ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

Μελέτη και υλοποίηση τεχνικών αυτόματης εξαγωγής σημαντικών φράσεων/λέξεων από κείμενο

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

της

Πιερράτου Ελευθερία

(ΑΕΜ 69)

Επιβλέπων καθηγητής: Δημόκας Νικόλαος
Επίκουρος Καθηγητής

Καστοριά, Σεπτέμβρης 2024

0.1 Ευχαριστίες

Ξεκινώντας, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Δημόκα Νικόλαο καθώς χωρίς την εμπιστοσύνη του και τη δυνατότητα που μου προσέφερε για την ελεύθερη επιλογή του θέματος της πτυχιακής δε θα υπήρχε αυτή η εργασία. Μέσω αυτής έμαθα πολλά πράγματα και τον ευχαριστώ που μου έδωσε την ευκαιρία να εμβαθύνω τις γνώσεις μου στον συγκεκριμένο τομέα. Επίσης, θα ήθελα να ευχαριστήσω προσωπικά και την καθηγήτριά μου κα. Καλογηράτου Ζαχαρούλα για την αγάπη της, τη στήριξη και την καθοδήγηση τα τελευταία 5 χρόνια.

Από τις ευχαριστίες δε θα μπορούσαν να λείπουν και οι δικοί μου άνθρωποι, η οικογένεια μου, οι φίλοι μου και πρόσωπα που με στήριξαν τόσο οικονομικά όσο και ψυχολογικά καθ' όλη τη διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω το προσωπικό του τμήματος Μαθηματικών του Πανεπιστημίου Δυτικής Μακεδονίας στην Καστοριά, καθώς ήταν πρόθυμοι να βοηθήσουν σε ό,τι χρειαζόταν αμέσως και φυσικά τους καθηγητές μου για όλες τις γνώσεις που μου προσέφεραν.

Πιερράτου Ελευθερία
Καστοριά 2024

0.2 Περίληψη

Στην παρούσα πτυχιακή εργασία παρουσιάζονται τα αποτελέσματα εκτενούς βιβλιογραφικής έρευνας, καθώς και αλγόριθμοι που αναπτύχθηκαν σχετικά με τις μεθόδους υλοποίησης τεχνικών αυτόματης εξαγωγής σημαντικών λέξεων και φράσεων από κείμενα.

Η εργασία αναπτύσσεται σε τέσσερα κεφάλαια. Στο $1^ο$ κεφάλαιο παρουσιάζονται οι λόγοι που καθιστούν σημαντικό το αντικείμενο της εργασίας καθώς και τα προβλήματα με τα οποία σχετίζεται. Στο $2^ο$ κεφάλαιο παρουσιάζονται οι βασικές έννοιες, μέθοδοι και τεχνικές που σχετίζονται με το αντικείμενο της εργασίας. Στόχος είναι να κατανοηθούν σύνθετοι ορισμοί και στοιχεία, με έμφαση στα word embeddings και τα Transformers, καθώς αυτά τα δύο σχετίζονται με τις τεχνικές Word2vec και BERT, αντίστοιχα, η μελέτη των οποίων αφορά το κύριο μέρος της εργασίας. Οι δύο αυτές τεχνικές, Word2Vec και BERT, παρουσιάζονται εκτενώς και αναπτύσσεται η λειτουργία τους και τα μοντέλα που χρησιμοποιούν. Σε ό,τι αφορά το Word2Vec, παρουσιάζονται τα μοντέλα CBOW και Skip-Gram, που εστιάζουν στη δημιουργία πολυδιάστατων διανυσμάτων για την αναπαράσταση των λέξεων με βάση τα συμφραζόμενα. Σε ό,τι αφορά το BERT, ως πιο σύγχρονη τεχνική που εκμεταλλεύεται το μετασχηματιστικό μοντέλο (Transformer), μελετάται το πώς η τεχνική επιχειρεί να κατανοήσει τη σημασία των λέξεων στο ευρύτερο πλαίσιο της πρότασης.

Στο $3^ο$ κεφάλαιο, προκειμένου να κατανοηθεί καλύτερα η τεχνική Word2vec, δίνονται παραδείγματα κώδικα με εφαρμογές στην γλώσσα προγραμματισμού Python. Κάθε παράδειγμα εξετάζει διαφορετική περίπτωση λειτουργίας της τεχνικής Word2vec, παρουσιάζοντας στην πράξη τις δυνατότητες και τα πλεονεκτήματα της αυτόματης εξαγωγής σημαντικών λέξεων και φράσεων από κείμενα. Επίσης, γίνεται σύγκριση της απόδοσης και της ακρίβειας των δύο μοντέλων εφαρμογής της τεχνικής Word2vec.

Τέλος, στο $4^ο$ κεφάλαιο παρουσιάζονται αποτελέσματα και συμπεράσματα που προκύπτουν και αφορούν τα πλεονεκτήματα και μειονεκτήματα των δύο τεχνικών και των μοντέλων που τις υποστηρίζουν. Όπως προκύπτει από τις εφαρμογές κώδικα που αναπτύχθηκαν, η επιλογή της κατάλληλης τεχνικής εξαρτάται από τις απαιτήσεις της κάθε εφαρμογής, με το Word2Vec να είναι ιδανικό για εφαρμογές που απαιτούν ταχύτητα και απόδοση, και το BERT να είναι προτιμητέο για εργασίες που απαιτούν βαθύτερη ανάλυση γλωσσικών δομών. Εν κατακλείδι, η υλοποίηση των τεχνικών αυτών μπορεί να προσφέρει πολύτιμες λύσεις σε ένα ευρύ φάσμα εφαρμογών, από τη βελτίωση των μηχανών αναζήτησης μέχρι την ανάλυση κειμένων σε επιχειρηματικό ή επιστημονικό πλαίσιο.

Λέξεις κλειδιά: *word2vec, BERT, CBOW, Skip-Gram, Natural Language Processing*

0.3 Abstract

This thesis presents the results of an extensive literature research, as well as algorithms developed on the methods of implementing techniques for automatic extraction of important words and phrases from texts.

The thesis is developed in four chapters. Chapter 1 presents the reasons that make the subject of the thesis important and the problems to which it is related. Chapter 2 presents the basic concepts, methods and techniques related to the subject of the thesis. The aim is to understand complex definitions and elements, with a focus on word embeddings and Transformers, as these two are related to the Word2vec and BERT techniques, respectively, the study of which is the main part of the thesis. These two techniques, Word2Vec and BERT, are extensively presented and their operation and the models they use are developed. As far as Word2Vec is concerned, the CBOW and Skip-Gram models are presented, which focus on the generation of multidimensional vectors for the contextual representation of words. As for BERT, as a more modern technique exploiting the Transformer model, it is studied how the technique attempts to understand the meaning of words in the broader context of the sentence.

In chapter 3, in order to better understand the Word2vec technique, code examples with applications in the Python programming language are given. Each example examines a different case of the Word2vec technique, demonstrating in practice the capabilities and advantages of automatically extracting important words and phrases from text. The performance and accuracy of the two implementation models of the Word2vec technique are also compared.

Finally, Chapter 4 presents the results and conclusions obtained concerning the advantages and disadvantages of the two techniques and the supporting models. As can be seen from the code applications developed, the choice of the appropriate technique depends on the requirements of each application, with Word2Vec being ideal for applications requiring speed and performance, and BERT being preferable for tasks requiring deeper analysis of language structures. In conclusion, the implementation of these techniques can provide valuable solutions in a wide range of applications, from search engine enhancement to text analysis in a business or scientific context.

