό συγκεκριμένες συνθήκες και απαιτούν ένα επιπλέον επίπεδο πολυπλοκότητας για τη δημιουργία των εξηγήσεων. Κοινά παραδείγματα είναι τα πακέτα LRP, SHAP και LIME.

Εγγενή: Ορισμένα μοντέλα μπορούν να εξηγηθούν κατευθείαν, χωρίς να τοποθετηθούν επιπλέον μοντέλα ή βιβλιοθήκες από πάνω. Αυτά είναι συνήθως απλούστερα και σε ορισμένες περιπτώσεις μπορεί να έχουν μικρότερη ισχύ πρόβλεψης, αν και ορισμένοι ερευνητές υποστηρίζουν ότι αυτό δεν ισχύει πάντα. Κοινά παραδείγματα είναι τα εξής: Generalised Linear Models, δέντρα αποφάσεων (Decision Trees), Monotonic Gradient Boosting, TabNet.

4.1.1 Δέντρα αποφάσεων (Decision Trees)

Τα δέντρο αποφάσεων είναι ένας supervised learning τύπος αλγορίθμου μηχανικής μάθησης που χρησιμοποιείται συνήθως για εργασίες ταξινόμησης και παλινδρόμησης. Πρόκειται για μια δομή που μοιάζει με διάγραμμα ροής, όπου κάθε εσωτερικός κόμβος αντιπροσωπεύει ένα χαρακτηριστικό ή γνώρισμα και κάθε κλάδος αντιπροσωπεύει έναν κανόνα απόφασης ή ένα αποτέλεσμα. Τα φύλλα του δέντρου αντιπροσωπεύουν τις ετικέτες κλάσης ή τα αποτελέσματα παλινδρόμησης.

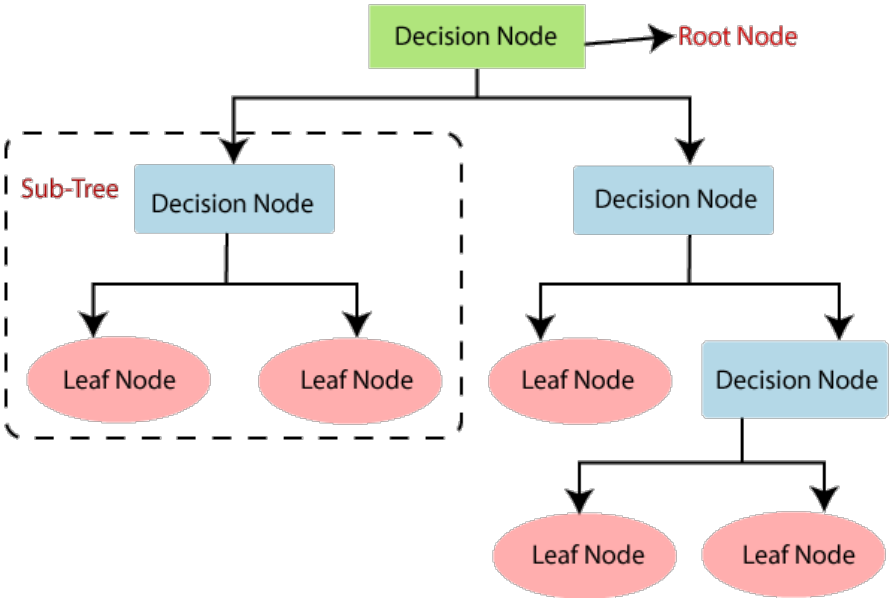
Η κατασκευή ενός δέντρου αποφάσεων περιλαμβάνει την επιλογή του καλύτερου χαρακτηριστικού σε κάθε κόμβο με βάση ένα συγκεκριμένο κριτήριο. Η διαδικασία αυτή επαναλαμβάνεται αναδρομικά μέχρι να ικανοποιηθεί ένα κριτήριο διακοπής, όπως η επίτευξη ενός μέγιστου βάθους ή ενός ελάχιστου αριθμού περιπτώσεων ανά κόμβο φύλλου.

Τα δέντρα αποφάσεων έχουν πολλά πλεονεκτήματα, όπως ότι είναι εύκολο να ερμηνευτούν και να απεικονιστούν και ότι μπορούν να χειριστούν τόσο κατηγορικά όσο και αριθμητικά δεδομένα. Μπορούν επίσης να χειριστούν ελλείπουσες τιμές και ακραίες τιμές και είναι ανθεκτικά σε άσχετα χαρακτηριστικά.

Ωστόσο, τα δέντρα αποφάσεων μπορούν επίσης να υποφέρουν από διάφορους περιορισμούς, όπως η υπερβολική προσαρμογή στα δεδομένα εκπαίδευσης και η ευαισθησία σε μικρές μεταβολές των δεδομένων. Για την αντιμετώπιση αυτών των ζητημάτων, έχουν αναπτυχθεί διάφορες μέθοδοι συνόλου, όπως τα Random Forest και το Boosting.

Συνοπτικά, τα δέντρα αποφάσεων είναι ένας δημοφιλής και ευέλικτος αλγόριθμος μηχανικής μάθησης που μπορεί να χρησιμοποιηθεί για ένα ευρύ φάσμα εργασιών. Έχουν τόσο πλεονεκτήματα όσο και περιορισμούς, αλλά μπορούν να βελτιωθούν μέσω διαφόρων τεχνικών και επεκτάσεων.

Το παρακάτω διάγραμμα εξηγεί τη γενική δομή ενός δέντρου αποφάσεων:



Εικόνα 41 : Δέντρο αποφάσεων

Πηγή: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

Το σπουδαίο με τα Δέντρα Αποφάσεων είναι ότι μπορούμε κυριολεκτικά να εξάγουμε ολόκληρο το δέντρο και να παρακολουθήσουμε γιατί το μοντέλο έκανε μια πρόβλεψη για οποιοδήποτε δείγμα στο σύνολο δεδομένων.

Μόλις τα δέντρα γίνουν πολύ μεγάλα (max_depth = 7+), γίνεται δύσκολο να τα παρακολουθήσει ο άνθρωπος λόγω της εκθετικής αύξησης του αριθμού των φύλλων. Ωστόσο, σε αυτό το σημείο θα μπορούσαμε ακόμα να γράψουμε κάποιο βασικό κώδικα για να αναδείξουμε τη διαδρομή που ακολούθησαν τα

δεδομένα μας για να φτάσουν στην πρόβλεψή τους, καθώς και να δοκιμάσουμε πώς θα αντιδράσει το μοντέλο σε αόρατα δεδομένα.

Τα δέντρα αποφάσεων θεωρούνται ότι είναι ένα από τα πιο επεξηγήσιμα μοντέλα λόγω των πολύ λίγων μαθηματικών. Παράλληλα αντιμετωπίζουν δυσκολίες στην εξήγηση των διαδικασιών εξαγωγής συμπερασμάτων και των τελικών αποτελεσμάτων τους όταν χρησιμοποιούνται σε πιο σύνθετα μοντέλα που δεν έχουν εκπαιδευτεί, με αποτέλεσμα να μην είναι από τα ιδανικότερα μοντέλα για νευρωνικά δίκτυα, όπως το BERT. Γι' αυτόν τον λόγο θα ήταν προτιμότερο να χρησιμοποιήσουμε ένα Post-hoc μοντέλο όπως το SHAP ή το LIME .

4.1.2 LRP (Layer-wise Relevance Propagation)

Η αντίστροφη ανάδραση προς τα πίσω (Layer-wise Relevance Propagation - LRP) είναι μια τεχνική που χρησιμοποιείται στη βαθιά μάθηση για την εξήγηση των προβλέψεων που κάνουν τα νευρωνικά δίκτυα. Είναι μια μέθοδος για την απόδοση της πρόβλεψης ενός νευρωνικού δικτύου στα χαρακτηριστικά εισόδου ή στα εισαγόμενα pixels μέσω της διάδοσης της σημασίας της πρόβλεψης προς τα πίσω μέσα από το δίκτυο.

Ο αλγόριθμος του LRP λειτουργεί με αντίστροφη διάδοση των βαθμολογιών σημασίας μέσω των επιπέδων του δικτύου. Σε κάθε επίπεδο, οι βαθμολογίες σημασίας επανακατανέμονται στα χαρακτηριστικά εισόδου που συνέβαλαν στην έξοδο εκείνου του επιπέδου, βάσει ενός συνόλου κανόνων που στοχεύουν να αποτυπώσουν τη σημασία κάθε χαρακτηριστικού εισόδου για την τελική πρόβλεψη.

Στόχος του LRP είναι να παρέχει μια διαφανή και ερμηνεύσιμη εξήγηση για τις προβλέψεις που κάνει ένα νευρωνικό δίκτυο, αναδεικνύοντας τα χαρακτηριστικά εισόδου που ήταν πιο σημαντικά για την πρόβλεψη. Αυτό μπορεί να είναι χρήσιμο για την κατανόηση της συμπεριφοράς του δικτύου, την ανίχνευση σημαντικών χαρακτηριστικών στα δεδομένα εισόδου και την ανίχνευση δυνητικών προκαταλήψεων ή σφαλμάτων στις προβλέψεις του δικτύου.

Το LRP έχει εφαρμοστεί σε μια ευρεία γκάμα από εργασίες στη βαθιά μάθηση, συμπεριλαμβανομένης της αναγνώρισης εικόνων, της επεξεργασίας φυσικής γλώσσας και της αναγνώρισης ομιλίας, και έχει αποδειχθεί ότι παρέχει ακριβείς και σημαντικές εξηγήσεις για τις προβλέψεις που κάνουν τα νευρωνικά δίκτυα.

4.1.3 SHAP

Το SHAP είναι μια μέθοδος εξήγησης μηχανικής μάθησης που βασίζεται στη θεωρία παιγνίων και χρησιμοποιεί τις κλασικές τιμές Shapley για να συνδέσει τη βέλτιστη κατανομή πιστώσεων με τοπικές εξηγήσεις. Η μέθοδος αυτή μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο μηχανικής μάθησης και παρέχει σημαντικές πληροφορίες σχετικά με το πώς λαμβάνονται οι αποφάσεις του μοντέλου και ποιος είναι ο συνεισφέρων παράγοντας σε κάθε πρόβλεψη. Ακόμη αξιοποιεί την ιδέα των τιμών Shapley για τη βαθμολόγηση της επιρροής των χαρακτηριστικών του μοντέλου. Ο τεχνικός ορισμός μιας τιμής Shapley είναι η "μέση οριακή συνεισφορά μιας τιμής χαρακτηριστικού σε όλους τους δυνατούς συνασπισμούς". Με άλλα λόγια, οι τιμές Shapley εξετάζουν όλες τις πιθανές προβλέψεις για ένα παράδειγμα χρησιμοποιώντας όλους τους πιθανούς συνδυασμούς εισόδων. Λόγω αυτής της εξαντλητικής προσέγγισης, η SHAP μπορεί να εγγυηθεί ιδιότητες όπως η συνέπεια και η τοπική ακρίβεια.

Για να μπορέσουμε να κατανοήσουμε περεταίρω την λειτουργία του μοντέλου SHAP, αρχικά θα αναλύσουμε την αρχιτεκτονική και τον τρόπο λειτουργείας του μοντέλου και ύστερα θα περιγράψουμε μερικά από τα πιθανά διαγράμματα που χρησιμοποιούνται έτσι ώστε να αποκαλύψουμε την ατομική συμβολή κάθε χαρακτηριστικού στην έξοδο του μοντέλου.

4.1.3.1 Τρόπος λειτουργείας του μοντέλου

Οι μέθοδος αυτού του μοντέλου εστιάζει στις εξηγήσεις για ένα μοντέλο f σε κάποιο μεμονωμένο σημείο x∗. Βασίζονται σε μια συνάρτηση τιμής eS όπου S είναι ένα υποσύνολο από δείκτες μεταβλητών S ⊆ {1, ..., p}. Συνήθως, η συνάρτηση αυτή ορίζεται ως η αναμενόμενη τιμή για μια υπό συνθήκη κατανομή στην οποία ισχύει η συνθήκη σε όλες τις μεταβλητές σε ένα υποσύνολο του S [24]

$$e_S = E[f(x)|x_S = x_S^*].$$

Εικόνα 42: Αναμενόμενη Τιμή

Η αναμενόμενη τιμή χρησιμοποιείται συνήθως για δεδομένα σε πίνακες. Αντίθετα, για άλλες μορφές δεδομένων, η συνάρτηση αυτή ορίζεται επίσης συχνά ως η πρόβλεψη του μοντέλου στο x∗ μετά το μηδενισμό των τιμών των μεταβλητών με δείκτες εκτός του S. Όποιος και αν είναι ο ορισμός που χρησιμοποιείται, η τιμή της eS μπορεί να θεωρηθεί ως η απόκριση του μοντέλου μόλις καθοριστούν οι μεταβλητές στο υποσύνολο S. Ο σκοπός της απόδοσης

είναι να αποσυνθέσει τη διαφορά f(x∗) - eØ σε μέρη που μπορούν να αποδοθούν σε μεμονωμένες μεταβλητές.

Ειδικότερα, η εκτίμηση της σημασίας της μεταβλητής i βασίζεται στην ανάλυση του τρόπου με τον οποίο η προσθήκη της μεταβλητής i στο σύνολο S θα επηρεάσει την τιμή της συνάρτησης eS. Η συμβολή μιας μεταβλητής i συμβολίζεται με φ(i) και υπολογίζεται ως σταθμισμένος μέσος όρος σε όλες τις πιθανά υποσύνολα S.

$$\phi(i) = \sum_{S \subseteq \{1,\dots,p\}/\{i\}} \frac{|S|!(p-1-|S|)!}{p!} \left(e_{S\cup\{i\}} - e_S\right).$$

Εικόνα 43 : Weighted average

Συνοπτικά, η ανάλυση μιας μόνο διάταξης δείχνει πώς η προσθήκη διαδοχικών μεταβλητών αλλάζει την τιμή της συνάρτησης eS. Το SHAP προκύπτει ως μέσος όρος αυτών των συνεισφορών σε όλες τις πιθανές διατάξεις. Αυτός ο αλγόριθμος είναι μια προσαρμογή των τιμών Shapley για την εξήγηση των μεμονωμένων προβλέψεων των μοντέλων μηχανικής μάθησης. Οι τιμές Shapley προτάθηκαν αρχικά για τη δίκαιη κατανομή των τιμών στα συνεργατικά παίγνια και είναι η μόνη λύση που βασίζεται στα αξιώματα της αποτελεσματικότητας, της συμμετρίας, της ψευδολογίας και της προσθετικότητας.

Εφόσον έχουμε αναλύσει την λογική του μοντέλου θα επεκταθούμε στην επεξήγηση της διαδικασίας που ακολουθεί το μοντέλο SHAP για την επεξήγηση ενός δείγματος είναι η εξής :

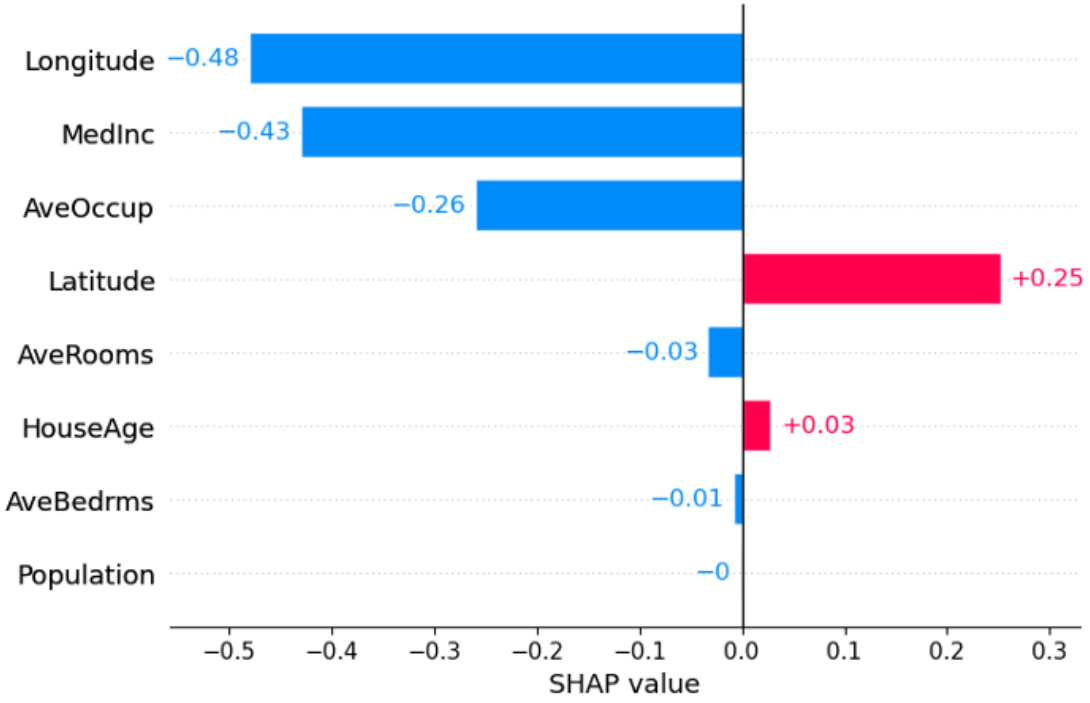
1. **Επιλογή δείγματος**: Επιλέγεται ένα συγκεκριμένο δείγμα ή παράδειγμα από το σύνολο δεδομένων που θέλουμε να εξηγήσουμε.
2. **Υπολογισμός πρόβλεψης βάσης**: Χρησιμοποιώντας του μοντέλου μας κάνουμε υπολογισμό πρόβλεψης χρησιμοποιώντας όλα τα attributes του συγκεκριμένου sample.
3. **Υπολογισμός πρόβλεψης αφαιρώντας την συνεισφορά ενός από τα attributes**: Για κάθε attribute, αφαιρούμαι την συνεισφορά του στην τελική πρόβλεψη του μοντέλου
    - Δίνονται τυχαίες τιμές μέσα από την κατανομή του συγκεκριμένου attribute.
    - Υπολογίζεται η πρόβλεψη για κάθε μία από τις τιμές που επιλέχθηκαν.