Keywords: *word2vec, BERT, CBOW, Skip-Gram, Natural Language Processing*

Περιεχόμενα

0.4 Κατάλογος Σχημάτων

0.5 Κατάλογος Κώδικα

0.6 Ακρωνύμια Πτυχιακής Εργασίας

Δεδομένου ότι η παρούσα εργασία αφορά έρευνα σε κλάδους της πληροφορικής, στο κείμενο συναντώνται αρκετά ακρωνύμια ή αγγλικές ορολογίες. Για την καλύτερη κατανόηση της εργασίας, οι συγκεκριμένες λέξεις/φράσεις θα παραμείνουν στην αγγλική γλώσσα κατά το εύρος του κειμένου. Για τη διευκόλυνση της κατανόησης του αναγνώστη, ωστόσο, παρατίθεται το παρακάτω ένα λεξικό επεξήγησης αυτών.

ΛΕΞΕΙΣ/ΦΡΑΣΕΙΣ	**ΕΛΛΗΝΙΚΗ ΟΡΟΛΟΓΙΑ**
Supervised learning	Επιβλεπόμενη μάθηση
Unsupervised learning	Μη-επιβλεπόμενη μάθηση
Semi - supervised learning	Μάθηση με ημιεπίβλεψη
Dataset	Σύνολο δεδομένων
Neural network	Νευρωνικό δίκτυο
Input layer	Επίπεδο εισόδου
Hidden layer	Κρυφό επίπεδο
Output layer	Επίπεδο εξόδου
Word Embeddings	Ενσωμάτωση λέξεων
One-hot encoding	Κωδικοποίηση one-hot
Vector analysis	Διανυσματική ανάλυση
Target word	Λέξη-στόχος
Transformer	Μετασχηματιστής
Self attention	Αυτό-προσοχή
Tokenization	Συμβολισμός
Mask	Μάσκα
Text Mining	Εξόρυξη κειμένου
Corpus	Σώμα κειμένων

ΑΚΡΩΝΥΜΙΑ	**ΕΠΕΞΗΓΗΣΗ**	**ΕΛΛΗΝΙΚΗ ΜΕΤΑΦΡΑΣΗ**
NLP	Natural Language Processing	Επεξεργασία Φυσικής Γλώσσας
AI	Artificial Intelligence	Τεχνητή Νοημοσύνη
ML	Machine Learning	Μηχανική Μάθηση
DL	Deep Learning	Βαθιά Μάθηση
IoT	Internet of Things	Διαδίκτυο των Πραγμάτων
Word2Vec	Word to Vector	—
DNNs	Deep neural networks	Βαθιά νευρωνικά δίκτυα
Shallow NNs	Shallow neural networks	Ρηχά νευρωνικά δίκτυα
CBOW	Common Bag of Words	—
seq2seq	sequence to sequence	“Διαδοχικό προς διαδοχικό”
BERT	Bidirectional Encoder Representations from Transformers	Αναπαραστάσεις κωδικοποιητή διπλής κατεύθυνσης από μετασχηματιστές
MLM	Masked Language Modelling	—
NSP	Next Sentence Prediction	—

0.7 Εισαγωγικό σημείωμα

Η παρούσα Πτυχιακή Εργασία εκπονήθηκε ως μάθημα επιλογής του όγδοου (8ου) εξαμήνου στο πλαίσιο του Προπτυχιακού Προγράμματος Σπουδών του τμήματος Μαθηματικών της Σχολής Θετικών Επιστημών του Πανεπιστημίου Δυτικής Μακεδονίας στην πόλη της Καστοριάς κατά το Εαρινό εξάμηνο του ακαδημαϊκού έτους 2023-2024.

Επιβλέπων καθηγητής κατά τη συγγραφή αυτής ήταν ο Επίκουρος Καθηγητής του τμήματος Πληροφορικής της Σχολής Θετικών Επιστημών του Πανεπιστημίου Δυτικής Μακεδονίας στην πόλη της Καστοριάς και καθηγητής του τμήματος Μαθηματικών κ. Δημόκας Νικόλαος.

Το αντικείμενο μελέτης της πτυχιακής αφορούσε τη μελέτη και υλοποίηση τεχνικών αυτόματης εξαγωγής σημαντικών φράσεων ή/και λέξεων από κείμενο, όπου μεγάλο μέρος αυτής της μελέτης αποτελούσε η έρευνα πάνω στις τεχνικές word2vec και BERT.

Κεφάλαιο 1

Εισαγωγή

1.1 Το Πρόβλημα

Αν κάτι θα μπορούσε να χαρακτηρίσει τον εικοστό (20ο) αιώνα θα ήταν η ανάπτυξη της επιστήμης και της τεχνολογίας. Είτε σχετίζεται με τη μεταφορά του ανθρώπινου είδους είτε τις τηλεπικοινωνίες, η επιστήμη και η τεχνολογία ήρθε για να διευκολύνει την ποιότητα ζωής των ανθρώπων. Μέχρι και σήμερα είναι ορατή η ανάπτυξη σε πολλούς τομείς με τα τελευταία επιτεύγματα να αφορούν το AI, το ML και τομείς της βιολογίας.

Το πρόβλημα που δημιουργήθηκε από αυτήν την ανάπτυξη, ωστόσο, ήταν η μαζική παραγωγή πληροφορίας. Μέχρι και σήμερα, παράγεται καθημερινά πολύ μεγάλος όγκος δεδομένων και πληροφορίας, ο οποίος δεν αφορά μόνο κάποιο συγκεκριμένο τομέα. Εκτός της σύγχρονης και ασύγχρονης επικοινωνίας που πλέον διευκολύνει τον άνθρωπο, η τεχνολογία αποτελεί και κύριο μέσο διάδοσης της επιστήμης, της πολιτικής, αλλά φυσικά και μέρους της καθημερινότητας είτε αυτό αφορά για παράδειγμα, σχόλια για κάποια ταινία -και όχι μόνο- είτε γενικά προτάσεις για οποιοδήποτε θέμα που μπορεί να φανταστεί κάποιος.

Όπως ειπώθηκε και παραπάνω, μέσω της ανάπτυξης της τεχνολογίας, η πληροφορία που παράγεται μέχρι και σήμερα αποτελείται από έναν τεράστιο όγκο δεδομένων. Από αυτόν τον όγκο προσπαθεί κανείς να συνάγει κάποια υπόθεση, κάποιο θεώρημα ή οτιδήποτε τον ενδιαφέρει. Γεννιέται τότε ένα βασικό ερώτημα:

"Πώς θα μπορέσω να αντλήσω την καλύτερη πληροφορία από το σύνολο των δεδομένων που υπάρχουν για να κατανοήσω όσο το δυνατόν καλύτερα αυτό που θέλω;"

Για την αντιμετώπιση του συγκεκριμένου ερωτήματος, θα μπορούσε, μεταξύ άλλων, να βρεθεί ένας αλγόριθμος για την αυτόματη εξαγωγή σημαντικών λέξεων ή φράσεων. Ο αλγόριθμος αυτός θα έχει ευρεία χρήση εκτός του προαναφερθέντος παραδείγματος. Γενικότερα, μεγάλο ενδιαφέρον μπορεί να έχει η εφαρμογή της αυτόματης εξαγωγής σημαντικών λέξεων ή φράσεων σε διάφορες εργασίες. Μερικές από αυτές είναι οι εξής:

- **Καλύτερη αναζήτηση της πληροφορίας:** *Τα αποτελέσματα που μπορούν να προκύψουν στους χρήστες των μηχανών αναζήτησης είναι συγκριτικά καλύτερα. Με τη χρήση της εξαγωγής λέξεων προβλήματα, όπως η κακή ορθογραφία, δεν εμποδίζουν την εμφάνιση ορθών αποτελεσμάτων.*

- **Δημιουργία περίληψης κάποιου κειμένου:** *Μέσω της αυτόματης εξαγωγής των φράσεων, αποσπώνται οι σημαντικές πληροφορίες ενός κειμένου οπότε είναι πιο εύκολη και γρήγορη η δημιουργία μιας περίληψης.*

- **Ανάλυση συναισθημάτων/ πεποιθήσεων:** *Η εξαγωγή λέξεων/φράσεων μπορεί να βοηθήσει στη σύγκλιση του συναισθήματος, των πεποιθήσεων κ.ο.κ. της κοινής γνώμης/ δείγματος για κάποιο ζήτημα. Σχετικά παραδείγματα αυτού μπορούν να θεωρηθούν τα σχόλια σε ένα άρθρο, ένα βίντεο, καθώς επίσης και οι κριτικές.*

- **Οργάνωση και ταξινόμηση της πληροφορίας:** *Μια επέκταση της ανάλυσης συναισθημάτων/πεποιθήσεων είναι η οργάνωση και η ταξινόμηση της πληροφορίας σύμφωνα με την ανάλυση που θα έχει γίνει. Δηλαδή, από τις πληροφορίες που θα έχουν προκύψει, θα κατατεθούν τα σχόλια, οι κριτικές ή γενικά τα κείμενα σε συγκεκριμένες κατηγορίες, παραδείγματος χάριν θετικά και αρνητικά ή σε αντιστοιχία με τη θεματολογία τους.*

Οι εφαρμογές της αυτόματης εξαγωγής σημαντικών λέξεων/φράσεων δεν εξαντλούνται στα παραπάνω. Αυτά, ενδεικτικά καθιστούν αντιληπτό ότι είναι σημαντική. Αν και η διαδικασία εξαγωγής σημαντικών λέξεων/φράσεων εξελίσσεται συνεχώς απαιτεί προσοχή, καθώς χρειάζεται να αντιμετωπιστούν πολλά προβλήματα κατά την υλοποίηση της. Κάποια από τα προβλήματα αυτά είναι ότι κάθε γλώσσα έχει λέξεις που η σημασία τους ποικίλουν ανάλογα με τα συμφραζόμενα. Αντίστοιχα, υπάρχουν προφορικές εκφράσεις και λέξεις που δεν αντιστοιχίζονται σε όλες τις γλώσσες.

1.2 Σπουδαιότητα

Έχοντας δει μερικές από τις χρήσεις της αυτόματης εξαγωγής λέξεων και φράσεων από κείμενα στην παραπάνω ενότητα, ίσως αναρωτηθεί κανείς "γιατί είναι τόσο σημαντικό να αναπτυχθεί κάτι τέτοιο;" Η απάντηση δίνεται σε ό,τι ακολουθεί, καλύπτοντας τόσο ακαδημαϊκούς όσο και απλά πρακτικούς λόγους:

- **Μικρή χρονική διάρκεια στην αναζήτηση πληροφοριών:** *Δεδομένου του όγκου των δεδομένων, οι εταιρείες, οι ερευνητές, αλλά ακόμα και οι απλοί πολίτες χρειάζονται τα αποτελέσματα στις αναζητήσεις τους να είναι άμεσα και γρήγορα. Με την ανάπτυξη της αυτόματης εξαγωγής σημαντικών λέξεων και φράσεων, μπορεί να επιτευχθεί η επιθυμητή ταχεία απόκριση.*

- **Αποτελεσματικότητα στις μηχανές αναζήτησης:** *Εκτός της γρήγορης αναζήτησης που πρέπει να γίνεται, ο άνθρωπος που αναζητά μια πληροφορία θέλει τα αποτελέσματα που λαμβάνει να είναι σχετικά με το θέμα που θέλει να βρει. Επομένως, η αυτόματη εξαγωγή λέξεων και φράσεων βοηθά στην επίτευξη της ακρίβειας και σχετικότητας των αποτελεσμάτων, καθώς οι μηχανές αναζήτησης βασίζονται στην ικανότητα να βρουν τα πιο σχετικά μέρη ενός κειμένου.*

- **Βοήθεια στην εξέλιξη της τεχνολογίας και επιστήμης:** *Χάρη στην πρόσβαση σε βάσεις δεδομένων υπάρχουν πάμπολλες εργασίες και επιστημονικά άρθρα στη διάθεση των ερευνητών. Η αυτόματη εξαγωγή των λέξεων και φράσεων βοηθά όλους τους ανθρώπους που θέλουν να ερευνήσουν, να ενημερωθούν για τις τελευταίες εξελίξεις της τεχνολογίας, ακόμα και να βοηθηθούν για ακαδημαϊκές εργασίες.*

Από τους παραπάνω ενδεικτικούς λόγους, αντιλαμβάνεται κανείς γιατί η αυτόματη εξαγωγή σημαντικών λέξεων ή φράσεων είναι ένα ερευνητικό πεδίο με μεγάλη σπουδαιότητα. Σπουδαιότητα η οποία υπόσχεται σημαντικές βελτιώσεις σε πολλούς τομείς εφαρμογής, προσφέροντας παράλληλα νέες ευκαιρίες για ερευνητική δραστηριότητα.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο συγκεκριμένο κεφάλαιο θα γίνει μία εισαγωγή αρχικά σε κλάδους της πληροφορικής όπως η Τεχνητή Νοημοσύνη, η Μηχανική Μάθηση, η Βαθιά Μάθηση και η Επεξεργασία Φυσικής Γλώσσας. Έχοντας αποκτήσει αυτές τις βασικές γνώσεις, θα ενταχθούν πιο περίπλοκες έννοιες και εργαλεία τα οποία είναι βασικά για την αυτόματη εξαγωγή λέξεων/ φράσεων, όπως τα Word Embeddings, το Word2Vec και το BERT.