- Υπολογίζεται ο μέσος όρος πρόβλεψης από όλα τα παραπάνω αποτελέσματα.

4. **Υπολογισμός συνεισφοράς των Attribute**: Χρησιμοποιώντας τα αποτελέσματα του βήματος 3, υπολογίζουμε τις διαφορές μεταξύ πρόβλεψης βάσης, και πρόβλεψης μετά την αφαίρεση της συνεισφοράς για κάθε attribute. Οι υπολογιζόμενες τιμές για κάθε attribute είναι οι τιμές SHAP.
5. **Οπτικοποίηση των αποτελεσμάτων**: Παρουσίαση των τιμών SHAP χρησιμοποιώντας κατάλληλες απεικονίσεις για την ευρύτερη κατανόηση των επεξηγήσεων.

4.1.3.2 Waterfall Plot

Στο διάγραμμα «καταρράκτη» ο άξονας x έχει τις τιμές της μεταβλητής-στόχου (dependent). Ως x ορίζουμε την επιλεγμένη παρατήρηση, f(x) είναι η προβλεπόμενη τιμή του μοντέλου, δεδομένης της εισόδου x και E[f(x)] είναι η αναμενόμενη τιμή της μεταβλητής-στόχου ή, με άλλα λόγια, ο μέσος όρος όλων των προβλέψεων.



Εικόνα 44: Waterfall Plot

Πηγή: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137

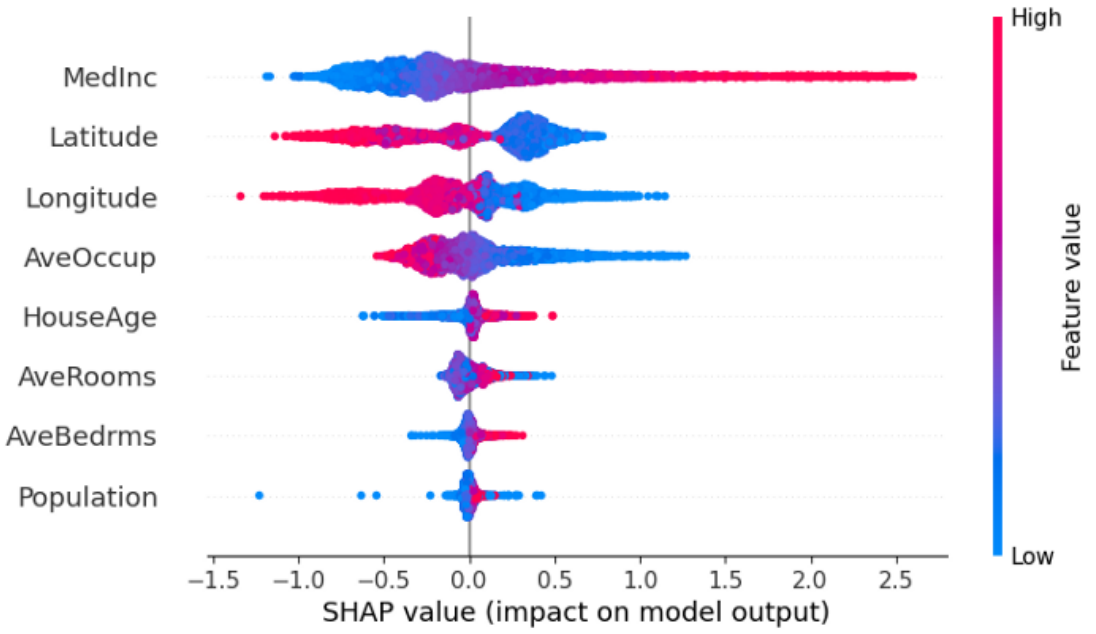
Η τιμή SHAP για κάθε χαρακτηριστικό σε αυτή την παρατήρηση δίνεται από το μήκος της ράβδου. Στο παραπάνω παράδειγμα, το Longitude έχει τιμή SHAP -0,48, το Latitude έχει τιμή SHAP +0,25 κ.ο.κ. Το άθροισμα όλων των τιμών SHAP θα είναι ίσο με E[f(x)] - f(x).

Η απόλυτη τιμή SHAP μας δείχνει πόσο ένα μεμονωμένο χαρακτηριστικό επηρέασε την πρόβλεψη, έτσι το Longitude συνέβαλε περισσότερο, το MedInc το δεύτερο, το AveOccup το τρίτο και το Population ήταν το χαρακτηριστικό με τη μικρότερη συμβολή στην πρόβλεψη.

Είναι πολύ σημαντικό να γνωρίζουμε ότι αυτές οι τιμές SHAP ισχύουν μόνο για αυτή την παρατήρηση. Με άλλα data points οι τιμές SHAP θα αλλάξουν. Για να γίνει κατανοητή η σημασία ή η συμβολή των χαρακτηριστικών για το σύνολο των δεδομένων, μπορεί να χρησιμοποιηθεί ένα άλλο διάγραμμα, το διάγραμμα bee swarm.

4.1.3.3 Bee Swarm Plot

Στο διάγραμμα Bee Swarm Plot τα χαρακτηριστικά είναι επίσης ταξινομημένα με βάση την επίδρασή τους στην πρόβλεψη, αλλά μπορούμε επίσης να δούμε πώς υψηλότερες και χαμηλότερες τιμές του χαρακτηριστικού θα επηρεάσουν το αποτέλεσμα.



Εικόνα 45: Bee Swarm plot

Πηγή: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137

Όλες οι μικρές κουκκίδες στο διάγραμμα αντιπροσωπεύουν μία μόνο παρατήρηση. Ο οριζόντιος άξονας αντιπροσωπεύει την τιμή SHAP, ενώ το

χρώμα του σημείου μας δείχνει αν η συγκεκριμένη παρατήρηση έχει υψηλότερη ή χαμηλότερη τιμή, σε σύγκριση με άλλες παρατηρήσεις.

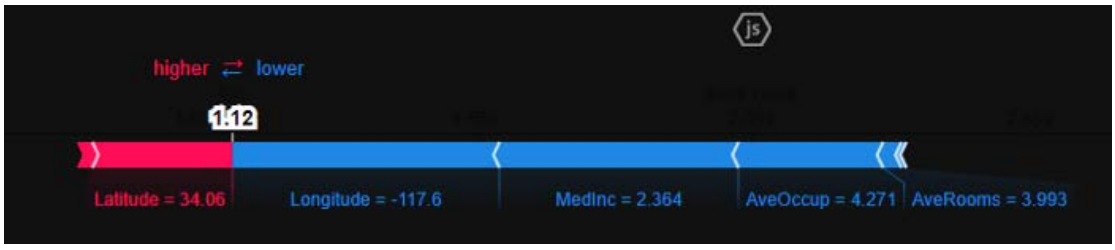
Για παράδειγμα, οι υψηλές τιμές της μεταβλητής Latitude έχουν υψηλή αρνητική συμβολή στην πρόβλεψη, ενώ οι χαμηλές τιμές έχουν υψηλή θετική συμβολή. Η μεταβλητή MedInc έχει πραγματικά υψηλή θετική συμβολή όταν οι τιμές της είναι υψηλές, και χαμηλή αρνητική συμβολή στις χαμηλές τιμές. Το χαρακτηριστικό Population δεν έχει σχεδόν καμία συμβολή στην πρόβλεψη, είτε οι τιμές του είναι υψηλές είτε χαμηλές.

Όλες οι μεταβλητές παρουσιάζονται με τη σειρά της συνολικής σημασίας του χαρακτηριστικού, με πιο σημαντική την πρώτη και λιγότερο σημαντική την τελευταία.

Ως αποτέλεσμα, το SHAP μπορεί να μας δείξει τόσο την συνολική (global) συνεισφορά με τη χρήση των εισαγωγών των χαρακτηριστικών, όσο και την τοπική (local) συνεισφορά των χαρακτηριστικών για κάθε περίπτωση του προβλήματος με τη διασπορά του διαγράμματος Bee Swarm.

4.1.3.4 Force plot

Το διάγραμμα δύναμης είναι ένας ακόμη τρόπος για να δούμε την επίδραση που έχει κάθε χαρακτηριστικό στην πρόβλεψη, για μια δεδομένη παρατήρηση. Σε αυτό το διάγραμμα οι θετικές τιμές SHAP εμφανίζονται στην αριστερή πλευρά και οι αρνητικές στη δεξιά πλευρά, σαν να ανταγωνίζονται η μία την άλλη. Η επισημασμένη τιμή είναι η πρόβλεψη για τη συγκεκριμένη παρατήρηση.

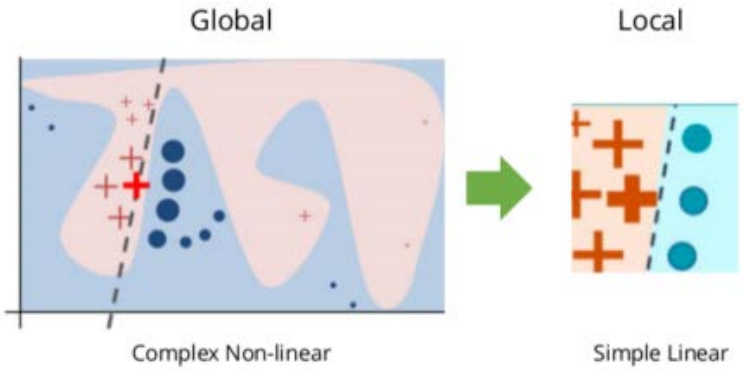


Πηγή: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137

Εικόνα 46: Force plot

4.1.4 LIME

Το LIME, αλλιώς γνωστό και ως local interpretable model-agnostic explanations, είναι μια τεχνική που προσεγγίζει οποιοδήποτε μοντέλο μηχανικής μάθησης μαύρου κουτιού (black box) με ένα τοπικό, ερμηνεύσιμο μοντέλο για την επεξήγηση κάθε μεμονωμένης πρόβλεψης δημιουργώντας γραμμικά μοντέλα γύρω από κάθε μια από αυτές. Η ιδέα προέρχεται από μια δημοσίευση του 2016 στην οποία οι συγγραφείς διαταράσσουν τα αρχικά σημεία δεδομένων, τα τροφοδοτούν στο μοντέλο μαύρου κουτιού και στη συνέχεια παρατηρούν τις αντίστοιχες εξόδους. Στη συνέχεια, η μέθοδος σταθμίζει αυτά τα νέα data points σε συνάρτηση με την προσέγγισή τους στο αρχικό σημείο. Τελικά, προσαρμόζει ένα υποκατάστατο μοντέλο, όπως η γραμμική παλινδρόμηση (linear regression), στο σύνολο δεδομένων με διακυμάνσεις χρησιμοποιώντας αυτά τα βάρη του δείγματος. Κάθε αρχικό σημείο δεδομένων μπορεί στη συνέχεια να εξηγηθεί με το πρόσφατα εκπαιδευμένο μοντέλο εξήγησης [25]. Δεν θα πρέπει να παραλείψουμε ότι όσο πιο απομακρυσμένο είναι το γειτονικό παράδειγμα από τον παράδειγμά μας, τόσο μικρότερη ισχύς έχει στην απόφαση του μοντέλου.



Εικόνα 47: LIME diagram

*Πηγή: https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/ *

- **Τεχνική προσέγγιση του μοντέλου**

Η βασική ιδέα του LIME είναι να εξηγήσουμε μια πρόβλεψη ενός σύνθετου μοντέλου fM, π.χ. ενός βαθιού νευρωνικού δικτύου, με την προσαρμογή ενός τοπικού υποκατάστατου μοντέλου fS, του οποίου οι προβλέψεις είναι εύκολο να εξηγηθούν. Ως εκ τούτου, το LIME αναφέρεται συχνά και ως τεχνική εξήγησης με βάση το υποκατάστατο. Τεχνικά, το LIME παράγει δείγματα στη γειτονιά Nxi της εισόδου που ενδιαφέρει xi, τα αξιολογεί χρησιμοποιώντας το μοντέλο-στόχο και στη συνέχεια προσεγγίζει το μοντέλο-στόχο σε αυτή την τοπική γειτονιά με μια απλή γραμμική συνάρτηση, δηλαδή ένα υποκατάστατο μοντέλο που είναι εύκολο να ερμηνευτεί. Έτσι, το LIME δεν

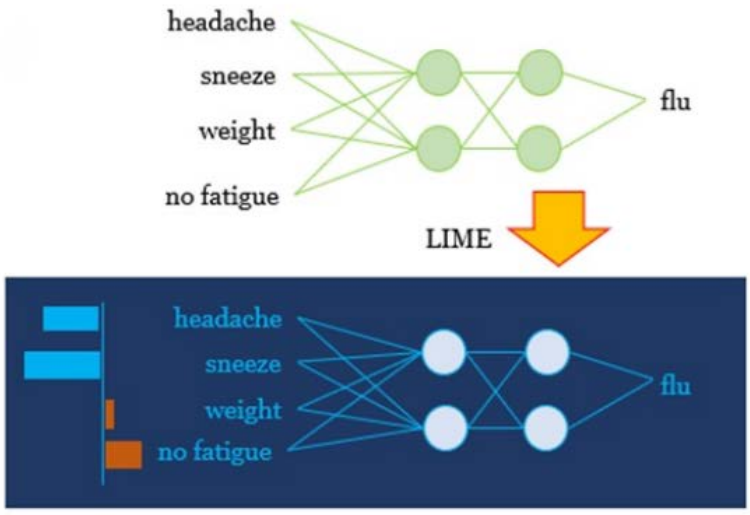
εξηγεί άμεσα την πρόβλεψη του μοντέλου-στόχου fM(xi), αλλά τις προβλέψεις ενός υποκατάστατου μοντέλου fS(xi), το οποίο προσεγγίζει τοπικά το μοντέλο-στόχο (δηλ, fM(x) ≈ fS(x) για x ∈ Nxi).

Πιο συγκεκριμένα, η γενική έκφραση για το δείγμα x με LIME παρουσιάζεται στην εξίσωση στην Εικόνα 48. Όπου G είναι μια κλάση δυνητικά ερμηνεύσιμων μοντέλων, g ∈ G είναι ένα μοντέλο που μπορεί να παρουσιαστεί στον χρήστη ως εξήγηση και Ω(g) είναι ένα μέτρο πολυπλοκότητας. Το L(f , g, π_x) ποσοτικοποιεί το πόσο αναξιόπιστο είναι το g στην προσέγγιση του εξηγημένου μοντέλου f στην τοποθεσία που ορίζεται από ένα μέτρο εγγύτητας μεταξύ μιας περίπτωσης z και ενός δείγματος x (π_x). Στόχος είναι η εξεύρεση ισορροπίας μεταξύ της ελαχιστοποίησης του L(f , g, π_x) και της διατήρησης των κατανοητών από τον άνθρωπο επιπέδων του Ω(g).

$$argmin_g \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Εικόνα 48 : Τύπος-LIME

Επομένως, το LIME υφίσταται έναν συμβιβασμό μεταξύ της ακρίβειας του μοντέλου και της πολυπλοκότητας.



Εικόνα 49: LIME

Πηγή: https://www.researchgate.net/figure/A1-Using-LIME-to-generate-explanation-for-text-classification-Headache-and-sneeze-are_fig2_334534363

Στην παραπάνω Εικόνα κάνουμε χρήση του LIME για τη δημιουργία εξηγήσεων για την ταξινόμηση κειμένων. Η λέξη headache και sneeze λαμβάνουν θετικές τιμές. Αυτό σημαίνει ότι και οι δύο παράγοντες έχουν θετική

συμβολή στη πρόβλεψη του μοντέλου για γρίπη. Από την άλλη πλευρά, η λέξη weight και no fatigue συμβάλλουν αρνητικά στην πρόβλεψη [26].

Επίσης οι τιμές Shapley παρέχουν τη μόνη εγγύηση ακρίβειας και συνέπειας και το LIME είναι στην πραγματικότητα ένα υποσύνολο του SHAP χωρίς όμως να έχει τις ίδιες ιδιότητες.

Το LIME δημιουργεί ένα προσαρμοσμένο σύνολο δεδομένων για την εφαρμογή ενός εξηγήσιμου μοντέλου, ενώ το SHAP απαιτεί ένα ολόκληρο δείγμα για τον υπολογισμό των τιμών SHAP. Αυτό σημαίνει ότι το LIME απαιτεί μόνο μία παρατήρηση, ενώ το SHAP απαιτεί πολλαπλές παρατηρήσεις.

4.1.4.1 Αναλυτική διαδικασία του μοντέλου

Η διαδικασία που ακολουθεί το LIME για την επεξήγηση ενός παραδείγματος είναι η εξής:

1. Αρχικά κάνει επιλογή ενός παραδείγματος που θέλει να επεξεργαστεί μέσα από ένα δείγμα παραδειγμάτων.
2. Δημιουργεί συνθετικά παραδείγματα τα οποία αποτελούν παραλλαγές του επιλεγμένου παραδείγματος τροποποιώντας τυχαία τις τιμές χαρακτηριστικών του.
3. Πραγματοποιείται ο υπολογισμός προβλέψεων για κάθε ένα από τα συνθετικά παραδείγματα χρησιμοποιώντας το βασικό μας μοντέλο.
4. Δημιουργείται ενα σετ απλών και εξηγήσιμων μοντέλων πρόβλεψης.
5. Επιλογή του καλύτερου από αυτά τα εξηγήσιμα μοντέλα βάση του κατά πόσο συμφωνούν οι προβλέψεις τους με το βασικό μοντέλο.
6. Χρήση του εξηγήσιμου μοντέλου πρόβλεψης για την επεξήγηση του συγκεκριμένου παραδείγματος από το οποίο αρχίσαμε.