2.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη (AI) αποτελεί μία έννοια που ο μέσος άνθρωπος θεωρεί ότι είναι καινούργια δεδομένης της έκτασης που έχει πάρει τα τελευταία χρόνια, ωστόσο ως όρος το AI ξεκινά από τα μέσα του εικοστού (20ου) αιώνα. Συγκεκριμένα, το 1955 ο John McCarthy έδωσε έναν πρώτο ορισμό του AI, σύμφωνα με τον οποίο το AI είναι "Η επιστήμη και η μηχανική που έχει ως στόχο να κάνει έξυπνες τις μηχανές" [1]. Ο συγκεκριμένος ορισμός δεν ήταν ο πρώτος ούτε και ο τελευταίος. Υπάρχουν πάρα πολλοί ορισμοί που δίνονται ακόμα και σήμερα για το τι είναι πραγματικά το AI, με στόχο την καλύτερη κατανόηση του αντικειμένου σύμφωνα με τους κλάδους της επιστήμης που χρησιμοποιούνται. Ενδεικτικά, μερικοί από αυτούς είναι οι εξής:

- "Το AI είναι ένα τεχνολογικό σύστημα το οποίο, αυτόνομα ή εν μέρει αυτόνομα, επεξεργάζεται δεδομένα που σχετίζονται με ανθρώπινες δραστηριότητες με τη χρήση γενετικού αλγορίθμου, νευρωνικού δικτύου, μηχανικής μάθησης ή άλλης τεχνικής, προκειμένου να παράγει περιεχόμενο ή να λαμβάνει αποφάσεις, συστάσεις ή προβλέψεις", σύμφωνα με τους Gulley και Hilliard [2].
- "Το AI συγκεντρώνει επιστήμες, θεωρίες και τεχνικές (συμπεριλαμβανομένης της μαθηματικής λογικής, της στατιστικής, των πιθανοτήτων, της υπολογιστικής νευροβιολογίας και της επιστήμης των υπολογιστών) και έχει ως στόχο να επιτύχει τη μίμηση από μια μηχανή των γνωστικών ικανοτήτων ενός ανθρώπου", σύμφωνα με το Συμβούλιο της Ευρώπης [2].
- "Η τεχνητή νοημοσύνη είναι η μελέτη του τρόπου με τον οποίο μπορούμε να κάνουμε τους υπολογιστές να κάνουν πράγματα που, προς το παρόν, οι άνθρωποι κάνουν καλύτερα", σύμφωνα με τους Rich και Knight [3].

Οι ορισμοί, όπως παρατηρείται παραπάνω, ποικίλουν μεταξύ τους, διατηρώντας παράλληλα την ίδια ερώτηση που έκανε ο πατέρας του AI το 1950. Ο Alan Turing, ο πατέρας του AI, όπως χαρακτηρίζεται πλέον, έθεσε το σημαντικό για την πορεία του κλάδου ερώτημα "Μπορούν να σκεφτούν οι μηχανές;" [4]. Για να απαντηθεί το ερώτημα αυτό, ο Turing δημιούργησε ένα τεστ που τελικά ονομάστηκε από τον ίδιο ως "Turing Test". Το τεστ συσχετιζόταν με έναν "ανακριτή" ο οποίος αναλάμβανε να κάνει μία ερώτηση. Οι απαντήσεις που θα λάμβανε θα ήταν δύο. Η μία ήταν από έναν άνθρωπο και η άλλη από μία υπολογιστική συσκευή. Η πρόκληση στο συγκεκριμένο τεστ ήταν να αντιληφθεί ο "ανακριτής" ποια απάντηση ήταν του ανθρώπου και ποια της μηχανής.

Αναζητώντας εφαρμογές του AI, καταγράφει κανείς μια συνεχή αύξηση των τεχνολογιών και των συσκευών που μπορούν να χρησιμοποιήσουν το AI. Ενδεικτικά κάποιες τεχνολογίες της τεχνητής νοημοσύνης είναι οι εξής:

- **Η Μηχανική Μάθηση,** *η οποία θα αναλυθεί στην επόμενη υπό-ενότητα.*

- **Η Επεξεργασία Φυσικής Γλώσσας,** *η οποία θα αναλυθεί σε επόμενες ενότητες.*
- **Η Ρομποτική,** *η οποία συν-επωφελούμενη από τους αισθητήρες του IoT (Internet of Things), πλέον μπορεί να αντιμετωπίσει λάθη και καταστάσεις που ο ανθρώπινος παράγοντας δε θα μπορούσε [5].*

Αντίστοιχα παραδείγματα μερικών συσκευών που χρησιμοποιούν την Τεχνητή Νοημοσύνη είναι οι εξής:

- **Σκούπες-ρομπότ**
- **Αυτοκίνητα με αυτόματο κιβώτιο**
- **Αλγόριθμοι στα μέσα κοινωνικής δικτύωσης**
- **Ψηφιακοί βοηθοί:** *παραδείγματος χάριν η Alexa και η Siri [6]*

Ενώ σύμφωνα με το Ευρωπαϊκό Κοινοβούλιο κάποιες καθημερινές και δυνητικές χρήσεις παρουσιάζονται στην εικόνα 2.1.

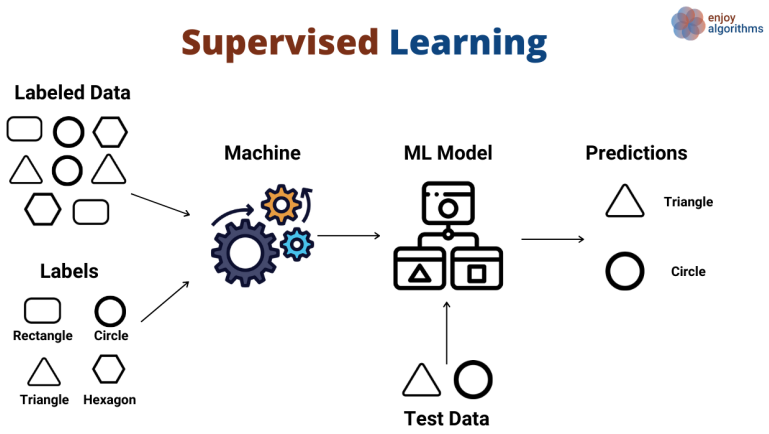
Εικόνα 2.1: Εφαρμογές του AI σύμφωνα με το Ευρωπαϊκό Κοινοβούλιο [7].

Κλείνοντας την εισαγωγή στην Τεχνητή Νοημοσύνη, διαπιστώνεται ότι τα πλεονεκτήματα της αξιοποίησης του AI είναι πολλά. Για παράδειγμα, μέσω αυτής μειώνεται η πιθανότητα ανθρώπινου λάθους με αποτέλεσμα να μπορούν να αναλαμβάνονται ριψοκίνδυνες εργασίες. Ωστόσο, υπάρχουν και μειονεκτήματα, όπως η μειωμένη δημιουργικότητα και ο αυτοματισμός που μπορεί να προκαλέσει προβλήματα στην παραγωγή, στην περίπτωση που κάτι παράγεται λάθος. Τέλος, σημαντικό είναι να αναφερθεί ότι υπάρχουν και ηθικά ζητήματα που απαιτούν προσοχή στην εξέλιξη και τη χρήση του AI.

2.2 Μηχανική Μάθηση

Η Μηχανική Μάθηση (ML) είναι ένας υπό-κλάδος του AI και συνεπώς και του κλάδου των υπολογιστών. Πρώτη φορά ως όρος εμφανίστηκε τη δεκαετία του 50 και συγκεκριμένα το 1959, από τον Arthur Samuel, ο οποίος εργαζόταν στην IBM. Προτού εισαχθεί ο όρος της Μηχανικής Μάθησης, χρησιμοποιούταν ο όρος "αυτό-εκπαιδευόμενοι υπολογιστές" [8]. Το ML επικεντρώνεται στους τρόπους με τους οποίους μαθαίνει ο άνθρωπος. Οι τρόποι αυτοί επιχειρείται να αναπαρασταθούν χρησιμοποιώντας δεδομένα και αλγόριθμους. Στόχος του ML είναι, τελικά, να βοηθήσει και να βελτιώσει την ακρίβεια του AI. Τρία μοντέλα που αξιοποιεί το ΜΛ για να επιτευχθεί ο στόχος του είναι τα εξής:

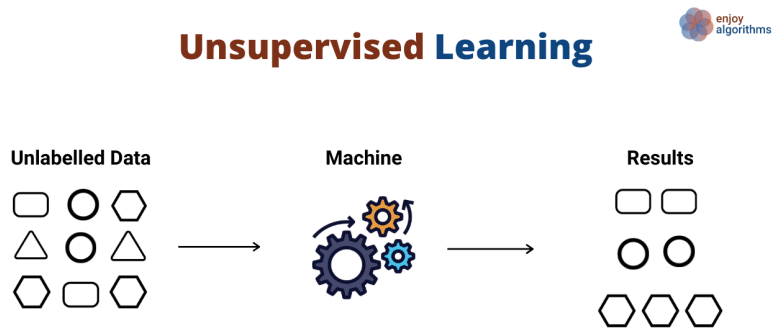
- **Supervised learning:** *Χρησιμοποιεί ένα σύνολο επισημασμένων datasets για να εκπαιδεύσει δεδομένα ή να προβλέψει αποτελέσματα με μεγάλη ακρίβεια. Κατά τη διάρκεια που τα δεδομένα εισόδου εισάγονται στο μοντέλο, το μοντέλο προσαρμόζει τα βάρη του έως ότου να προσαρμοστεί (fit) καταλλήλως, όπως αναπαριστάται στην εικόνα 2.2. Αυτό συμβαίνει στα πλαίσια της διαδικασίας διασταυρούμενης επικύρωσης για να διασφαλιστεί ότι το μοντέλο αποφεύγει την υπέρ-προσαρμογή (over-fitting) ή την υπό-προσαρμογή (under-fitting). Ένα από τα παραδείγματα εφαρμογής αυτής, η επιβλεπόμενη μάθηση βοηθά να λυθούν προβλήματα όπως η ανάπτυξη φακέλου που ξεχωρίζει τα ανεπιθύμητα μηνύματα από τα υπόλοιπα εισερχόμενα [9].*



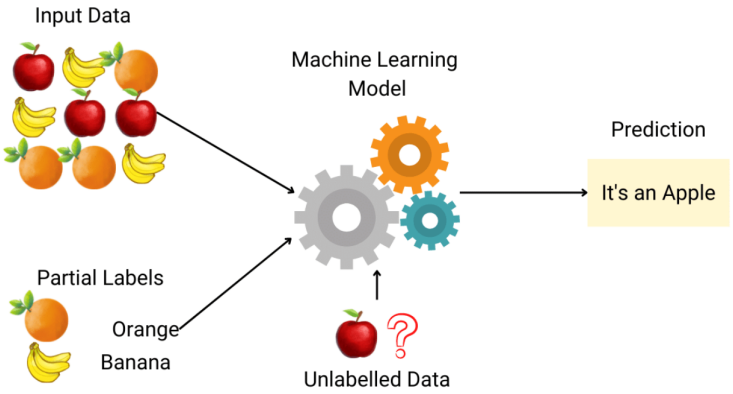
Εικόνα 2.2: Σχηματική επεξήγηση της επιβλεπόμενης μάθησης [10].

- **Unsupervised learning:** *Χρησιμοποιεί αλγορίθμους ML για να αναλύσει και να ομαδοποιήσει μη-επισημασμένα datasets, όπως αναπαρίσταται στην εικόνα 2.3. Αυτοί οι αλγόριθμοι ανακαλύπτουν κρυμμένα μοτίβα ή ομαδοποιήσεις δεδομένων χωρίς να χρειάζεται καμία ανθρώπινη παρέμβαση. Με τη μέθοδο αυτή, βρίσκονται ομοιότητες και διαφορές στις πληροφορίες, οι οποίες χρησιμοποιούνται σε κλάδους όπως η ανάλυση δεδομένων, η αναγνώριση εικόνων και μοτίβων.*

- **Semi-supervised learning:** *Χρησιμοποιείται ως μία μέση λύση, καθώς αποτελεί τη χρυσή τομή της επιβλεπόμενης και μη-επιβλεπόμενης μάθησης, όπως αναπαριστάται στην εικόνα 2.4. Καθώς εκπαιδεύεται το μοντέλο, χρησιμοποιείται ένα μικρότερο σε μέγεθος επισημασμένο dataset για να κάνει την ταξινόμηση. Ενώ αντίστοιχα, χρησιμοποιείται ένα μεγαλύτερο μη-επισημασμένο dataset για να γίνει η εξαγωγή χαρακτηριστικών. Με τη χρήση της ημι-επιβλεπόμενης μάθησης μπορεί να λυθεί το πρόβλημα της ύπαρξης λίγων δεδομένων για την επιβλεπόμενη μάθηση.*

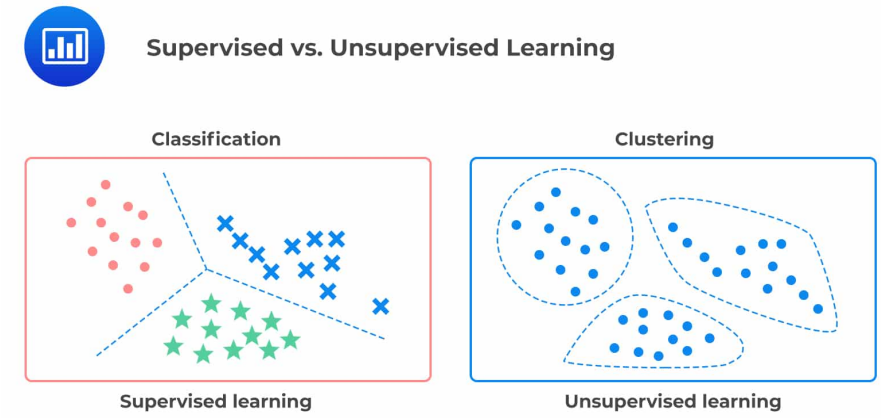


Εικόνα 2.3: Σχηματική επεξήγηση της μη-επιβλεπόμενης μάθησης [11].



Εικόνα 2.4: Σχηματική επεξήγηση της ημι-επιβλεπόμενης μάθησης [12].

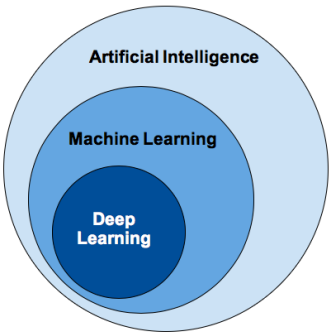
Για να φανεί καλύτερα η διαφορά των supervised και unsupervised learning, παρακάτω παρατίθεται η εικόνα 2.5 που εξηγεί τη διαφορά αυτή. Στην supervised παρατηρεί κανείς ότι τα δεδομένα, ανεξάρτητα του είδους τους, είναι όλα συγκεντρωμένα μαζί και μετά την εφαρμογή του μοντέλου τα δεδομένα έχουν χωριστεί ανάλογα με το είδος τους. Αντίθετα, στην unsupervised τα δεδομένα αποτελούν μία οντότητα, η οποία θα ομαδοποιηθεί σύμφωνα με τα κοινά τους στοιχεία.



Εικόνα 2.5: διαφορά της supervised και unsupervised learning [13].