4.2 LIME vs SHAP

Το LIME (Local Interpretable Model-Agnostic Explanations) και το SHAP (SHapley Additive exPlanations) είναι δύο δημοφιλείς μέθοδοι που χρησιμοποιούνται για την εξήγηση των προβλέψεων των μοντέλων μηχανικής μάθησης. Ενώ και οι δύο μέθοδοι στοχεύουν στην παροχή εξηγήσεων για τις προβλέψεις του μοντέλου, υπάρχουν ορισμένες διαφορές μεταξύ τους ως προς τα παρακάτω:

1) **Μεθοδολογία**: Το LIME και το SHAP χρησιμοποιούν διαφορετικές προσεγγίσεις για τη δημιουργία εξηγήσεων. Το LIME παράγει τοπικές προσεγγίσεις του μοντέλου εκπαιδεύοντας ερμηνεύσιμα μοντέλα, όπως γραμμικά μοντέλα, για τοπική επεξηγηματικότητα. Το SHAP, από την άλλη πλευρά, χρησιμοποιεί τη θεωρητική προσέγγιση παιγνίων για την απόδοση βαθμολογιών σημαντικότητας στα χαρακτηριστικά εισόδου με τον υπολογισμό των τιμών Shapley, οι οποίες βασίζονται στη θεωρία συνεργατικών παιγνίων.
2) **Model agnosticity**: Το LIME είναι μια model-agnostic μέθοδος, πράγμα που σημαίνει ότι μπορεί να εφαρμοστεί σε κάθε τύπο μοντέλου μαύρου κουτιού. Το SHAP, από την άλλη πλευρά, είναι model-specific, πράγμα που σημαίνει ότι απαιτεί πρόσβαση στις εσωτερικές λειτουργίες του μοντέλου, όπως οι παράμετροι και η αρχιτεκτονική του μοντέλου.
3) **Πολυπλοκότητα**: Το SHAP είναι πιο απαιτητική μέθοδος από υπολογιστική άποψη από το LIME, καθώς απαιτεί την επίλυση μεγάλου αριθμού γραμμικών εξισώσεων για τον υπολογισμό των τιμών Shapley. Αντίθετα, το LIME είναι σχετικά ταχύτερο και απλούστερο στον υπολογισμό.
4) **Έξοδος**: Το LIME παρέχει εξηγήσεις με τη μορφή βαθμολογιών ή βαρών σημαντικότητας χαρακτηριστικών, οι οποίες υποδεικνύουν τη συμβολή κάθε χαρακτηριστικού στην πρόβλεψη. Το SHAP, από την άλλη πλευρά, παρέχει εξηγήσεις με τη μορφή τιμών Shapley, οι οποίες υποδεικνύουν την οριακή συμβολή κάθε χαρακτηριστικού στην πρόβλεψη.

4.2.1 Πλεονεκτήματα και μειονεκτήματα

LIME:

Πλεονεκτήματα: Το LIME είναι model-agnostic, απλό στη χρήση και μπορεί να παράγει εξηγήσεις τόσο για δεδομένα σε πίνακες όσο και για δεδομένα κειμένου. Μπορεί να χρησιμοποιηθεί για την επεξήγηση οποιουδήποτε τύπου μοντέλου, συμπεριλαμβανομένων των σύνθετων μοντέλων βαθιάς μάθησης. Το LIME μπορεί επίσης να παράγει επεξηγήσεις για μεμονωμένες περιπτώσεις, οι οποίες μπορεί να είναι χρήσιμες για το debugging και την ανάλυση σφαλμάτων.

Μειονεκτήματα: Το LIME παράγει προσεγγίσεις του μοντέλου, οι οποίες ενδέχεται να μην αποτυπώνουν την πραγματική συμπεριφορά του μοντέλου σε ορισμένες περιπτώσεις. Μπορεί επίσης να είναι ευαίσθητο στην επιλογή της τοπικής γειτονιάς (local neighborhood) και μπορεί να παράγει διαφορετικές

εξηγήσεις για την ίδια περίπτωση υπό διαφορετικά μεγέθη τοπικής επεξηγηματικότητας.

SHAP:

Πλεονεκτήματα: Το SHAP είναι μια θεωρητικά ορθή μέθοδος που μπορεί να παρέχει ακριβείς και συνεπείς εξηγήσεις για πολύπλοκα μοντέλα. Μπορεί να χειριστεί μη γραμμικά μοντέλα και μπορεί να παράγει εξηγήσεις για μεμονωμένες περιπτώσεις καθώς και για ολόκληρο το σύνολο δεδομένων. Το SHAP μπορεί επίσης να χειριστεί αλληλεπιδράσεις μεταξύ των χαρακτηριστικών, γεγονός που είναι σημαντικό για την αποτύπωση σύνθετων σχέσεων μεταξύ των χαρακτηριστικών.

Μειονεκτήματα: Το SHAP απαιτεί πρόσβαση στην εσωτερική λειτουργία του μοντέλου, η οποία μπορεί να μην είναι πάντα δυνατή ή πρακτική. Μπορεί επίσης να είναι υπολογιστικά απαιτητική, ιδίως για μεγάλα σύνολα δεδομένων ή μοντέλα με πολλά χαρακτηριστικά. Τέλος, το SHAP μπορεί να παράγει αρνητικές τιμές για ορισμένα χαρακτηριστικά, οι οποίες μπορεί να είναι δύσκολο να ερμηνευθούν.

4.2.2 Πειραματικές μελέτες

Αρκετές πειραματικές μελέτες έχουν συγκρίνει τις επιδόσεις των LIME και SHAP όσον αφορά την ικανότητά τους να παρέχουν ακριβείς και αξιόπιστες εξηγήσεις για μοντέλα μηχανικής μάθησης. Ακολουθούν ορισμένα βασικά ευρήματα από αυτές τις μελέτες:

1) **Ακρίβεια**: Το SHAP έχει βρεθεί ότι παρέχει πιο ακριβείς εξηγήσεις από τη LIME. Για παράδειγμα, μια μελέτη των Hooker κ.ά. (2018) σύγκρινε τις επιδόσεις του LIME και του SHAP σε διάφορα μοντέλα ταξινόμησης εικόνων και διαπίστωσε ότι το SHAP υπερείχε σταθερά έναντι του LIME τόσο ως προς την ακρίβεια όσο και ως προς τη σταθερότητα.
2) **Συνέπεια**: Το SHAP έχει επίσης βρεθεί ότι είναι πιο συνεπής μέθοδος από το LIME στη δημιουργία εξηγήσεων. Σε μια μελέτη των Lundberg και Lee (2017), οι συγγραφείς σύγκριναν τη συνέπεια των LIME και SHAP σε διαφορετικά υποσύνολα δεδομένων και διαπίστωσαν ότι το SHAP ήταν πιο συνεπές από το LIME στη δημιουργία εξηγήσεων.
3) **Σταθερότητα**: Έχει διαπιστωθεί ότι το LIME είναι πιο ανθεκτικό σε αλλαγές στα δεδομένα εισόδου σε σχέση με το SHAP. Για παράδειγμα, μια μελέτη των Datta κ.ά. (2018) σύγκρινε την ανθεκτικότητα του LIME

και του SHAP στην εξήγηση των προβλέψεων ενός μοντέλου ταξινόμησης κειμένου και διαπίστωσε ότι το LIME ήταν πιο ανθεκτικό σε αλλαγές στο κείμενο εισόδου, όπως η προσθήκη ή η αφαίρεση λέξεων.

4) **Ερμηνευσιμότητα**: Έχει διαπιστωθεί ότι τόσο το LIME όσο και το SHAP παρέχουν ερμηνεύσιμες εξηγήσεις, αλλά το LIME θεωρείται γενικά ότι είναι πιο διαισθητικό και πιο εύκολο να κατανοηθεί από μη ειδικούς. Αυτό οφείλεται στο γεγονός ότι το LIME παράγει εξηγήσεις με τη μορφή χαρακτηριστικών βαρών, τα οποία μπορούν εύκολα να ερμηνευθούν ως ένδειξη της σημασίας κάθε χαρακτηριστικού στην πρόβλεψη. Αντίθετα, το SHAP παράγει εξηγήσεις με τη μορφή τιμών Shapley, οι οποίες μπορεί να είναι πιο δύσκολο να ερμηνευθούν από μη ειδικούς.

Συνολικά, η επιλογή μεταξύ του LIME και του SHAP εξαρτάται από τις ειδικές ανάγκες της εφαρμογής και το συμβιβασμό μεταξύ ακρίβειας, συνέπειας, ανθεκτικότητας και ερμηνευσιμότητας. Σε γενικές γραμμές, η SHAP είναι μια πιο ισχυρή και ακριβής μέθοδος, αλλά μπορεί να είναι πιο δαπανηρή υπολογιστικά και λιγότερο διαισθητική από τη LIME.

5 *Υλοποίηση του Μοντέλου*

Χρησιμοποιήθηκε η τεχνική της Βαθιάς Μάθησης για δύο σύνολα κειμένων στα οποία έγινε η χρήση και του μοντέλου LIME αλλά και του SHAP για την επεξήγηση του αλγορίθμου αλλά και την άντληση πληροφοριών για το ποιο είναι το καταλληλότερο επεξηγηματικό μοντέλο με την μεγαλύτερη ακρίβεια στο πρόβλημά μας. Για την επίτευξη αυτού του στόχου, έγινε τροποποίηση του βασικού κώδικα χρησιμοποιώντας την προγραμματιστική γλώσσα Python.

5.1 Βιβλιοθήκες

Για την προσέγγισή μας στη Βαθιά Μάθηση, χρησιμοποιήθηκαν συγκεκριμένες βιβλιοθήκες που μας παρείχαν τις απαραίτητες λειτουργίες για τη δημιουργία των τεχνητών νευρωνικών δικτύων. Παρακάτω μια σύντομη περιγραφή των πιο σημαντικών βιβλιοθηκών που χρησιμοποιήθηκαν.

• **Torch**: Είναι μια βιβλιοθήκη ML ανοικτού κώδικα που χρησιμοποιείται για τη δημιουργία βαθιών νευρωνικών δικτύων και είναι γραμμένη στη γλώσσα σεναρίων Lua. Είναι μία από τις προτιμώμενες πλατφόρμες για την έρευνα της βαθιάς μάθησης. Η βιβλιοθήκη αυτή παρέχει μια εξειδικευμένη επιστημονική υποστήριξη και μια πληθώρα αλγορίθμων Βαθιάς Μάθησης και τεχνητών νευρωνικών δικτύων, τα οποία μπορούν να εφαρμοστούν σε διάφορες εφαρμογές. Η συγκεκριμένη βιβλιοθήκη παρέχει έτοιμες συναρτήσεις για τη δημιουργία και εκπαίδευση των τεχνητών νευρωνικών δικτύων, κάνοντας τη διαδικασία πολύ πιο εύκολη και αποτελεσματική για τους χρήστες της. Συνολικά, το Torch αποτελεί μια αξιόπιστη επιλογή για όσους αναζητούν ένα αποτελεσματικό και ευέλικτο εργαλείο για την ανάπτυξη εφαρμογών Μηχανικής Μάθησης.

• **Pandas**: Η βιβλιοθήκη Pandas είναι μια δημοφιλής βιβλιοθήκη γραμμένη σε Python που χρησιμοποιείται για την ανάλυση και επεξεργασία δεδομένων. Δημιουργήθηκε από τον Wes McKinney το 2008, ως απάντηση στην ανάγκη για ένα ισχυρό και ευέλικτο εργαλείο ποσοτικής ανάλυσης. Η βιβλιοθήκη Pandas έχει γίνει ένα από τα δημοφιλέστερα εργαλεία Python για την επεξεργασία δεδομένων, και μπορεί να χρησιμοποιηθεί για τη διαχείριση αριθμητικών πινάκων, χρονοσειρών και άλλων δεδομένων με ευκολία. Η βιβλιοθήκη Pandas παρέχει επίσης πολλές λειτουργίες για την εξαγωγή, τον

συνδυασμό και την ανάλυση δεδομένων, καθιστώντας την μια πολύτιμη εργαλειοθήκη για τους επαγγελματίες της ανάλυσης δεδομένων και της επιστήμης των δεδομένων.

• **Tensorflow**: Η βιβλιοθήκη TensorFlow είναι μια πλατφόρμα ανοιχτού κώδικα για τη δημιουργία και εκπαίδευση βαθιών νευρωνικών δικτύων. Μπορεί να χρησιμοποιηθεί σε διάφορες εργασίες μηχανικής μάθησης, όπως η επεξεργασία εικόνων, η αναγνώριση φωνής και η ανάλυση φυσικής γλώσσας. Ωστόσο, εστιάζει ιδιαίτερα στην εκπαίδευση και την εξαγωγή συμπερασμάτων από βαθιά νευρωνικά δίκτυα, καθώς παρέχει ευέλικτα εργαλεία για την κατασκευή και την εκπαίδευση πολύπλοκων μοντέλων.

• **Pytorch**: Είναι μια βελτιστοποιημένη βιβλιοθήκη για tensors που βασίζεται στην Python και την Torch και χρησιμοποιείται κυρίως για εφαρμογές που χρησιμοποιούν GPU και CPU. Το PyTorch πολλές φορές προτιμάται έναντι άλλων πλαισίων Deep Learning όπως το TensorFlow και το Keras, καθώς χρησιμοποιεί δυναμικούς υπολογιστικούς γράφους και είναι πλήρως Pythonic. Επιτρέπει σε επιστήμονες, προγραμματιστές και debuggers νευρωνικών δικτύων να εκτελούν και να δοκιμάζουν τμήματα του κώδικα σε πραγματικό χρόνο. Έτσι, οι χρήστες δεν χρειάζεται να περιμένουν να υλοποιηθεί ολόκληρος ο κώδικας για να ελέγξουν αν ένα μέρος του κώδικα λειτουργεί ή όχι. Η Pytorch και η Tensoflow είναι αρκετά παρόμοιες βιβλιοθήκες και παίζουν κύριο ρόλο για την υλοποίηση του αλγορίθμου BERT.

• **Transformers**: Η βιβλιοθήκη Transformers είναι μια βιβλιοθήκη ανοικτού κώδικα για την επεξεργασία φυσικής γλώσσας (NLP) που παρέχει προ-εκπαιδευμένα μοντέλα για διάφορες εργασίες NLP, όπως η μοντελοποίηση γλώσσας και η ταξινόμηση κειμένου. Είναι χτισμένη πάνω στο PyTorch και το TensorFlow και προσφέρει ένα εύχρηστο API για fine-tuning pre-trained μοντέλα ή training custom μοντέλα από το μηδέν. Παρέχει επίσης βοηθητικά προγράμματα και εργαλεία για την εργασία με δεδομένα NLP και επιτρέπει την εύκολη ενσωμάτωση με άλλες βιβλιοθήκες NLP. Η υποστήριξη της βιβλιοθήκης για σύγχρονα προ-εκπαιδευμένα μοντέλα, όπως τον BERT, την έχει κάνει δημοφιλή μεταξύ των ερευνητών και των επαγγελματιών της κοινότητας NLP.

5.2 Περιγραφή των Datasets

Για καλύτερα συμπεράσματα και αποτελέσματα του μοντέλου, θα κάνουμε χρήση 2 διαφορετικών Datasets, έτσι ώστε να έχουμε και την δυνατότητα σύγκρισης και αποφυγής λανθασμένων αποτελεσμάτων.

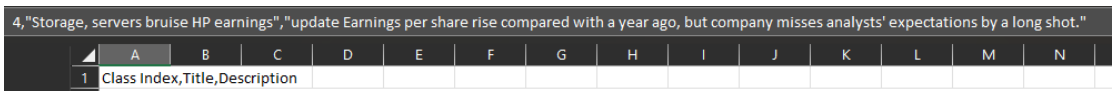
5.2.1 AG News Classification Dataset

Η AG περιλαμβάνει πάνω από ένα εκατομμύριο ειδησεογραφικά άρθρα, από τα οποία, περισσότερα από 2000 προέρχονται από την ComeToMyHead, μια ακαδημαϊκή μηχανή αναζήτησης ειδήσεων που λειτουργεί από τον Ιούλιο του 2004. Αυτό το σύνολο δεδομένων διατίθεται από την ακαδημαϊκή κοινότητα για ερευνητικούς σκοπούς σε διάφορους τομείς, όπως εξόρυξη δεδομένων (όπως ομαδοποίηση και ταξινόμηση), ανάκτηση πληροφοριών (όπως κατάταξη και αναζήτηση), XML, συμπίεση δεδομένων, ροή δεδομένων και άλλες μη εμπορικές δραστηριότητες.

Το σύνολο δεδομένων ταξινόμησης θεμάτων ειδήσεων της AG δημιουργήθηκε με την επιλογή των τεσσάρων μεγαλύτερων κλάσεων από το αρχικό σώμα δεδομένων. Καθεμία από αυτές τις τέσσερις κλάσεις περιέχει 30.000 δείγματα για εκπαίδευση και 1.900 δείγματα για δοκιμή, με αποτέλεσμα συνολικά 120.000 δείγματα εκπαίδευσης και 7.600 δείγματα δοκιμής. Το αρχείο "classes.txt" παρέχει έναν κατάλογο των κλάσεων που αντιστοιχούν σε κάθε label.

Τα δεδομένα εκπαίδευσης και δοκιμής αποθηκεύονται στα αρχεία "train.csv" και "test.csv", αντίστοιχα. Κάθε αρχείο έχει τρεις στήλες: δείκτης κλάσης (Class Id) (που κυμαίνεται από 1 έως 4), τίτλος (Title) και περιγραφή (Description). Ο τίτλος και η περιγραφή περικλείονται σε διπλά εισαγωγικά (") και κάθε εσωτερικό διπλό εισαγωγικό αντιπροσωπεύεται από δύο διπλά εισαγωγικά (""). Οι νέες γραμμές αναπαρίστανται με μια ανάποδη κάθετο ακολουθούμενη από το χαρακτήρα "n" ("\n").

Τα class ids είναι αριθμημένα από 1-4, όπου το 1 αφορά γενικά θέματα για όλον τον κόσμο, το 2 για τον αθλητισμό, το 3 για τις επιχειρήσεις και το 4 για την επιστήμη/τεχνολογία.



Εικόνα 50 :AG Dataset

5.2.2 IMDB Dataset

Το σύνολο δεδομένων IMDB είναι μια ευρηματική συλλογή 50.000 κριτικών ταινιών που μπορεί να χρησιμοποιηθεί για επεξεργασία φυσικής γλώσσας ή ανάλυση κειμένου. Είναι ειδικά σχεδιασμένο για τη δυαδική sentiment classification και είναι σημαντικά μεγαλύτερο από προηγούμενα σύνολα δεδομένων αναφοράς. Το σύνολο δεδομένων αποτελείται από 25.000 κριτικές για εκπαίδευση και 25.000 για δοκιμή. Ο στόχος είναι να χρησιμοποιηθούν αλγόριθμοι ταξινόμησης ή βαθιάς μάθησης για την πρόβλεψη του αριθμού των θετικών και αρνητικών κριτικών.

5.3 *Προετοιμασία της εισόδου*

Για την κατάλληλη προετοιμασία του input που θα δεχτεί ο BERT, αρχικά φορτώθηκε το σύνολο δεδομένων από το CSV αρχείο χρησιμοποιώντας το pandas. Στη συνέχεια, αρχικοποιήθηκε ο tokenizer BERT χρησιμοποιώντας το προ-εκπαιδευμένο μοντέλο "bert-base-uncased".

```
# Initialize the BERT tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

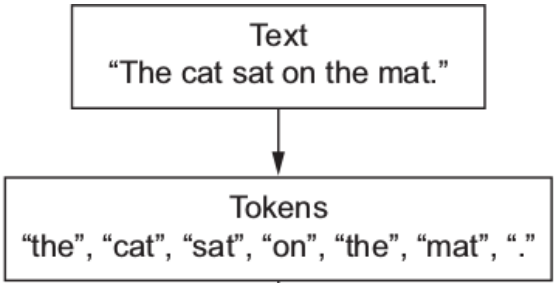
Εικόνα 51: Αρχικοποιήση του BERT-tokenizer

Ο tokenizer BERT είναι ένα εργαλείο που μετατρέπει το ακατέργαστο κείμενο σε μορφή που μπορεί να επεξεργαστεί από μοντέλα που βασίζονται σε BERT. Ακολουθεί μια συγκεκριμένη διαδικασία tokenization που χωρίζει τις λέξεις σε υπολέξεις (subwords), οι οποίες στη συνέχεια αντιστοιχίζονται σε μοναδικά ακέραια IDs (0,1).

Ο tokenizer εκτελεί μερικές σημαντικές λειτουργίες:

1) **Tokenization:** Ο tokenizer παίρνει το ακατέργαστο κείμενο εισόδου και το χωρίζει σε έναν κατάλογο μεμονωμένων token. Ακολουθεί μια προσέγγιση tokenization λέξη-τεμάχιο, όπου οι λέξεις αναλύονται σε υπολέξεις με βάση ένα προκαθορισμένο λεξιλόγιο.
2) **Χειρισμός λέξεων εκτός λεξιλογίου (OOV):** Ο tokenizer χειρίζεται τις λέξεις OOV αναλύοντάς τες σε μικρότερες υπολέξεις που υπάρχουν στο λεξιλόγιο. Για παράδειγμα, η λέξη "unbelievable" μπορεί να διασπαστεί σε "un", "believe" και "able".
3) **Ειδικά Tokens:** Ο tokenizer προσθέτει ειδικά tokens στην αρχή και στο τέλος κάθε ακολουθίας εισόδου. Αυτά τα σημεία είναι τα [CLS] (ταξινόμηση) και [SEP] (διαχωριστικό). Το token [CLS] χρησιμοποιείται για να προσδιορίσει την αρχή της ακολουθίας εισόδου, ενώ το token [SEP] χρησιμοποιείται για να διαχωρίσει πολλαπλές ακολουθίες εισόδου.
4) **Padding:** Ο tokenizer γεμίζει τις μικρότερες ακολουθίες με ένα ειδικό διακριτικό [PAD] για να τις κάνει να έχουν το ίδιο μήκος με τις μεγαλύτερες ακολουθίες. Αυτό είναι απαραίτητο για τη μαζική επεξεργασία σε μοντέλα βαθιάς μάθησης.

Συνολικά, ο tokenizer του BERT είναι ένα σημαντικό στοιχείο των μοντέλων που βασίζονται στο BERT, το οποίο βοηθά στη μετατροπή του ακατέργαστου κειμένου σε μορφή που μπορεί να γίνει κατανοητή από το μοντέλο.



Εικόνα 52 :Tokenization

Με την παρακάτω εντολή γίνεται η προσθήκη των ειδικών token [CLS] και [SEP] για να επισημάνουμε την αρχή και το τέλος της ακολουθίας.

```
# Tokenize the text column and add special tokens
df['tokenized_text'] = df['text'].apply(lambda x: tokenizer.encode(x, add_special_tokens=True))
```

Εικόνα 53: Adding special tokens

Τέλος, όπως προαναφέραμε στην διαδικασία του tokenization, κάνουμε pad τις αλληλουχίες που έχουν μετατραπεί σε token σε σταθερό μήκος 128 χρησιμοποιώντας το token [PAD], το οποίο είναι το token ID 0 στο λεξιλόγιο BERT. Το DataFrame που προκύπτει έχει δύο πρόσθετες στήλες:

- το "tokenized_text" που περιέχει τα token IDs των tokenized ακολουθιών και
- το "padded_text" που περιέχει τα padded token IDs.

```
max_length = 128  # define your desired max sequence length
df['padded_text'] = df['tokenized_text'].apply(lambda x: x[:max_length] + [0]*(max_length-len(x)))
```

✓ 2.3s

```
[101, 2023, 2832, 1010, 2174, 1010, 22891, 2033, 2053, 2965, 1997, 2004, 17119, 18249, 2075, 1996, 9646, 1997,
```

Εικόνα 54: padded text

Στην συνέχεια κάνουμε import το BertForSequenceClassification. Το μοντέλο BertForSequenceClassification της βιβλιοθήκης Transformers είναι ένα προ-εκπαιδευμένο μοντέλο BERT που έχει επιπλέον head ταξινόμηση για εργασίες ταξινόμησης ακολουθιών. Η head ταξινόμηση αποτελείται από ένα γραμμικό επίπεδο που λαμβάνει την έξοδο του τελευταίου κρυμμένου επιπέδου του μοντέλου BERT και την αντιστοιχίζει σε ένα διάνυσμα μεγέθους num_labels, όπου num_labels είναι ο αριθμός των κλάσεων στην εργασία ταξινόμησης.

5.4 Pre-process

Πρώτα γίνεται αρχικοποίηση του μοντέλου LabelEncoder από την βιβλιοθήκη sklearn και συγκεκριμένα από το sklearn.preprocessing έτσι ώστε να οριστεί ένας label encoder για τη μετατροπή των string labels σε numeric labels. Για το πρώτο Dataset κάνουμε αντιστοίχιση των κατηγοριών world, sport, business, technology με 0, 1, 2 και 3 αντίστοιχα.

Ύστερα δημιουργούμε την κλάση TextClassificationDataset η οποία έχει διαμορφωθεί με τέτοιο τρόπο, έτσι ώστε όταν επεξεργαστούμε το dataset, με βάση τα ορίσματα που του δίνουμε, θα μπορούμε να το μετατρέψουμε σε μία μορφή που χρησιμοποιείτε εύκολα στον BERT, και θα έχουμε ως έξοδο το επιθυμητό αποτέλεσμα.

Τέλος κάνουμε την πιο βασική διαδικασία του pre-process, το λεγόμενο “split”, δηλαδή ο διαχωρισμός του αρχείου σε training set και validation ή testing set. Στο παράδειγμά μας χωρίσαμε το dataset σε 80% για το training και 20% για το testing. Σε γενικές γραμμές, η τοποθέτηση του 80% των δεδομένων στο σύνολο εκπαίδευσης, του 20% στο σύνολο δοκιμής είναι μια καλή επιλογή. Η βέλτιστη κατανομή του συνόλου δοκιμής και εκπαίδευσης εξαρτάται από παράγοντες όπως το use case, η δομή του μοντέλου, η διάσταση των δεδομένων κ.λπ.

```
# Split the dataset into training and validation sets
train_df = df.sample(frac=0.8, random_state=42)
val_df = df.drop(train_df.index)

# Define the dataset and data loader for training
train_dataset = TextClassificationDataset(train_df['Description'].values, train_df['Class Index'].values, tokenizer, max_length)
train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)

# Define the dataset and data loader for validation
val_dataset = TextClassificationDataset(val_df['Description'].values, val_df['Class Index'].values, tokenizer, max_length)
val_loader = DataLoader(val_dataset, batch_size=16, shuffle=True)
```

Εικόνα 55: Spliter and DataLoader

Ακόμη για την υλοποίηση του διαχωρισμού έγινε η χρήση του Loader για να μπορούμε να διαχειριστούμε το πόσα samples (γραμμές από το αρχείο) θα επεξεργάζονται ταυτόχρονα στο νευρωνικό δίκτυο. Είναι αρκετά σημαντική η χρήση των samples, διότι μπορούμε αναλόγως την επεξεργαστική ισχύ που διαθέτουμε (π.χ. GPU, μεγάλος παραλληλισμός εργασιών) να αυξήσουμε ή αντιστοίχως να μειώσουμε το batch_size (ο αριθμός των tasks που θα επεξεργάζεται ταυτόχρονα) με βάση τις δυνατότητες του επεξεργαστή μας. Με αυτόν τον τρόπο μπορούμε να επιτύχουμε ταυτόχρονο training.

Από το output του πρώτου input του training set μπορούμε να διαπιστώσουμε ότι υπάρχουν τα special tokens στην αρχή [101] αλλά και στο τέλος [102] του input_ids, που είναι τα [CLS] και [SEP] αντίστοιχα.

```
{'input_ids': tensor([  101,  1000,  7955,  2288, 20118,  4904,  1051,  1005,  1996,  8909,
         4402,  1051,  1005,  1040, 25811,  1005,  4654,  3401,  2361,  1005,
         1999, 14347,  5233,  2007,  1996,  2060, 16422,  1010,  2030,  2004,
        18502,  2000,  1996,  2712,  5932,  4830, 12384,  2917,  1010,  2030,
         2013,  7488,  6805,  2030, 11629,  2030,  4629, 25624,  8091,  1005,
         9932, 13728, 11187,  2030, 27941,  1005, 21358,  5686,  2027, 12731,
         2094,  2202,  2000,  1996,  2300,  2021,  3432,  2246,  2005, 12173,
         2000,  1037,  2785,  1051,  1005,  2689,  2008, 11333,  1005,  1050,
         1005,  1056,  1037,  2978,  9202, 16185,  2099,  1037,  2096,  1012,
          102,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0]), 'attention_mask': tensor([1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0]), 'labels': tensor(1)}
1
```

Εικόνα 56: Training set output of the 1st input

5.5 Fine-tuning του μοντέλου BERT

5.5.1 Γενική περιγραφή

Το αλγόριθμος BERT της Google αποτέλεσε μια αλλαγή στη μοντελοποίηση φυσικής γλώσσας, ιδίως λόγω της προ-εκπαίδευσης / fine-tuning. Μετά την προ-εκπαίδευση με unsupervised τρόπο σε έναν τεράστιο όγκο δεδομένων κειμένου, το μοντέλο μπορεί να τελειοποιηθεί γρήγορα σε μια συγκεκριμένη εργασία με σχετικά λίγα labels, επειδή τα γενικά γλωσσικά πρότυπα έχουν ήδη διδαχθεί κατά την προ-εκπαίδευση.

Η λεπτομερής ρύθμιση του BERT είναι "straightforward", απλά προσθέτοντας ένα επιπλέον στρώμα μετά το τελικό στρώμα του BERT και εκπαιδεύοντας ολόκληρο το δίκτυο για λίγες μόνο epochs. Συνήθως μπορούμε να επιτύχουμε ισχυρές επιδόσεις στα τυποποιημένα προβλήματα αναφοράς NLP GLUE, SQuAD και SWAG, τα οποία εξετάζουν διαφορετικές πτυχές της εξαγωγής συμπερασμάτων φυσικής γλώσσας, μετά από fine-tuning για μόλις 2-4 εποχές (epochs) με τον ADAM optimizer, με Learning rate μεταξύ 1e-5 έως 5e-5, μια μέθοδος που έχει υιοθετηθεί συνήθως στην ερευνητική κοινότητα.

Λόγω της αξιοσημείωτης επιτυχίας του, αυτό το παράδειγμα pre-training / fine-tuning έχει γίνει συνήθης πρακτική στον τομέα. Ωστόσο, από επιστημονικής άποψης, στην πραγματικότητα δεν κατανοούμε πολύ καλά τη διαδικασία του fine-tuning. Πολλές φορές είναι δύσκολο να κατανοήσουμε ποια στρώματα αλλάζουν κατά τη διάρκεια του fine-tuning, ή ακόμη αν χρειάζεται ο κώδικας fine-tuning και πόσο σταθερά αποτελέσματα μπορούμε να παράγουμε.

Στη βιβλιογραφία, ο προτεινόμενος αριθμός εποχών για το BERT έχει οριστεί μεταξύ 2 και 4, όπως προαναφέραμε, με το 3 να αποτελεί κοινή επιλογή. Ωστόσο, δεν επιτυγχάνεται μέγιστη απόδοση και η ακρίβεια εξακολουθεί να βελτιώνεται μετά και από 10 εποχές, αν και σε μικρότερο βαθμό. Ο αντίκτυπος των εποχών εκπαίδευσης σε ένα σύνολο δεδομένων με πολλές κατηγορίες ειδήσεων είναι μεγαλύτερος, καθώς δεν έχει τόσο καλή απόδοση, αλλά μπορούμε να υποθέσουμε ότι ο αντίκτυπος του αριθμού των εποχών μειώνεται σε ένα μικρότερο σύνολο κατηγοριών ειδήσεων. Σε αυτήν την εργασία επιλέχθηκε η εκπαίδευση των μοντέλων να γίνει σε 10 εποχές για τα περισσότερα πειράματα, καθώς προσφέρει μια καλή αντιστάθμιση μεταξύ του χρόνου εκπαίδευσης και της ακρίβειας.

5.5.2 Υλοποίηση fine-tuning

Για την υλοποίηση του αλγορίθμου αρχικά κατεβάζουμε το μοντέλο, κάνοντας χρήση της βιβλιοθήκης BertForSequenceClassification, η οποία

παρέχει τον αλγόριθμο του BERT σε μορφή έτοιμη για Classification. Το μοντέλο που θα χρησιμοποιήσουμε είναι το bert-base-uncased. Προτιμάμε την χρήση του uncased μοντέλου από το cased διότι όπως προαναφέραμε γίνεται διαμόρφωση του κειμένου πριν την επεξεργασία (μετατροπή σε πεζά, αφαίρεση σημείων στίξης, κ.λπ.), ενώ στο cased μοντέλο το κείμενο είναι το ίδιο με το κείμενο εισόδου (χωρίς αλλαγές).

5.5.2.1 Optimizer

Για την εκπαίδευση των νευρωνικών δικτύων απαιτούνται αλγόριθμοι που να επιτρέπουν την προσαρμογή των παραμέτρων του δικτύου, όπως τα βάρη, η παράμετρος eps, καθώς και ο ρυθμός μάθησης (learning rate), προκειμένου να ελαχιστοποιηθεί το σφάλμα. Η διαδικασία προσαρμογής αυτών των παραμέτρων καθορίζεται από διάφορους τύπους αλγορίθμων.

- Το **learning rate** είναι η πιο σημαντική υπερπαράμετρος του νευρωνικού δικτύου. Είναι η παράμετρος που καθορίζει την προσαρμογή των βαρών του δικτύου μας σε σχέση με την κλίση απώλειας. Ουσιαστικά ορίζει πόσο γρήγορα ή αργά θα κινηθούμε προς τα βέλτιστα βάρη. Μπορεί να αποφασίσει πολλά πράγματα κατά την εκπαίδευση του δικτύου. Στους περισσότερους optimizers του Keras, η προεπιλεγμένη τιμή του ρυθμού μάθησης είναι 0,001. Είναι η προτεινόμενη τιμή για την έναρξη της εκπαίδευσης.

- Η παράμετρος **weight decay** είναι μια υπερπαράμετρος που ελέγχει την ισχύ της κανονικοποίησης της αποσύνθεσης του βάρους στη μηχανική μάθηση. Καθορίζει το συμβιβασμό μεταξύ overfitting και underfitting εξισορροπώντας την ικανότητα του μοντέλου να προσαρμόζεται στα δεδομένα εκπαίδευσης με την ικανότητά του να γενικεύει σε νέα δεδομένα. Η βέλτιστη τιμή καθορίζεται συνήθως με διασταυρωμένη επικύρωση ή άλλες τεχνικές συντονισμού υπερπαραμέτρων.