2.3 Βαθιά Μάθηση

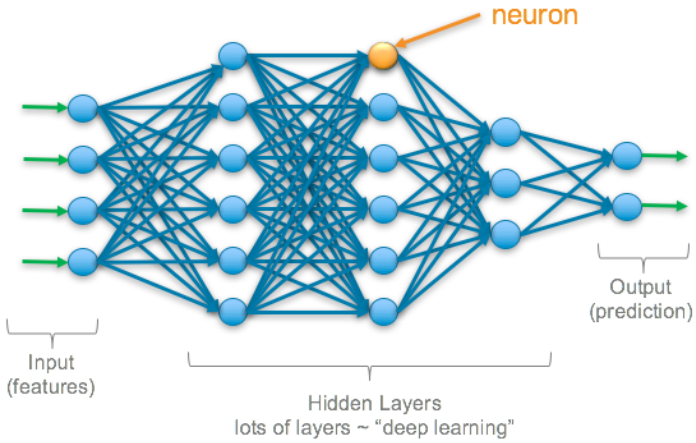
Η Βαθιά Μάθηση (DL) αποτελεί έναν υπό-κλάδο του ML και συνεπώς και του AI, όπως αναπαρίσταται στην εικόνα 2.6. Σκοπός του μοντέλου είναι να εκτελέσει μία προσομοίωση, κάτι ανάλογο με αυτό που στους ανθρώπους, εγκεφαλικά, είναι "φυσική λειτουργία". Συγκεκριμένα, η προσομοίωση αφορά τη λήψη αποφάσεων όπως αυτές λαμβάνονται από έναν ανθρώπινο εγκέφαλο. Για αυτόν ακριβώς τον λόγο, και έχοντας ήδη αναφέρει τον στόχο του AI, γίνεται αντιληπτό γιατί αρκετές εφαρμογές του AI στηρίζονται στο DL.



Εικόνα 2.6: Σχηματική επεξήγηση σχέσης AL, ML, DL

Εφόσον το DL είναι ήδη υπό-κλάδος του ML, θα αναρωτιόταν κανείς "ποια η διαφορά του ML και του DL". Η απάντηση βρίσκεται στη δομή της εκάστοτε αρχιτεκτονικής του νευρωνικού δικτύου [14]. Δηλαδή, κάποιων κυκλωμάτων νευρώνων οι οποίοι είναι συνδεδεμένοι μεταξύ τους και επιθυμούν να επιλύσουν κάποιο υπολογιστικό πρόβλημα. Τα μοντέλα του ML που δεν εντάσσονται ταυτόχρονα και στην κατηγορία του DL χρησιμοποιούν νευρωνικά δίκτυα τα οποία έχουν το πολύ δύο υπολογιστικά επίπεδα. Τα μοντέλα που εντάσσονται ταυτόχρονα και στο DL χρησιμοποιούν πάνω από τρία επίπεδα και, συνήθως, περίπου εκατοντάδες ή και χιλιάδες επίπεδα.

Τα μοντέλα που χρησιμοποιεί το DL για να λειτουργήσει δε θα αναφερθούν ονομαστικά, καθώς είναι αρκετά περίπλοκα. Ωστόσο, θα αναλυθεί η εικόνα 2.7, κατά την οποία φαίνεται ο τρόπος με τον οποίο δουλεύει το DL.



Εικόνα 2.7: Σχηματική αναπαράσταση του τρόπου λειτουργίας του DL [15].

Όπως φαίνεται και στην εικόνα 2.7, όπως και στο ML, αρχικά θα δοθούν τα δεδομένα προς ανάλυση (input). Αντίστοιχα, στο τέλος θα προκύψει και πάλι μία πρόβλεψη ή μία απόφαση (output). Παρ' όλα αυτά το σημαντικό στοιχείο αφορά τον τρόπο εξαγωγής των συγκεκριμένων αποτελεσμάτων. Όπως φαίνεται, υπάρχουν κάποια κρυφά επίπεδα τα οποία βοηθούν στο να παραχθούν οι επιθυμητές προβλέψεις ή αποφάσεις. Τα κρυφά επίπεδα ταυτίζονται με τους νευρώνες που αναφέρθηκαν παραπάνω και βοηθούν στον υπολογισμό του εκάστοτε προβλήματος.

2.4 Επεξεργασία Φυσικής Γλώσσας

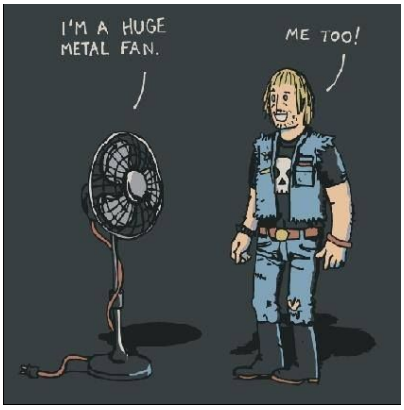
Η Επεξεργασία Φυσικής Γλώσσας (NLP) είναι ένας τομέας της πληροφορικής. Οι τεχνολογικοί τομείς, ωστόσο, που βρίσκονται πίσω από αυτήν είναι τρεις. Συγκεκριμένα η NLP παίρνει στοιχεία της Τεχνητής Νοημοσύνης (AI), της Μηχανικής Μάθησης (ML) και της Βαθιάς Μάθησης (DL). Εκτός των διάφορων τομέων της πληροφορικής σημαντικό στοιχείο αποτελεί και η κατανόηση της φυσικής γλώσσας. Μέσω του συνόλου αυτού, οι υπολογιστές επιχειρείται να κατανοήσουν και να ερμηνεύσουν ανθρώπινες δραστηριότητες. Σε ό,τι αφορά την παρούσα εργασία, συγκεκριμένα την προφορική και γραπτή επικοινωνία των ανθρώπων, παραδείγματος χάριν τις σκέψεις, ιδέες, τα γράμματα, έναν διάλογο ή μία ομιλία. Οι εφαρμογές της NLP υπάρχουν άμεσα στη ζωή των ανθρώπων χωρίς να το γνωρίζουν αρκετοί από αυτούς. Μερικά από τα πολύ διαδεδομένα παραδείγματα είναι τα εξής [16]:

- **Chatbots:** *η αυτόματη εξυπηρέτηση πελατών στις ιστοσελίδες πολλών εταιρειών, διευκολύνοντας πολλούς πελάτες, εξοικονομώντας χρόνο από τους εργαζόμενους και εξοικονομώντας χρήματα από την εκάστοτε εταιρεία.*
- **Αυτόματη συμπλήρωση κειμένου:** *παράδειγμα αποτελεί η αυτόματη συμπλήρωση που εμφανίζει η πλατφόρμα του Gmail της Google, όπου η εταιρεία προτείνει, σύμφωνα με το ύφος του κειμένου και το λεξιλόγιο που χρησιμοποιεί ο χρήστης, την ολοκλήρωση της πρότασης που γράφει. Αντίστοιχα γίνεται και με τις αυτόματες απαντήσεις σε κατηγορίες email όπως τα ευχαριστήρια.*
- **Έλεγχος ορθογραφίας:** *αποτρέπει τον χρήστη από την ανορθόγραφη αποστολή ενός κειμένου. Γεγονός που είναι άκρως σημαντικό σε ανώτερες θέσεις και γενικότερα θέσεις υψηλότερης εκπαίδευσης.*
- **Ανίχνευση spam email:** *πολύς κόσμος έχει λάβει μηνύματα που προσπάθησαν να τον εξαπατήσουν με στόχο κυρίως τους τραπεζικούς τους λογαριασμούς. Πλέον αρκετές πλατφόρμες διαχωρίζουν τέτοιου είδους μηνύματα, με σκοπό να προστατεύσουν τον κόσμο.*

Φυσικά, υπάρχουν και άλλες και ενδιαφέρουσες εφαρμογές της NLP, οι οποίες στοχεύουν στην αναζήτηση λέξεων - κλειδιών, την αυτόματη περίληψη κ.ο.κ., στοιχεία που θα αναλυθούν και παρακάτω στην παρούσα πτυχιακή εργασία.