```
# Load the pre-trained BERT model and add a classification layer on top
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=4)

# Define the optimizer and learning rate scheduler
optimizer = torch.optim.Adam(model.parameters(), lr=2e-4, weight_decay=0.01)
```

Εικόνα 57: Optimizer

5.5.2.2 Αντιμετώπιση Overfit

Η εκπαίδευση ενός νευρωνικού δικτύου έχει σαν στόχο να μπορεί να γενικεύσει στα νέα δεδομένα που δεν έχει δει, βασιζόμενο στην εκπαίδευση που έχει λάβει από το αρχικό σύνολο δεδομένων. Αν ένα μοντέλο έχει μικρή χωρητικότητα,

τότε δεν μπορεί να μάθει τα δεδομένα, ενώ αν έχει πολύ μεγάλη χωρητικότητα μπορεί να μάθει πολύ καλά τα δεδομένα του συνόλου εκπαίδευσης, σε βαθμό να αποτυγχάνει να γενικεύσει σε νέα δεδομένα. Η κατάσταση όπου ένα μοντέλο μαθαίνει πολύ καλά τα δεδομένα εκπαίδευσης, αλλά αποτυγχάνει να αντιμετωπίσει νέα δεδομένα που δεν έχει δει, ονομάζεται overfitting. Το overfitting είναι ένα πολύ συνηθισμένο πρόβλημα στη Μηχανική Μάθηση και υπάρχει ένα εκτεταμένο φάσμα βιβλιογραφίας αφιερωμένο στη μελέτη μεθόδων για την αντιμετώπιση του.

Κάποιες μέθοδοι αντιμετώπισης είναι οι εξής :

- Hold-out
- Cross-validation
- L1 / L2 regularization
- Remove layers / number of units per layer
- Dropout

5.5.2.3 **AdamW**

Ο αλγόριθμος που χρησιμοποιήθηκε στην προσέγγιση μας για την εκπαίδευση του CNN νευρωνικού δικτύου, είναι ο αλγόριθμος AdamW. Ο αλγόριθμος AdamW είναι μια στοχαστική μέθοδος gradient descent που βασίζεται στην προσαρμοστική εκτίμηση των ροπών πρώτης και δεύτερης τάξης με μια πρόσθετη μέθοδο αποσύνθεσης των βαρών. Ο στόχος αυτού του αλγορίθμου είναι να ενεργήσει με προσοχή κατά μήκος του χώρου των παραμέτρων, ώστε να βρει την ελάχιστη τιμή της συνάρτησης κόστους. Δεν θέλουμε να προχωρήσουμε πολύ γρήγορα, διότι ενδέχεται να υπερπηδήσουμε το ελάχιστο σημείο. Αντίθετα, θέλουμε να μειώσουμε το ρυθμό του αλγορίθμου, ώστε να εξετάζει προσεκτικά την περιοχή και να βρίσκουμε με ακρίβεια το ελάχιστο.

5.5.3 Training

Στη λειτουργία εκπαίδευσης, ο BERT εκπαιδεύεται ώστε να κατανοεί τη σημασία των λέξεων στο κείμενο εισόδου και να ταξινομεί τα κείμενα σε διαφορετικές κατηγορίες ειδήσεων. Αυτό οφείλεται στο γεγονός ότι ο BERT

είναι ένα προ-εκπαιδευμένο μοντέλο που έχει μάθει να αναπαριστά τη σημασία των λέξεων μέσω της εκπαίδευσής του σε ένα μεγάλο σώμα κειμένων και έχει ρυθμιστεί λεπτομερώς για συγκεκριμένες εργασίες με την προσθήκη μιας head ταξινόμησης πάνω στο προ-εκπαιδευμένο μοντέλο και την εκπαίδευση ολόκληρου του μοντέλου από άκρη σε άκρη.

Κατά τη διάρκεια της διαδικασίας εκπαίδευσης, τα κείμενα εισόδου συμβολίζονται από τον tokenizer, ο οποίος τα αναλύει σε subword tokens. Αυτά τα subword tokens μετατρέπονται στη συνέχεια σε χαρακτηριστικά εισόδου, τα οποία τροφοδοτούνται στο μοντέλο BERT. Το μοντέλο BERT αποτελείται από μια ακολουθία στρωμάτων μετασχηματισμού, τα οποία επεξεργάζονται τα χαρακτηριστικά εισόδου για να μάθουν τις αναπαραστάσεις των υπολέξεων των κειμένων εισόδου.

Η έξοδος του μοντέλου BERT τροφοδοτείται στη συνέχεια στην ταξινόμηση head, η οποία αποτελείται από ένα ή περισσότερα πλήρως συνδεδεμένα στρώματα που αντιστοιχούν την έξοδο του μοντέλου BERT στην προβλεπόμενη κατηγορία ειδήσεων.

Κατά τη διάρκεια της εκπαίδευσης, τα βάρη τόσο του μοντέλου BERT όσο και της ταξινόμησης ενημερώνονται με τη χρήση backpropagation και gradient descent, έτσι ώστε το μοντέλο να μαθαίνει να αναπαριστά τη σημασία των λέξεων και να ταξινομεί τα κείμενα εισόδου.

Πιο συγκεκριμένα στην διαδικασία του training γίνονται οι εξής ενέργειες :

- Καθορισμός των συνόλων δεδομένων εκπαίδευσης και validation: Ο κώδικας υποθέτει ότι τα σύνολα δεδομένων train και val έχουν ήδη οριστεί πριν από την εκτέλεση αυτού του κώδικα.
- Καθορισμός μιας συνάρτησης για τον υπολογισμό των μετρικών κατά τη διάρκεια της εκπαίδευσης: Η συνάρτηση compute_metrics λαμβάνει τις προβλεπόμενες ετικέτες και υπολογίζει την ακρίβεια.
- Ορισμός Label encoder: Ο κώδικας ορίζει έναν κωδικοποιητή ετικετών για τη μετατροπή των ετικετών συμβολοσειράς σε αριθμητικές ετικέτες.
- Φόρτωση του προ-εκπαιδευμένου μοντέλου BERT: Ο κώδικας φορτώνει ένα προ-εκπαιδευμένο μοντέλο BERT με ένα επίπεδο ταξινόμησης ακολουθιών στην κορυφή. Η παράμετρος num_labels καθορίζει τον αριθμό των κλάσεων για την εργασία ταξινόμησης.
- Ορισμός του optimizer και του learning rate scheduler: Ο κώδικας ορίζει έναν Adam optimizer με learning rate 2e-4 και weight decay 0,01.
- Καθορισμός των ορίων εκπαίδευσης: Ο κώδικας ορίζει τα ορίσματα εκπαίδευσης, όπως ο κατάλογος εξόδου (output directory), ο αριθμός των εποχών, τα μεγέθη των παρτίδων (batch sizes), τα βήματα

προθέρμανσης (warmup steps), οι ρυθμίσεις καταγραφής (logging settings), η στρατηγική αξιολόγησης (evaluation strategy) και το αν θα φορτώνεται το καλύτερο μοντέλο στο τέλος (load_best_model_at_end = True).

- Δημιουργία του εκπαιδευτή και εκπαίδευση του μοντέλου: Ο κώδικας δημιουργεί ένα αντικείμενο Trainer με το καθορισμένο μοντέλο, τα ορίσματα εκπαίδευσης και τα σύνολα δεδομένων. Προσθέτει επίσης ένα callback πρόωρης διακοπής με patience 3 και threshold 0,01. Η συνάρτηση compute_metrics χρησιμοποιείται για την παρακολούθηση της ακρίβειας κατά τη διάρκεια της εκπαίδευσης

5.6 *Επεξηγηματικότητα (LIME, SHAP)*

Για την υλοποίηση των αλγορίθμων κάνουμε χρήση της βιβλιοθήκης lime και shap αντίστοιχα, η οποία θα μας βοηθήσει να επιλύσουμε την ερμηνευσιμότητα του μοντέλου, παράγοντας τοπικά αξιόπιστες επεξηγήσεις. Απαραίτητη προϋπόθεση για την λειτουργία του μοντέλου είναι η χρήση μίας prediction συνάρτησης, η οποία θα είναι και η βασική συνάρτηση που θα κάνει την επεξήγηση του μοντέλου. Η συγκεκριμένη συνάρτηση παίρνει ως input το text και ως output θα επιστρέφει την κατηγορία ειδήσεων με το μεγαλύτερο ποσοστό πιθανότητας.

Ακόμη κάνουμε αρχικοποίηση του αντικειμένου explainer και του δίνουμε σαν όρισμα τις κλάσεις, δηλαδή τις κατηγορίες ειδήσεων (για το πρώτο Dataset).

```
import lime
import lime.lime_text
from lime import lime_text

label_map = ['World', 'Athletics', 'Bussiness', 'Technology']
explainer = lime_text.LimeTextExplainer(class_names=label_map)
# Initialize the LIME explainer
```

Εικόνα 58 : initialize LIME explainer

Το αντικείμενο και η συνάρτηση που αρχικοποιήσαμε είναι ζωτικής σημασίας για την δημιουργία του επεξηγητή.

```
def predict(text):
    # Define the device to run the model on
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    # Tokenize the text and convert to input tensors
    inputs = tokenizer(text, padding=True, truncation=True, max_length=max_length, return_tensors='pt')
    input_ids = inputs['input_ids']
    attention_mask = inputs['attention_mask']

    # Run the model and return the softmax output
    with torch.no_grad():
        outputs = model(input_ids=input_ids, attention_mask=attention_mask)
        logits = outputs.logits
        softmax_output = torch.softmax(logits, dim=1).detach().cpu().numpy()

    return softmax_output
```

Εικόνα 59: Predict Function for LIME

5.6.1 LIME Explainer

Το τελικό στάδιο του αλγορίθμου LIME, είναι η παραγωγή του explainer (επεξηγητή), για ένα συγκεκριμένο κείμενο που θα του ορίσουμε.

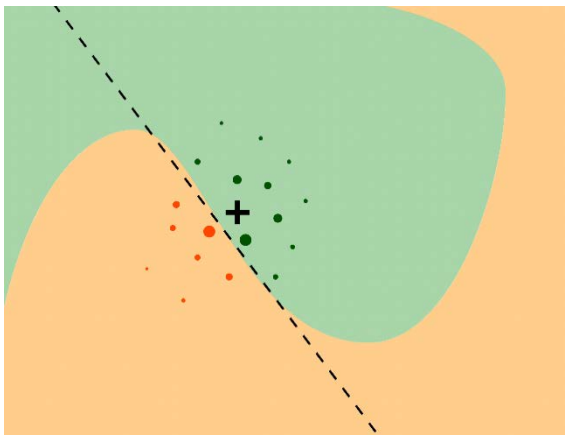
```
# Select a random text for explanation
#text = test_df.sample()['text'].values[0]
text = "Developers get early code for new operating system'skin' still being crafted"
# Generate an explanation for the selected text
exp = explainer.explain_instance(text, predict, num_features=20, num_samples=300, top_labels=4)

# Print the explanation
exp.show_in_notebook(text=text)
```

Εικόνα 60 : Explainer

Όπως παρατηρούμε και στην Εικόνα 60, ο επεξηγητής παίρνει ως ορίσματα το συγκεκριμένο **text** που έχουμε ορίσει, την συνάρτηση **predict** που έχουμε διαμορφώσει για το συγκεκριμένο πρόβλημα, το **num_features** που είναι ο μέγιστος αριθμός χαρακτηριστικών που υπάρχουν στην εξήγηση, το **num_samples** που ορίζεται ως το μέγεθος της περιοχής για την εκμάθηση του γραμμικού μοντέλου και τέλος τα **top_labels**, που είναι το ποσό των κλάσεων που θέλουμε να εμφανιστούν στα αποτελέσματα μας.

5.6.1.1 Τοπική ερμηνευσιμότητα

Με βάση το αποτέλεσμα που έβγαλε η σύνθετη συνάρτηση predict στο συγκεκριμένο sample και από τα γειτονικά 300 samples (με βάση το παράδειγμά μας num_samples=300) παρατηρούμε ποια από τις απλές συναρτήσεις (π.χ. γραμμική παλινδρόμηση, δέντρο απόφασης) είναι η πιο ιδανική για να προσομοιώσει καλύτερα την σύνθετη συνάρτηση μας.



Εικόνα 61: Local interpretability

Πηγή : https://ema.drwhy.ai/LIME.html

5.6.2 SHAP Explainer

Για την δημιουργία του επεξηγητή SHAP, χρησιμοποιήθηκε η βιβλιοθήκη transformers. Από την βιβλιοθήκη χρησιμοποιήθηκε ο SequenceClassificationExplainer, ένα εργαλείο επεξηγηματικότητας που χρησιμοποιείται στην επεξεργασία φυσικής γλώσσας και αναλύει τις εσωτερικές λειτουργίες των μοντέλων ταξινόμησης ακολουθιών για να εντοπίσει σημαντικά χαρακτηριστικά και σημεία στην ακολουθία εισόδου που συνέβαλαν περισσότερο στην απόφαση του μοντέλου, καθώς και τυχόν σφάλματα ή ασυνέπειες στις προβλέψεις του μοντέλου.

Ουσιαστικά δημιουργούμε μία σύνθετη συνάρτηση πρόβλεψης αντίστοιχη με την predict του αλγορίθμου LIME η οποία έχει ακριβώς την ίδια λειτουργεία.

Ύστερα δημιουργούμε την συνάρτηση explain_model_prediction_shap η οποία ουσιαστικά είναι υπεύθυνη για την δημιουργία του επεξηγητή.

```
def model_predict(x):
    inputs = tokenizer(x, padding=True, truncation=True, return_tensors="pt")
    outputs = model(**inputs)
    return torch.softmax(outputs.logits, dim=1).detach().numpy()


def explain_model_prediction_shap(x):
    explainer = SequenceClassificationExplainer(model, tokenizer)
    shap_values = explainer(x)
    class_index = explainer.predicted_class_index
    # Print the explanations (SHAP values)
    for token, shap_value in shap_values:
        print(f"{token}: {shap_value:.4f}")

    return shap_values,class_index
```

Εικόνα 62: Predict Function and explainer for Shap

Η τεχνική των Shapley values μπορεί να χρησιμοποιηθεί για να εξηγήσει οποιοδήποτε μοντέλο μηχανικής μάθησης ή συνάρτηση στην Python. Η κύρια διεπαφή εξήγησης της βιβλιοθήκης SHAP δέχεται οποιοδήποτε συνδυασμό ενός μοντέλου και επιστρέφει ένα αντικείμενο υποκλάσης που υλοποιεί το συγκεκριμένο αλγόριθμο εκτίμησης που επιλέχθηκε.

5.7 Αποτελέσματα (αναλυτικά)

Για την επίτευξη υψηλών επιδόσεων στα αποτελέσματα ακρίβειας και επεξηγηματικότητας, κατά την διάρκεια της διπλωματικής εργασίας διεξάχθηκαν διαφόρων ειδών πειράματα κάνοντας τροποποιήσεις στις υπερπαραμέτρους (**hyperparameters**). Παρακάτω αναλύονται τα στάδια που ακολουθήθηκαν για την πραγματοποίηση των πειραματικών δοκιμών, καθώς και τα αποτελέσματα που συλλέχθηκαν.

5.7.1 Ακρίβεια

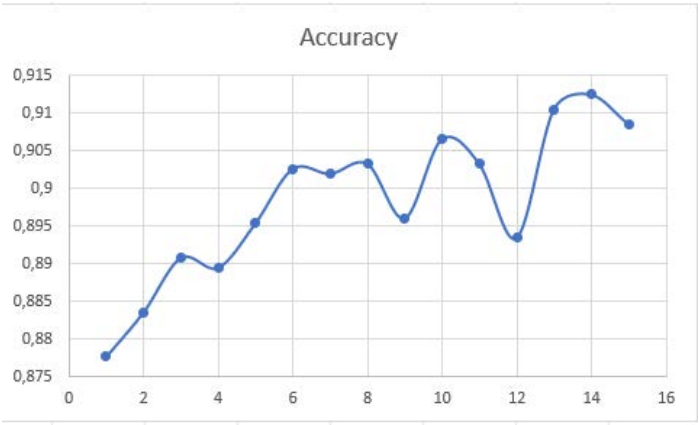
Η ακρίβεια ενός αλγορίθμου ταξινόμησης μηχανικής μάθησης είναι ένας τρόπος μέτρησης του πόσο συχνά ο αλγόριθμος ταξινομεί σωστά ένα data point. Ουσιαστικά είναι ο αριθμός των σωστά προβλεπόμενων data points επί του συνόλου των data points.

Η ακρίβεια που πετύχαμε για το πρώτο Dataset με τις 4 κατηγορίες ειδήσεων είναι της τάξεως του 91%. Μία αρκετά ικανοποιητική αλλά και αντιπροσωπευτική απόδοση για ένα μοντέλο σαν το BERT. Στις παρακάτω εικόνες απεικονίζονται η ανοδική πορεία του accuracy στην πάροδο των εποχών σε αριθμητική μορφή, σε μορφή διαγράμματος αλλά και σε διάγραμμα που μπορούμε με ευκολία να το συγκρίνουμε με το Validation Loss.

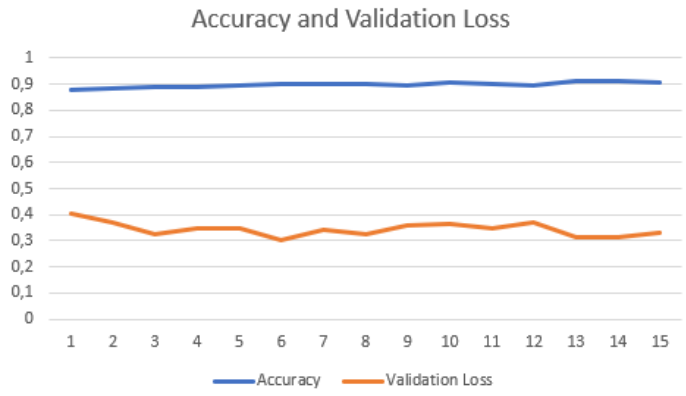
[760/760 07:37, Epoch 2/2]

Step	Training Loss	Validation Loss	Accuracy
50	0.811500	0.405174	0.877632
100	0.443600	0.369865	0.883553
150	0.383300	0.326838	0.890789
200	0.395700	0.347658	0.889474
250	0.331300	0.348291	0.895395
300	0.353500	0.302980	0.902632
350	0.323400	0.340195	0.901974
400	0.271700	0.322992	0.903289
450	0.216900	0.359095	0.896053
500	0.174000	0.367010	0.906579
550	0.209200	0.347742	0.903289
600	0.193900	0.372774	0.893421
650	0.228900	0.312157	0.910526
700	0.171500	0.312987	0.912500
750	0.145100	0.328739	0.908553

Εικόνα 63: Ακρίβεια μοντέλου για AG Dataset



Εικόνα 64 : Accuracy σε διάγραμμα



Εικόνα 65: Accuracy vs Validation Loss σε διάγραμμα

Αντίστοιχα χρησιμοποιώντας τον ίδιο κώδικα και για το δεύτερο Dataset με τις θετικές και αρνητικές κριτικές ταινιών πετύχαμε 88% ακρίβεια.

[500/500 26:30, Epoch 0/1]

Step	Training Loss	Validation Loss	Accuracy
50	0.569000	0.602925	0.693100
100	0.419800	0.377158	0.839300
150	0.416700	0.368481	0.850000
200	0.382800	0.392751	0.837300
250	0.391500	0.350062	0.845200
300	0.385600	0.346033	0.859100
350	0.331600	0.323237	0.860700
400	0.340400	0.318419	0.865300
450	0.328200	0.317428	0.870500
500	0.316100	0.316222	0.870900

Εικόνα 66: Ακρίβεια μοντέλου για IMDB Dataset

5.7.1.1 Υπερπαράμετροι

Η βέλτιστη επιλογή των τιμών των υπερπαραμέτρων εξαρτάται από το εκάστοτε πρόβλημα. Δεδομένου ότι οι αλγόριθμοι, οι στόχοι, οι τύποι

δεδομένων και οι όγκοι δεδομένων αλλάζουν σημαντικά από το ένα έργο στο άλλο, δεν υπάρχει μια μοναδική βέλτιστη επιλογή για τις τιμές των υπερπαραμέτρων που να ταιριάζει σε όλα τα μοντέλα και όλα τα προβλήματα. Οι συγκεκριμένοι υπερπαράμετροι που επιλέχθηκαν είναι :

- Αριθμός Epochs
- Early_Stopping
- Learning Rate
- Batch size

5.7.1.1.1 Epochs

Μια εποχή ορίζεται ως ο συνολικός αριθμός επαναλήψεων όλων των δεδομένων εκπαίδευσης σε έναν κύκλο για την εκπαίδευση του μοντέλου μηχανικής μάθησης. Ένας άλλος τρόπος για να οριστεί μια εποχή είναι ο αριθμός των περασμάτων ενός συνόλου δεδομένων εκπαίδευσης γύρω από έναν αλγόριθμο. Ένα πέρασμα μετράται όταν το σύνολο δεδομένων έχει κάνει τόσα προς τα εμπρός όσα και προς τα πίσω περάσματα. Ο αριθμός των εποχών θεωρείται υπερπαράμετρος και καθορίζει τον αριθμό των φορών που πρέπει να περάσει ολόκληρο το σύνολο δεδομένων από τον αλγόριθμο μάθησης.

Στην παρούσα εργασία ορίστηκαν 10 epochs για την εκπαίδευση του μοντέλου, έτσι ώστε να έχει την δυνατότητα κάλυψης όλων των σημείων (coverage). Στο μοντέλο μας όμως έχουμε προσθέσει την λειτουργία του EarlyStopping η οποία κάνει το μοντέλο να σταματήσει την εκπαίδευση όταν μια μετρική που έχουμε ορίσει έχει σταματήσει να βελτιώνεται, για την αποφυγή overfitting. Τις περισσότερες φορές το μοντέλο θα σταματούσε να εκπαιδεύεται μεταξύ 2 και 3 epochs.

5.7.1.1.2 EarlyStopping

Το Early Stopping είναι μια τεχνική που χρησιμοποιείται για την αποφυγή του overfitting. Περιλαμβάνει τη διακοπή της εκπαίδευσης ενός μοντέλου όταν η απόδοσή του σε ένα validation set αρχίζει να υποβαθμίζεται ή δεν βελτιώνεται πέρα από ένα ορισμένο όριο. Αυτό αποτρέπει το μοντέλο από το να εξειδικευτεί υπερβολικά στα δεδομένα εκπαίδευσης και έχει ως αποτέλεσμα καλύτερες επιδόσεις σε νέα δεδομένα. Στο παράδειγμά μας έχουμε ορίσει το early_stopping_patience=3 και το early_stopping_threshold=0.01. Δηλαδή αν ο αλγόριθμος τρέξει 3 συνεχόμενες φορές και η διαφορά της απόδοσης του validation set είναι μικρότερη η ίση του 0.01 θα γίνει διακοπή της διαδικασίας.

5.7.1.1.3 Learning Rate

Η υπερπαράμετρος Learning Rate ήταν μία από τις βασικότερες παράμετρους στην οποία έγιναν οι περισσότερες τροποποιήσεις. Στους

παρακάτω πίνακες φαίνονται οι αποδόσεις του μοντέλου αν αυξήσουμε ή μειώσουμε την τιμή τους για την χρήση και των 2 Dataset.

Ακρίβεια για AG News Classification Dataset

Πίνακας 1: Best Learning Rate για AG Dataset

Learning Rate	Epochs	Time(min)	Accuracy
2e-4	2	00:09	**91%**
2e-5	2	00:09	90%
3e-5	3	00:12	90%
4e-5	3	00:13	89%
5e-5	2	00:10	90%

Ακρίβεια για IMDB Dataset

Πίνακας 2: Best Learning Rate για IMDB Dataset

Learning Rate	Epochs	Time(min)	Accuracy
2e-4	1	00:20	**88%**
2e-5	2	00:29	84%
3e-5	2	00:28	86%
4e-5	2	00:30	86%
5e-5	2	00:28	85%

Από τους παραπάνω πίνακες παρατηρούμε λοιπόν ότι η πιο ιδανική τιμή και για τα 2 Dataset είναι Lr=2e-4 το οποίο μας επιφέρει για το AG Dataset με την μεγαλύτερη απόδοση (91%) και για το IMDB Dataset (88%).

5.7.1.1.4 Batch siz

Ένα batch size με τιμή 32 σημαίνει ότι 32 δείγματα από το σύνολο δεδομένων εκπαίδευσης θα χρησιμοποιηθούν για την εκτίμηση της κλίσης σφάλματος (error gradient) πριν από την ενημέρωση των βαρών του μοντέλου. Το μέγεθος του batch επηρεάζει ορισμένους δείκτες, όπως ο συνολικός χρόνος εκπαίδευσης, ο χρόνος εκπαίδευσης ανά εποχή, η ποιότητα του μοντέλου και

άλλα παρόμοια. Συνήθως, επιλέγουμε το μέγεθος του batch_size ως δύναμη του δύο, στο εύρος μεταξύ 16 και 512. Αλλά γενικά, το μέγεθος 32 είναι μια καλή αρχική επιλογή.

Στο πρόβλημά μας κάναμε χρήση της τιμής 32 και παρατηρήσαμε ότι πέρα από την μείωση του χρόνου εκτέλεσης καταφέραμε να πετύχουμε και υψηλότερες τιμές ακρίβειας, σε σχέση με την τιμή 16.

5.7.2 Αποτελέσματα επεξηγηματικότητας για AG Dataset

Για κάθε Dataset έχουμε εφαρμόσει και τα δύο επεξηγηματικά μοντέλα για να μπορούμε να συγκρίνουμε με μεγαλύτερη ευκολία τα αποτελέσματα.

- **LIME**

Όταν χρησιμοποιούμε το LIME για να εξηγήσουμε μια πρόβλεψη μοντέλου μηχανικής μάθησης για ένα συγκεκριμένο δείγμα δεδομένων, η έξοδος που λαμβάνουμε είναι μια λίστα επεξηγήσεων. Κάθε επεξήγηση αντιπροσωπεύει την συνεισφορά κάθε χαρακτηριστικού στην τελική πρόβλεψη του μοντέλου για το συγκεκριμένο δείγμα. Αυτό παρέχει τοπική ερμηνευσιμότητα, καθώς μας δίνει μια κατανόηση για το πώς το μοντέλο λαμβάνει την απόφασή του βάσει των χαρακτηριστικών του δείγματος. Επιπλέον, μας επιτρέπει να προβλέψουμε το ποσό που θα αλλάξει η πρόβλεψη αν αλλάξουμε τα χαρακτηριστικά του δείγματος.

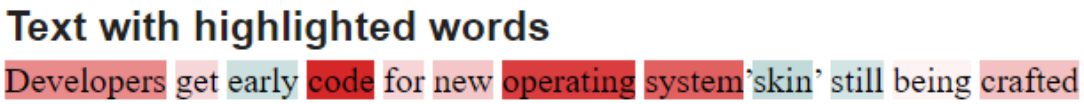
Εικόνα 67: LIME output

Στην παραπάνω εικόνα βλέπουμε το αποτέλεσμα του αλγορίθμου LIME για το πρώτο Dataset (κατηγορία ειδήσεων) αν με ποσοστό ακρίβειας του αλγορίθμου BERT 91% και text=”Developers get early code for new operating system’skin’ still being crafted”, ένα κείμενο από την 4^{η} κατηγορία ειδήσεων (τεχνολογίας), τρέξουμε το πρόγραμμα. Στα αριστερά μπορούμε να διακρίνουμε τις πιθανότητες πρόβλεψης και των 4^{ων} κατηγοριών, καθώς και την χρωματική διαφοροποίηση για καλύτερο διαχωρισμό. Στην περίπτωση μας μπορούμε να διακρίνουμε με ευκολία ότι η κατηγορία με την μεγαλύτερη πιθανότητα είναι η τέταρτη (ειδήσεις με θέμα την τεχνολογία) με ποσοστό 97%. Ακόμη, όπως βλέπουμε δεξιά από τις πιθανότητες πρόβλεψης, γίνεται λεπτομερής ανάλυση της θετικής η αρνητικής επιρροής που συνεισφέρει κάθε λέξη για την κάθε κατηγορία.

Ο λόγος ύπαρξης αυτών των γραφικών είναι για να μπορούμε να δώσουμε στον χρήστη μια εμφανή λεπτομερή επεξήγηση από την οποία μπορεί να αποκομίσει πληροφορίες για την τελική επιλογή του αλγορίθμου. Για

παράδειγμα, οι λέξεις code, operating, system και Developers είναι οι λέξεις που συνεισφέραν πιο πολύ θετικά για την 4η κατηγορία ειδήσεων με ποσά 0,08, 0,07, 0,06, 0,04 αντίστοιχα, σε αντίθεση με όλες τις άλλες κατηγορίες, όπου βλέπουμε ότι όλες αυτές οι λέξεις, συνεισφέραν αρνητικά.

Σχολιάζοντας τα αποτελέσματα, μπορούμε να παρατηρήσουμε ότι η λέξη "code" είχε την μεγαλύτερη βαρύτητα διότι σχετίζεται άμεσα με την κατηγορία της τεχνολογίας και αντιστοίχως παίζει τον μεγαλύτερο αρνητικό ρόλο για τις υπόλοιπες. Κάτι τέτοιο μας δείχνει ότι ο κώδικας λειτουργεί σωστά και δίνει την κατάλληλη βαρύτητα στις λέξεις.

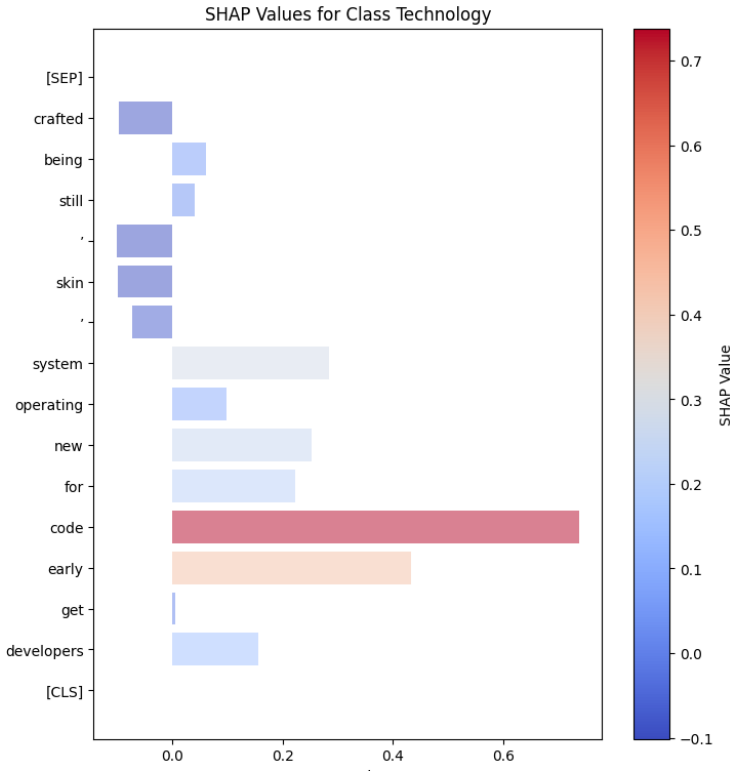


Εικόνα 68: Κείμενο με υπογραμμισμένες λέξεις

Τέλος βλέπουμε το κείμενο που έχουμε εισάγει με υπογραμμισμένες χρωματικές διαφοροποιήσεις με βάση την θετικότητα ή αρνητικότητα της λέξης για την κάθε κατηγορία. Πέρα από την διαφορά στο χρώμα με βάση την κατηγορία, μπορούμε να διακρίνουμε και διαφορά στη βαθύτητα του χρώματος με βάση το ποσοστό της κάθε λέξης.

- **SHAP**

Όπως και με το LIME, όταν χρησιμοποιούμε το SHAP για να εξηγήσουμε μια πρόβλεψη μοντέλου μηχανικής μάθησης, μπορούμε να λάβουμε μια λίστα επεξηγήσεων που αναπαριστούν τη συνεισφορά κάθε χαρακτηριστικού στην τελική πρόβλεψη. Ωστόσο, οι επεξηγήσεις του SHAP βασίζονται στη θεωρία των παιγνίων και αναπαριστούν την αναμενόμενη αξία της συνεισφοράς του κάθε χαρακτηριστικού στην πρόβλεψη, λαμβάνοντας υπόψη τη σημασία του χαρακτηριστικού και τις διακυμάνσεις των χαρακτηριστικών στο υπόλοιπο σύνολο των δεδομένων. Επομένως, το SHAP παρέχει τόσο τοπική όσο και γενική ερμηνευσιμότητα, επιτρέποντας στους χρήστες να κατανοήσουν πώς λειτουργεί το μοντέλο σε επίπεδο δεδομένων και συνολικά.



Εικόνα 69: SHAP output (Technology)

Στην παραπάνω εικόνα απεικονίζονται τα αποτέλεσμα του αλγορίθμου SHAP αν πάρουμε ως input τα ίδια δεδομένα που πήραμε και στο παράδειγμα με το LIME. Ακόμη εμφανίζουμε και την βαρύτητα της κάθε λέξης για την κατηγορία με το μεγαλύτερο ποσοστό.