Κλείνοντας αυτήν τη μικρή εισαγωγή στη NLP, δε γίνεται να αποφευχθεί η αναφορά στα προβλήματα που μπορεί να αντιμετωπίσει η NLP συλλέγοντας δεδομένα από κείμενα. Μερικά από αυτά τα προβλήματα είναι τα εξής [17]:

- Σύγχυση της προφορικής με τη γραπτή γλώσσα.
- Αλλαγές στη σύνταξη των προτάσεων.
- Σύγχυση του νοήματος μίας λέξεις εξαιτίας της πολυσημίας αυτής, με παράδειγμα να αναπαριστάται στην εικόνα 2.8.
- Πολυπλοκότητα και εξέλιξη της κάθε μίας γλώσσας.



Εικόνα 2.8: Σύγχυση νοήματος λέξης εξαιτίας της πολυσημίας αυτής [17].

2.5 Word Embeddings

2.5.1 Εισαγωγή

Μία από τις εφαρμογές/κλάδους της NLP που δεν αναλύθηκε στην εισαγωγή της, η οποία αφορά άμεσα την αυτόματη εξαγωγή λέξεων ή φράσεων, είναι η ενσωμάτωση λέξεων ή κοινώς τα word embeddings. Τι είναι όμως τα word embeddings; Όπως ειπώθηκε και στην εισαγωγή της NLP, δεδομένου ότι οι υπολογιστές μέσω των αλγορίθμων της DL δεν μπορούν να επεξεργαστούν την ανθρώπινη γλώσσα, είναι αναγκαίο να βρεθεί ένας τρόπος ούτως ώστε να μετατραπεί η ανθρώπινη γλώσσα, προφορική και γραπτή, σε μία κατανοητή για τους υπολογιστές. Για αυτή τη δουλειά είναι υπεύθυνα τα word embeddings.

Συγκεκριμένα, τα word embeddings χρησιμοποιούν διανύσματα σε πολυδιάστατους χώρους, δηλαδή αριθμούς για λέξεις εισόδου, προκειμένου να αναπαραστήσουν τις λέξεις σε "κάτι" κατανοητό για τους υπολογιστές. Μία παραδοσιακή μέθοδος αναπαράστασης των λέξεων σε διανύσματα είναι το one-hot encoding (κωδικοποίηση one-hot) [18]. Για παράδειγμα, έστω οι λέξεις "αστυνομικό", "κωμωδία", "επιστημονικό". Και οι τρεις μαζί ορίζουν το λεξιλόγιο του παραδείγματος αυτού. Το διάνυσμα της κάθε λέξης θα οριστεί σύμφωνα με το σύνολο των λέξεων και τη σειρά εμφάνισης της καθεμίας. Δηλαδή, στο παράδειγμα αυτό:

- "αστυνομικό" = [1 , 0 , 0]
- "κωμωδία" = [0 , 1 , 0]
- "επιστημονικό" = [0 , 0 , 1]

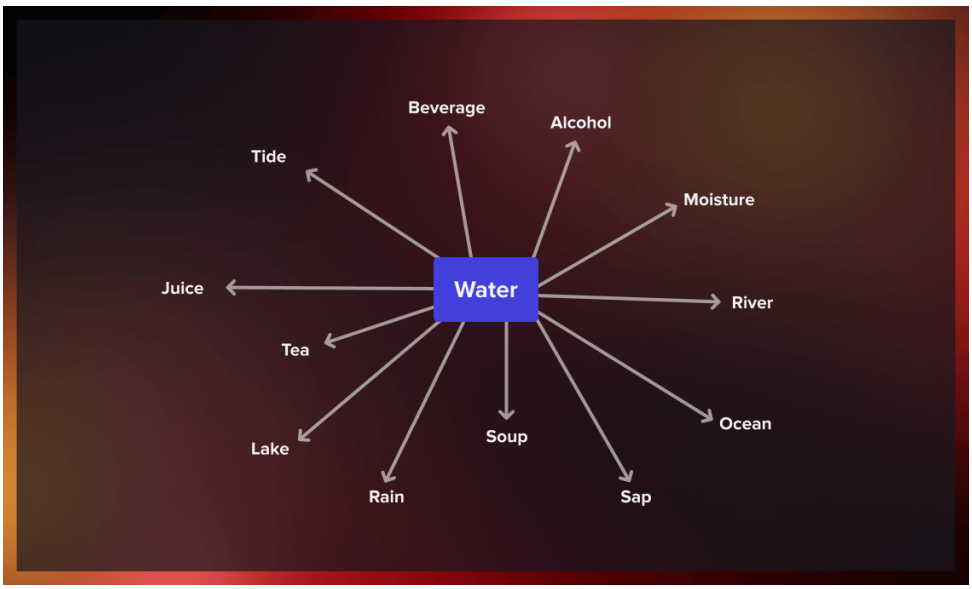
Ένα βασικό πρόβλημα του one-hot encoding, ωστόσο, είναι ότι, όπως και στο παράδειγμα αυτό, δε θα μπορούσε να δοθεί κάποια πληροφορία για τη σημασιολογική σχέση του λεξιλογίου. Για να εντοπιστεί η ομοιότητα ή άλλη σχέση μεταξύ λέξεων χρησιμοποιείται η κατεύθυνση και η απόσταση των διανυσμάτων.

Για να εκπαιδευτεί, λοιπόν, σωστά το μοντέλο ούτως ώστε να εξαχθούν αποτελέσματα, χρειάζεται ένα (μεγάλο) κείμενο, το οποίο μας ενδιαφέρει. Το κείμενο αυτό, αρχικά, θα μετατραπεί από προτάσεις σε λέξεις (τακτική tokenizing) και ταυτόχρονα θα αφαιρεθούν τα κενά (spaces) και τα σημεία στίξης. Αυτό θα γίνει, προκειμένου να μείνουν μονάχα οι λέξεις, οι οποίες είναι το μόνο στοιχείο που μας ενδιαφέρει από το κείμενο για την ανάλυσή του. Έπειτα θα οριστεί το κάθε παράθυρο (window), το οποίο θα μετακινείται κατά μία λέξη δεξιά, ορίζοντας κάθε φορά μία "λέξη-στόχο" (target word) και το νέο window, λαμβάνοντας τις τριγύρω λέξεις ως συμφραζόμενα. Κάθε λέξη λαμβάνει ένα και μοναδικό διάνυσμα, σύμφωνα με το window που θα οριστεί, το οποίο προδίδει το πού βρίσκεται η κάθε λέξη στον πολυδιάστατο χώρο. Τελικώς, τα μοντέλα του word embedding στοχεύουν στο να μπορούν να προβλέψουν μία λέξη σύμφωνα με τα συμφραζόμενα, καθώς και το αντίστροφο.