```
[CLS]: 0.0000
developers: 0.1546
get: 0.0051
early: 0.4332
code: 0.7382
for: 0.2224
new: 0.2520
operating: 0.0985
system: 0.2838
’: -0.0735
skin: -0.0997
’: -0.1014
still: 0.0394
being: 0.0602
crafted: -0.0972
[SEP]: 0.0000
```

Εικόνα 70: Βάρη λέξεων SHAP

Παρατηρώντας τα αποτελέσματα και συγκρίνοντας τους 2 αλγορίθμους μπορούμε να διακρίνουμε ότι οι λέξεις που παίζουν θετικό ρόλο έχουν πολύ υψηλότερα ποσά βαρύτητας από τον αλγόριθμο LIME καθιστώντας το SHAP πιο επεξηγήσιμο αλγόριθμο με μικρότερη πιθανότητα επιλογής λανθασμένης κλάσης. Για παράδειγμα η λέξη “code” έχει βαρύτητα της τάξεως του 0,73 σε αντίθεση με το LIME οπού έχει 0,08.

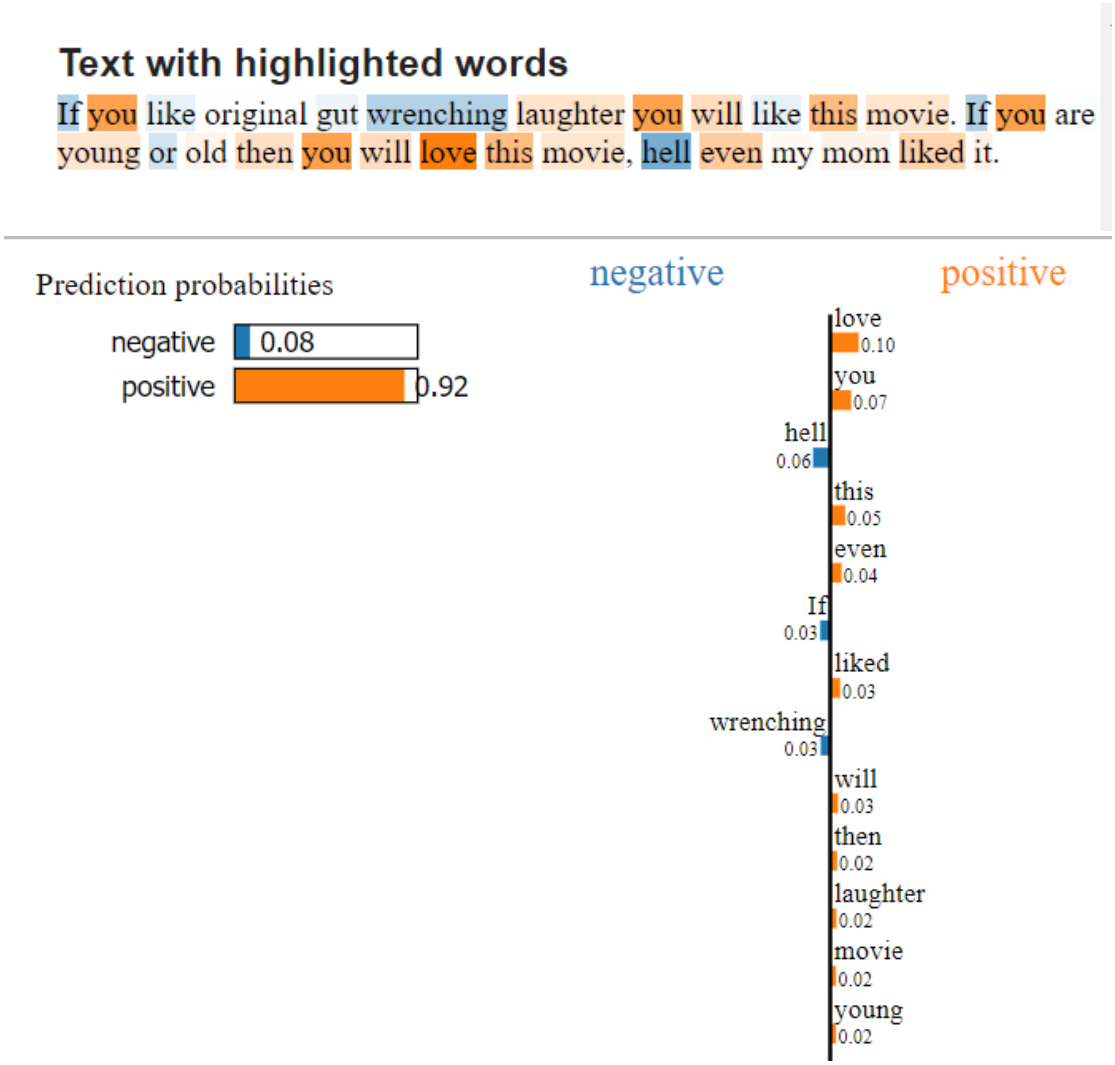
5.7.3 Αποτελέσματα επεξηγηματικότητας για IMDB Dataset

Αντίστοιχα, για το IMDB Dataset χρησιμοποιήσαμε και τους δύο αλγορίθμους, για να μπορέσουμε να συγκρίνουμε και να σχολιάσουμε τα αποτελέσματα μεταξύ του LIME και του SHAP. Παρακάτω παρουσιάζουμε τα δύο πιο σημαντικά αποτελέσματα των αλγορίθμων που μας δίνουν πληροφορίες για τον τρόπο λειτουργίας τους και τις διαφορές τους.

Τα δύο παραδείγματα θα είναι μία θετική και μία αρνητική κριτική ταινιών από το Dataset μας.

- **LIME**

Για το πρώτο παράδειγμα χρησιμοποιήσαμε μία θετική κριτική από το Dataset, (“If you like original gut wrenching laughter you will like this movie. If you are young or old then you will love this movie, hell even my mom liked it.”). Με ποσοστό ακρίβειας του αλγορίθμου BERT 88% ο αλγόριθμος LIME κατάφερε να αναγνωρίσει ότι η κριτική είναι θετική με πιθανότητα πρόβλεψης της τάξεως του 92%. Στην παρακάτω εικόνα βλέπουμε τα αποτελέσματα του αλγορίθμου χρησιμοποιώντας χρωματικές διαφοροποιήσεις για την καλύτερη επεξηγηματικότητα των αποτελεσμάτων. Στην δεξιά στήλη διακρίνουμε ποιες λέξεις έπαιξαν θετικό ή αρνητικό ρόλο για την απόφαση του αλγορίθμου στην κλάση με την μεγαλύτερη πρόβλεψη πιθανότητας.

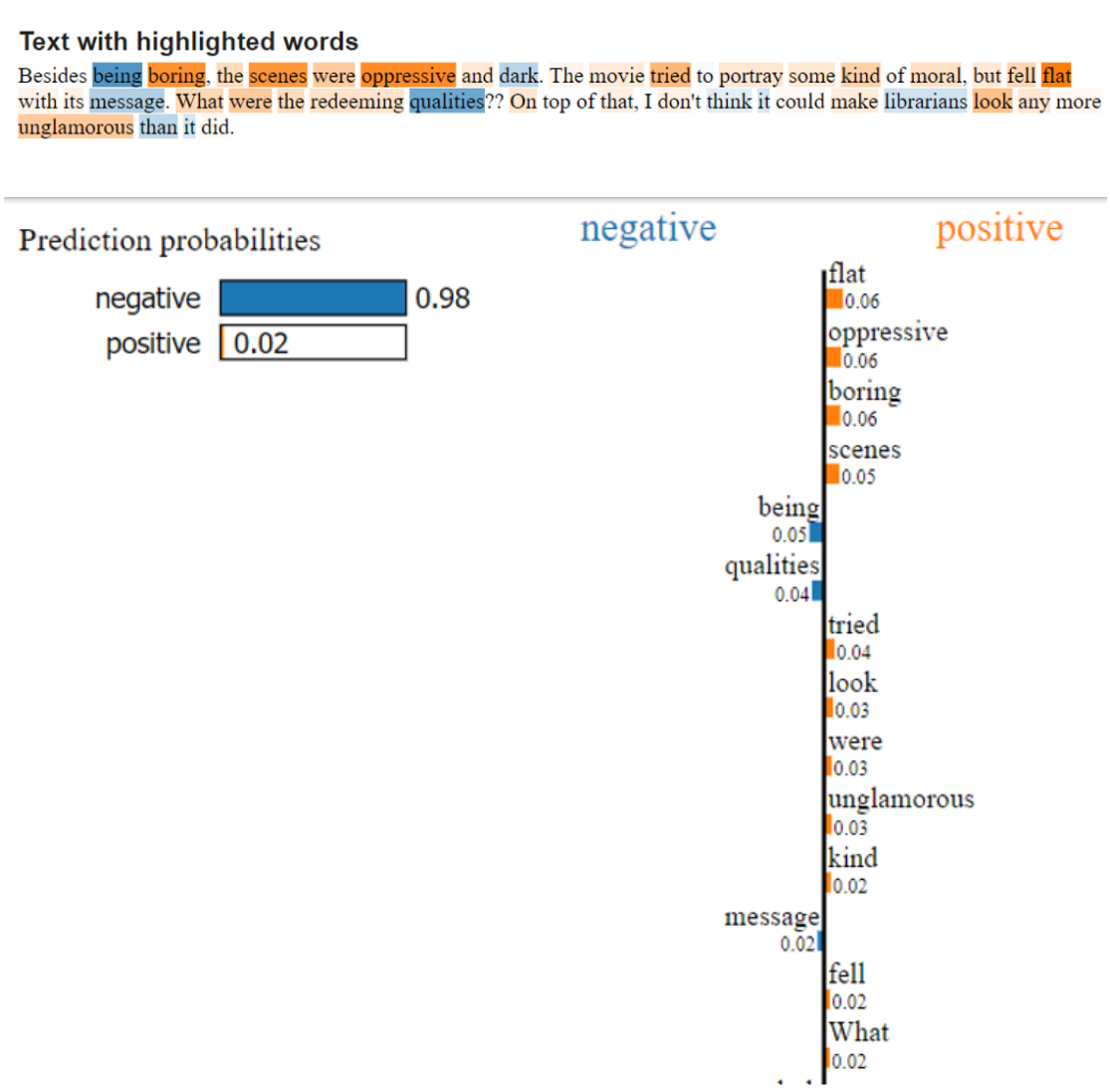


Εικόνα 71: LIME output για θετική πρόταση

Παρατηρώντας τα αποτελέσματα βλέπουμε αμέσως ότι η λέξη "love" παίζει θετικό ρόλο με αρκετά υψηλό ποσοστό, κάτι το οποίο είναι σχετικά λογικό μιας και η συγκεκριμένη λέξη γενικότερα είναι θετική. Αντίθετώς η λέξη "hell" ενώ συνήθως χαρακτηρίζεται ως αρνητική λέξη, στην συγκεκριμένη περίπτωση λόγο των συμφραζόμενων παίζει θετικό ρόλο. Επομένως διαπιστώνουμε ότι ο αλγόριθμος εξετάζει την κάθε λέξη μεμονωμένα περισσότερο από ότι θα έπρεπε. Γενικότερα καταλαβαίνουμε ότι μπορεί να κάνει κάποια λάθη σε ειδικές λέξεις, αλλά μπορεί να αναγνωρίσει την θετικότητα ή αρνητικότητα όλης της πρότασης.

Για το δεύτερο παράδειγμα χρησιμοποιήσαμε μία αρνητική κριτική από το Dataset, ("Besides being boring, the scenes were oppressive and dark. The movie tried to portray some kind of moral, but fell flat with its message. What were the redeeming qualities?? On top of that, I don't think it could make librarians look any more unglamorous than it did.."). Με ποσοστό ακρίβειας του

αλγορίθμου BERT 88% ο αλγόριθμος LIME κατάφερε να αναγνωρίσει ότι η κριτική είναι αρνητική με πιθανότητα πρόβλεψης της τάξεως του 98%. Στην παρακάτω εικόνα βλέπουμε τα αποτελέσματα του αλγορίθμου χρησιμοποιώντας χρωματικές διαφοροποιήσεις για την καλύτερη επεξηγηματικότητα των αποτελεσμάτων. Όμοια με το θετικό παράδειγμα, βλέπουμε με πορτοκαλί χρωματισμό τις λέξεις που έπαιξαν θετικό ρόλο για τον χαρακτηρισμό τις πρότασης ως αρνητική και αντίστοιχα με μπλε αυτές που έπαιξαν αρνητικό ρόλο.



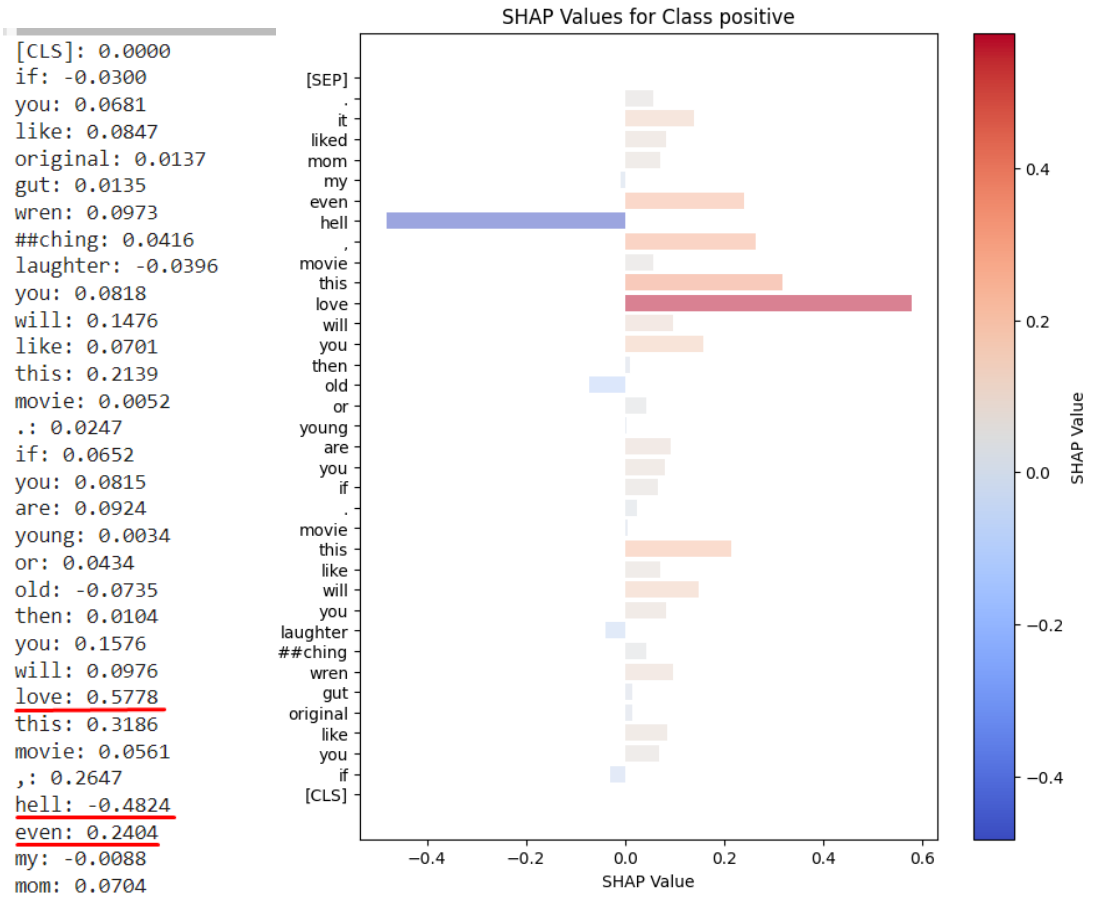
Εικόνα 72: LIME output για αρνητική πρόταση

Παρατηρούμε από το παραπάνω παράδειγμα ότι πετυχαίνουμε αρκετά μεγαλύτερο ποσοστό ακρίβειας και αυτό οφείλεται στο γεγονός ότι η πρόταση έχει πολλές λέξεις, επομένως και περισσότερα δεδομένα με αποτέλεσμα να

έχουμε μία πιο ακριβής πρόβλεψη. Γενικότερα παρατηρούμε ότι δίνει στις περισσότερες λέξεις τις σωστές τιμές βαρύτητας με εξαίρεση ίσως την λέξη “dark” η οποία ενώ έχει αρνητική χρήση στο κείμενο, ο αλγόριθμος την εκτιμάει ως θετική.

- **SHAP**

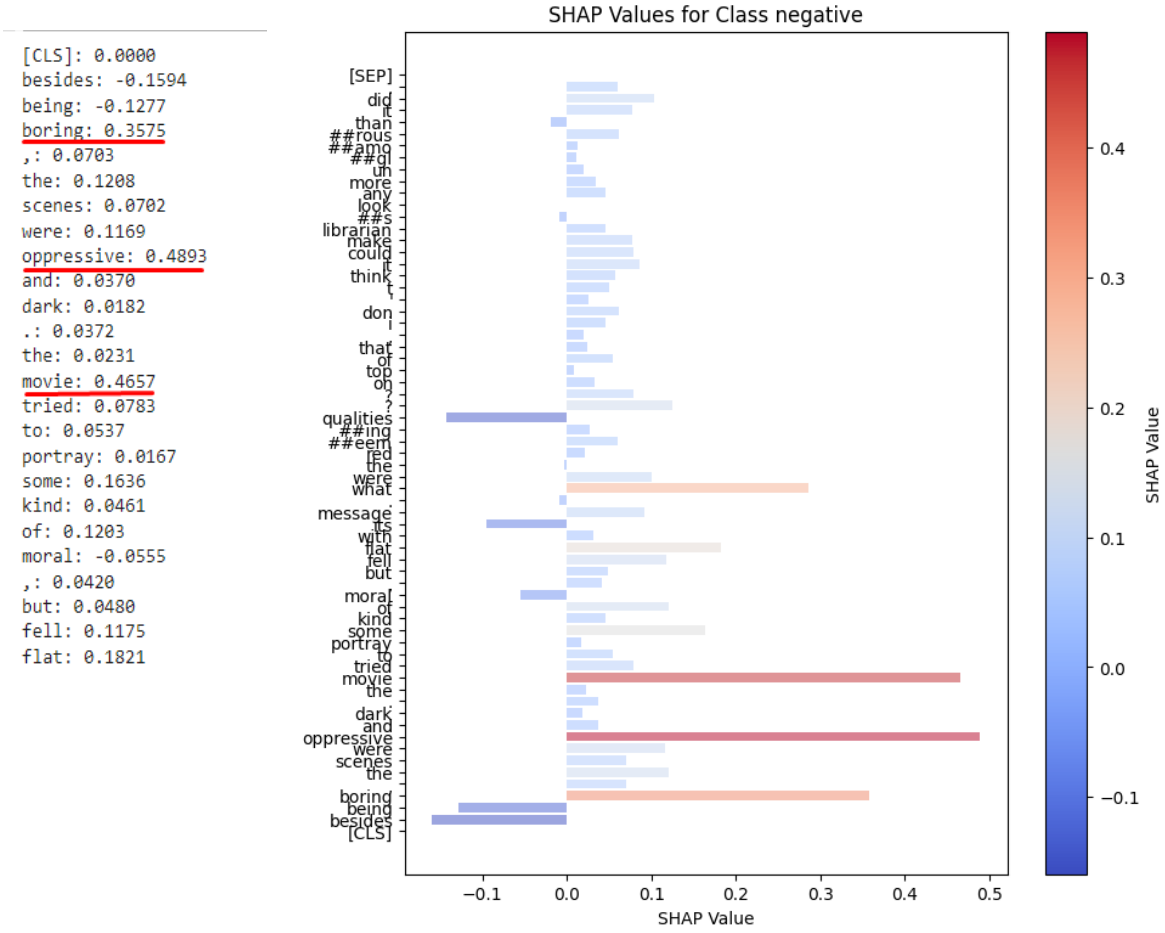
Για την θετική κριτική του αλγορίθμου SHAP χρησιμοποιήσαμε τα ίδια δεδομένα με αυτά που χρησιμοποιήσαμε στον αλγόριθμο LIME. Παρακάτω απεικονίζονται τα αποτελέσματα του αλγορίθμου.



Εικόνα 73: SHAP output για θετική πρόταση

Για την προσέγγιση της επεξήγησης των αποτελεσμάτων μέσω του SHAP μπορούμε να διακρίνουμε ότι υπάρχει μεγαλύτερη ακρίβεια στην εύρεση των λέξεων που επηρεάζουν θετικά ή αρνητικά την πρόταση. Για παράδειγμα οι λέξεις “love, hell, even”, επηρεάζουν με μεγαλύτερη τιμή βαρύτητας θετικά ή και αρνητικά αντίστοιχα την πρόταση σε αντίθεση με το LIME.

Για την αρνητική κριτική του αλγορίθμου SHAP χρησιμοποιήσαμε τα ίδια δεδομένα με αυτά που χρησιμοποιήσαμε στον αλγόριθμο LIME. Παρακάτω απεικονίζονται τα αποτελέσματα του αλγορίθμου.



Εικόνα 74: SHAP output για αρνητική πρόταση

Αντίστοιχα και με την θετική κριτική του SHAP παρατηρούμε ότι υπάρχει μεγαλύτερη ακρίβεια στην επεξήγηση κάθε λέξης ξεχωριστά. Για παράδειγμα οι λέξεις “oppressive” και “boring”, έχουν θετική βαρύτητα 0.48 και 0.35 αντίστοιχα, ενώ με τον αλγόριθμο LIME έχουνε 0.06.

5.7.4 Word Cloud

Το σύννεφο λέξεων είναι μια οπτική αναπαράσταση δεδομένων κειμένου που εμφανίζει τις πιο συχνά χρησιμοποιούμενες λέξεις ή φράσεις σε ένα συγκεκριμένο σύνολο δεδομένων. Οι λέξεις είναι συνήθως τοποθετημένες με τρόπο που τονίζει τη συχνότητα ή τη σημασία τους, με μεγαλύτερες και πιο εμφανείς λέξεις να υποδεικνύουν εκείνες που εμφανίζονται συχνότερα.

Τα σύννεφα λέξεων χρησιμοποιούνται συχνά για να συνοψίσουν και να επικοινωνήσουν τα βασικά θέματα και ιδέες σε ένα μεγάλο σώμα κειμένου και μπορούν να αποτελέσουν ένα χρήσιμο εργαλείο για τον εντοπισμό μοτίβων,

τάσεων και άλλων πληροφοριών που μπορεί να μην είναι άμεσα εμφανείς μόνο από μια ανάλυση κειμένου. Στο πλαίσιο των εξηγήσιμων μοντέλων, τα νέφη λέξεων μπορούν να χρησιμοποιηθούν για τον εντοπισμό των σημαντικότερων παραγόντων ή μεταβλητών που οδηγούν σε ένα συγκεκριμένο αποτέλεσμα ή απόφαση.

Για το παράδειγμά μας θα δημιουργήσουμε ένα word cloud για κάθε θετική ή αρνητική επιρροή κάθε κλάσης για το LIME αλλά και για το SHAP και στα 2 Datasets έτσι ώστε να μπορέσουμε να συγκρίνουμε τις σημαντικότερες λέξεις από κάθε κλάση στους 2 αλγορίθμους.

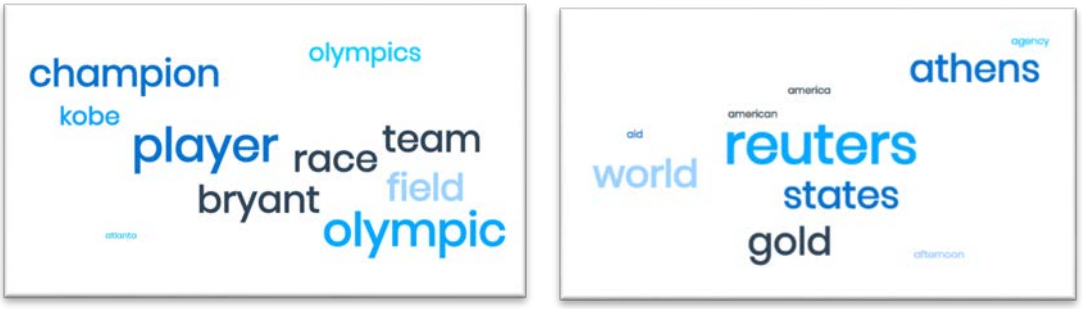
Επομένως αφού υλοποιήσαμε μία συνάρτηση η οποία από τα πρώτα 100 κείμενα των Dataset αποθηκεύει σε αρχεία τις λέξεις οι οποίες έχουν τιμή βάρους μεγαλύτερη του 0,03, χρησιμοποιήσαμε αυτά τα αρχεία για την δημιουργία των word cloud και είχαμε τα εξής αποτελέσματα :

Για τον αλγόριθμο LIME στο AG Dataset έχουμε στις εικόνες που βρίσκονται στην αριστερή πλευρά τις λέξεις που έπαιξαν θετική επιρροή για την επιλογή της συγκεκριμένης κλάσης και στις εικόνες που είναι στην δεξιά πλευρά αυτές που έπαιξαν αρνητική.

- AG Dataset
  - World

Εικόνα 75: Positive for World in LIME

  - Athletics



Εικόνα 76 :Positive for Athletics LIME

- Business

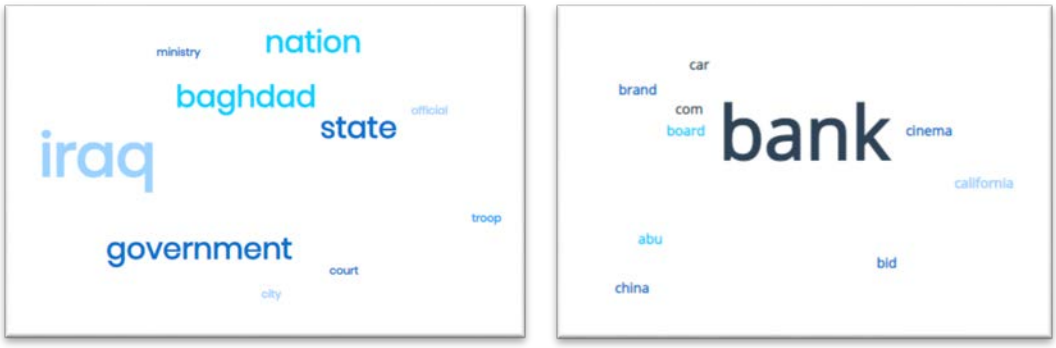
Εικόνα 77:Positive for Business in LIME

- Technology

Εικόνα 78: Positive for Technology in LIME

Για τον αλγόριθμο SHAP στο AG Dataset έχουμε τα εξής αποτελέσματα:

- World



Εικόνα 79: Positive for World in SHAP

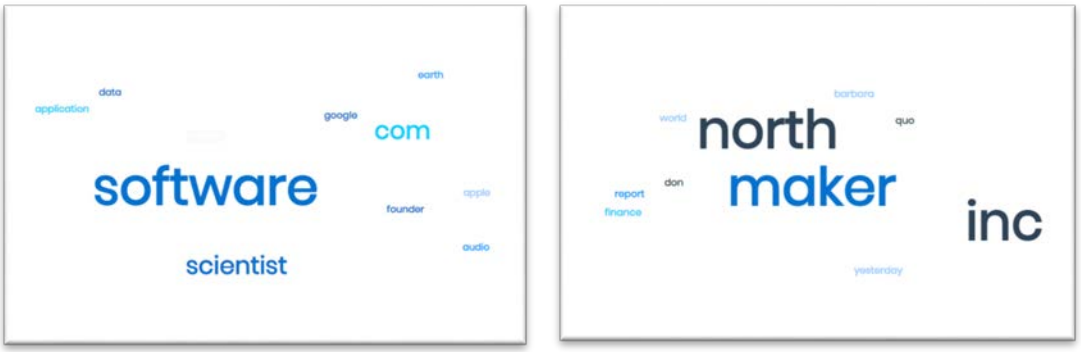
- Athletics

Εικόνα 80: Positive for Athletics in SHAP

- Business

Εικόνα 81: Positive for Business in SHAP

- Technology



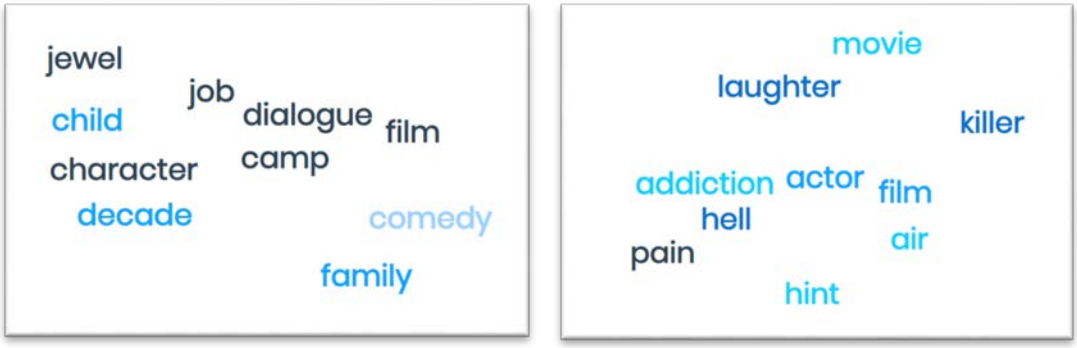
Εικόνα 82: Positive for Technology in SHAP

- IMDB Dataset

Για τον αλγόριθμο LIME έχουμε τα εξής αποτελέσματα :

Εικόνα 83 : Positive in LIME

Για τον αλγόριθμο SHAP έχουμε τα εξής αποτελέσματα :



Εικόνα 84: Positive in SHAP

Εξετάζοντας τα αποτελέσματα μπορούμε να παρατηρήσουμε ότι και στις 2 μεθόδους είχαμε επιτυχημένα αποτελέσματα με αρκετά εύστοχες λέξεις στα περισσότερα Word Cloud. Ωστόσο μπορούμε να διακρίνουμε ότι και για τα 2 Datasets το LIME έχει παράξει λέξεις πιο σχετικές για την κλάση, κάτι που μας αποδεικνύει ότι είναι αρκετά αξιόπιστο μοντέλο.

5.7.5 Σχολιασμός αποτελεσμάτων

Παρατηρούμε ότι το μοντέλο SHAP αποδίδει καλύτερα σε σχέση με το μοντέλο LIME διότι καταφέρνει να αναγνωρίσει με μεγαλύτερη επιτυχία μεμονωμένες λέξεις, ως αποτέλεσμα να επεξηγεί καλύτερα στον χρήστη τον τρόπο εύρεσης του αποτελέσματος. Επιπρόσθετα ο χρόνος εκτέλεσης των 2 αλγορίθμων είναι σχετικά ίδιος, αν τρέξουμε τον αλγόριθμο LIME με 300 num_samples (όπως έχουμε κάνει και στα παραδείγματα). Αν όμως θέλουμε να αυξήσουμε την επεξηγηματικότητα του LIME αλγορίθμου, θα πρέπει να αυξήσουμε και τον αριθμό των samples με αποτέλεσμα να καθιστά το LIME πιο χρονοβόρο μοντέλο από το SHAP.

6 *Συμπεράσματα*

Η Τεχνητή Νοημοσύνη (ΤΝ) έχει φέρει επανάσταση στον τρόπο με τον οποίο προσεγγίζουμε πολλά προβλήματα και έχει οδηγήσει σε ανακαλύψεις σε διάφορους τομείς, συμπεριλαμβανομένης της Ταξινόμησης Κειμένου. Η βαθιά μάθηση έχει επιτρέψει την ανάπτυξη ισχυρών μοντέλων που μπορούν να μαθαίνουν και να γενικεύουν από μεγάλα σύνολα δεδομένων. Το BERT το SHAP και το LIME είναι παραδείγματα τέτοιων μοντέλων που έχουν επιδείξει αξιοσημείωτες επιδόσεις σε εργασίες επεξεργασίας φυσικής γλώσσας.

Ωστόσο, καθώς η τεχνητή νοημοσύνη εξελίσσεται, υπάρχει μια αυξανόμενη ανάγκη για επεξηγηματικότητα, δηλαδή την ικανότητα κατανόησης του τρόπου με τον οποίο ένα σύστημα τεχνητής νοημοσύνης καταλήγει στις αποφάσεις του. Η επεξηγηματικότητα είναι ζωτικής σημασίας για τη διασφάλιση της διαφάνειας, της λογοδοσίας και της εμπιστοσύνης στα συστήματα ΤΝ. Καθώς η έρευνα στην ΤΝ συνεχίζει να εξελίσσεται, είναι πιθανό να δούμε να δίνεται μεγαλύτερη έμφαση στην ανάπτυξη επεξηγήσιμων συστημάτων ΤΝ που μπορούν να παρέχουν σαφείς και ερμηνεύσιμες εξηγήσεις για τις αποφάσεις τους.

6.1 Σχολιασμός των αποτελεσμάτων

Κατά τη διάρκεια της εργασίας, εξετάσαμε εκτενώς τους όρους Τεχνητής Νοημοσύνης, Deep Learning και Μηχανικής Μάθησης, και επικεντρωθήκαμε στην ανάλυση των τρόπων ταξινόμησης κειμένων, των αλγορίθμων νευρωνικών δικτύων και των τεχνικών εκπαίδευσής τους. Επιπλέον, αναλύσαμε τα σημαντικότερα επεξηγήσιμα μοντέλα στον τομέα αυτό, δίνοντας έμφαση στον αλγόριθμο LIME και SHAP.

Όσον αφορά το πειραματικό κομμάτι της εργασίας και της υλοποίησης των αλγορίθμων καταφέραμε ύστερα από προ-επεξεργασία, προ-εκπαίδευση και fine tuning του μοντέλου BERT να επιτύχουμε ποσοστό ακρίβειας της τάξεως του 91%. Επιπρόσθετα καταφέραμε να επεξηγήσουμε τον αλγόριθμο BERT με μεγάλη επιτυχία και με τα 2 επεξηγήσιμα μοντέλα (SHAP,LIME) με πολύ υψηλά ποσοστά πρόβλεψης. Η βασική διαπίστωση η οποία έγινε για τα μοντέλα αυτά ύστερα από αρκετά παραδείγματα και από την χρήση του Word Cloud, είναι ότι το SHAP επέφερε καλύτερη ερμηνευσιμότητα από το LIME και είναι πιο κατάλληλο μοντέλο για τον BERT.

Τέλος θα πρέπει να σημειωθεί ότι η ενσωμάτωση των Αλγόριθμων LIME και SHAP στο πρόγραμμά μας, ήταν ένα πολύ απαιτητικό task, διότι τα βαθιά νευρωνικά δίκτυα αντιμετωπίζονται συνήθως ως black box, καθώς είναι δύσκολο να εξηγηθούν οι διαδικασίες που χρησιμοποιούν για την εξαγωγή συμπερασμάτων και των τελικών αποτελεσμάτων τους.

6.2 Μελλοντικές ενέργειες

Συμπερασματικά, η μελέτη μας κατέδειξε την αποτελεσματικότητα του BERT στη βελτίωση της ακρίβειας των εργασιών ταξινόμησης κειμένου. Στα πλαίσια της εργασίας η χρήση του LIME και του SHAP έδειξε πολλά υποσχόμενα αποτελέσματα στην παροχή επεξηγηματικότητας για τα μοντέλα BERT, αλλά απαιτείται περαιτέρω έρευνα για την πλήρη διερεύνηση των δυνατοτήτων του και τη διερεύνηση εναλλακτικών μεθόδων για τη βελτίωση της ερμηνευσιμότητας των μοντέλων. Σαν επόμενα βήματα, προτείνουμε οι μελλοντικές μελέτες να επικεντρωθούν στην ανάπτυξη υβριδικών μοντέλων που εξισορροπούν τα οφέλη της βαθιάς μάθησης με την ανάγκη για ερμηνευσιμότητα, καθώς και στη διερεύνηση της χρήσης του LIME και SHAP σε άλλους τύπους εργασιών επεξεργασίας φυσικής γλώσσας. Επιπλέον, πιο αυστηρές αξιολογήσεις των μεθόδων ερμηνευσιμότητας των μοντέλων, θα διασφαλίσουν την αξιοπιστία και την γενίκευση των μοντέλων σε διαφορετικούς τομείς και σύνολα δεδομένων.

7 Βιβλιογραφία

[1] Camacho Olmedo et al. (eds.), Geomatic Approaches for Modeling Land Change Scenarios, Lecture Notes in Geoinformation and Cartography, https://doi.org/10.1007/978-3-319-60801-3_27

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Kaiser, Łukasz Kaiser, Illia Polosukhin "Attention Is All You Need"

[3] Bryan McCann, James Bradbury, Caiming Xiong, Richard Socher, "Learned in Translation: Contextualized Word Vectors"

[4] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, "Deep contextualized word representations"

[5] Xipeng Qiu , Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai & Xuanjing Huang "Pre-trained Models for Natural Language Processing: A Survey"*

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

[7] Shervin Minaee Nal Kalchbrenner, Erik Cambria, Singapore Narjes Nikzad, Meysam Chenaghlu, Jianfeng Gao," Deep Learning Based Text Classification: A Comprehensive Review"

[8] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," arXiv preprint arXiv:1503.00075, 2015.

[9] X. Zhu, P. Sobihani, and H. Guo, "Long short-term memory over recursive structures," in International Conference on Machine Learning, 2015, pp. 1604–1612.

[10] Y. Kim, "Convolutional neural networks for sentence classification," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014.

[11] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, 2015, pp. 649–657.

[12] J. D. Prusa and T. M. Khoshgoftaar, "Designing a better data representation for deep neural networks and text classification," in Proceedings - 2016 IEEE 17th International Conference on Information Reuse and Integration, IRI 2016, 2016.

[13] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "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, 2016, pp. 1480–1489.

[14] X. Zhou, X. Wan, and J. Xiao, "Attention-based lstm network for cross-lingual sentiment classification," in Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 247–256.

[15] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[16] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional lstm model and inner-attention," arXiv preprint arXiv:1605.09090, 2016.

[17] C. d. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," arXiv preprint arXiv:1602.03609, 2016.

[18] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," arXiv preprint arXiv:1805.04174, 2018.

[19] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 6586–6593

[20] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

[21] S. Sukhbaatar, J. Weston, R. Fergus et al., "End-to-end memory networks," in Advances in neural information processing systems, 2015, pp. 2440–2448.

[22] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in 33rd International Conference on Machine Learning, ICML 2016, 2016.

[23] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," arXiv:1901.11504, 2019

[24] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek and Wojciech Samek "Explainable AI Methods - A Brief Overview"

[25] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "Why Should I Trust You?" Explaining the Predictions of Any Classifier"

[26] Erico Tjoa, and Cuntai Guan, Fellow, IEEE "A Survey on Explainable Artificial Intelligence (XAI): towards Medical XAI"