Μερικές από τις πιο διαδεδομένες μεθόδους για την εκπαίδευση του word embedding είναι η Word2Vec (Word to Vector) που θα αναλυθεί στην επόμενη υπό-ενότητα, το GloVe (Global Vectors for word representation), το fastText και το TF-IDF (Term Frequency - Inverse Document Frequency).

Κάποιες από τις χρήσεις του word embedding είναι οι παρακάτω, οι οποίες σχετίζονται και με χρήσεις της NLP:

1. Εύρεση συνώνυμων και μη ή ομαδοποίηση λέξεων, παρατηρείται και στην εικόνα 2.9.
2. Ταξινόμηση κειμένου (Text classification): τα word embeddings βοηθούν στην ταξινόμηση των κειμένων έχοντας ως στόχο την ανάλυση, την αναγνώριση spam και γενικώς τη θεματολογία αυτών [18].
3. "Έξυπνη εξυπηρέτηση πελατών"/ "Chatbot": ο χρήστης μπορεί να επικοινωνήσει με την εταιρεία άμεσα και εύκολα μέσω αυτοματοποιημένων απαντήσεων [19].
4. BERT (Bidirectional Encoder Representations from Transformers): το word embedding χρησιμοποιείται ως βάση για την εκπαίδευση του συγκεκριμένου μοντέλου, όπως θα αναλυθεί σε επόμενη ενότητα.

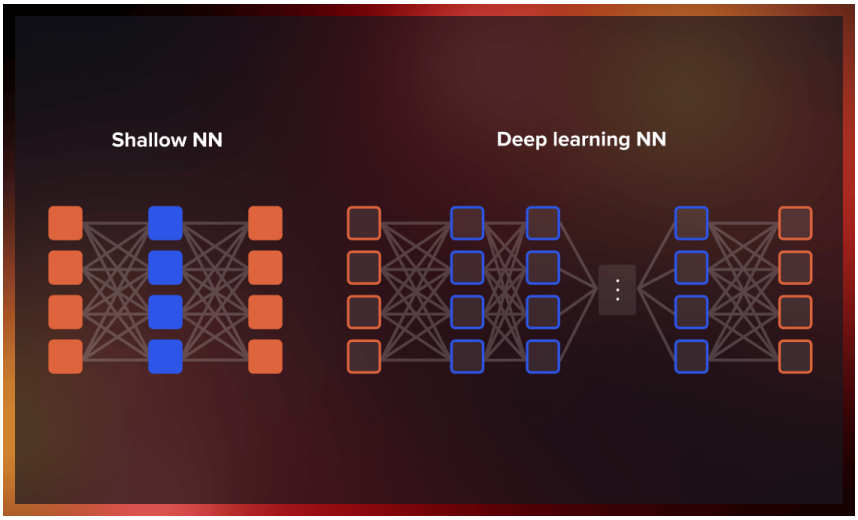


Εικόνα 2.9: Παράδειγμα αποτελέσματος λέξεων που συνδέονται με τη λέξη "water" [20].

2.5.2 Word2Vec - Ορισμός της έννοιας

Στη συγκεκριμένη υπό-ενότητα θα αναλυθεί η τεχνική του word2vec, η οποία χρησιμοποιείται ως συντομογραφία για το word to vector. Όπως αναφέρθηκε και παραπάνω, η τεχνική αυτή αφορά μία από τις πιο διαδεδομένες μεθόδους αναπαράστασης των word embeddings. Η "δημιουργία" της ήρθε ως αποτέλεσμα της ραγδαίας εξέλιξης που έλαβε τις τελευταίες δεκαετίες το AI διευκολύνοντας με αυτόν τον τρόπο μοντέλα των NLP και ML. Συγκεκριμένα, ο Tomáš Mikolov (Τσέχος επιστήμονας ο οποίος ασχολείται με τον κλάδο του ML) έφτασε πολύ κοντά στη δημιουργία της τεχνικής αυτής το 2010, ενώ η επίσημη κυκλοφορία ήρθε το 2013 από τον ίδιο σε συνεργασία με ομάδα της Google.

Ένας από τους βασικούς λόγους που το word2vec υπερτερεί συγκριτικά με τα μοντέλα του DL είναι η χρήση των νευρωνικών δικτύων. Το DL χρησιμοποιεί κατά κύριο λόγο βαθιά νευρωνικά δίκτυα (Deep neural networds: DNNs), όπως ειπώθηκε και σε εισαγωγική ενότητα, τα οποία εκτός από τα πορίσματα εισόδου/εξόδου έχουν και πολλά (> 2) κρυφά στρώματα. Στο word2vec τα νευρωνικά δίκτυα είναι ρηχά (Shallow neural networks: Shallow NNs), δηλαδή το κρυφό στρώμα είναι ένα ή το πολύ δύο (< 2) μεταξύ των πορισμάτων εισόδου/εξόδου. Η διαφορά των δύο εικονίζεται στην εικόνα 2.10. Μπορεί να μη φαίνεται τεράστια, ωστόσο με το μοντέλο του word2vec (Shallow NNs) τα σημασιολογικά αποτελέσματα που πρέπει να εξαχθούν προκύπτουν ταχύτερα από τα DNNs, γεγονός που την καθιστά σημαντική.



Εικόνα 2.10: Διάγραμμα διαφορών κρυφών νευρωνικών δικτύων σε Shallow NNs και DNNs [20].

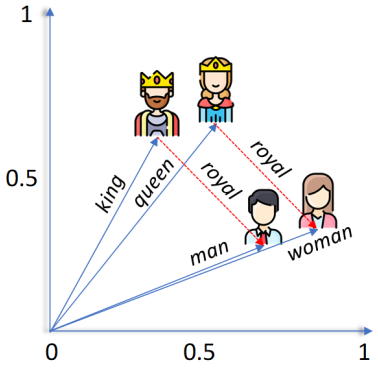
Τι είναι όμως πρακτικά το word2vec; Όπως λέει και η συντομογραφία, είναι μία τεχνική που μετατρέπει τις λέξεις σε διανύσματα και ειδικότερα σε διανύσματα σε πολυδιάστατο χώρο. Συγκεκριμένα παίρνει ένα σύνολο λέξεων ή/και κειμένων ως όρισμα και τα μετατρέπει σε μοναδικά κατά το όρισμα διανύσματα. Στόχος της τεχνικής αυτής είναι να αποτυπωθεί η σημασιολογική έννοια των λέξεων προκειμένου να βρεθεί η σχέση τους με το κείμενο στο οποίο ανήκουν ή ακόμα και η σχέση μεταξύ τους [20]. Εξαιτίας αυτού, η τοποθέτηση των λέξεων σε πολυδιάστατο χώρο γίνεται με καθορισμένο τρόπο. Οι λέξεις με ίδιο ή παρόμοιο νόημα βρίσκονται πιο κοντά συγκριτικά με τις λέξεις που απέχουν νοηματικά.

Μία εξέλιξη της τεχνικής του "word to vector" αφορά, επίσης, την ανάλυση των ήδη αναγνωρισμένων για την τεχνική διανυσμάτων. Ένα από τα πιο διαδεδομένα παραδείγματα για να επεξήγηση του vector analysis (διανυσματική ανάλυση) είναι το επικείμενο παράδειγμα "$king - man + woman = queen$".

Έστω ότι αναζητείται το αποτέλεσμα του "king" αν αφαιρεθεί το "man" και προστεθεί το "woman". Αρχικά η τεχνική θα δημιουργήσει 4 διανύσματα, ένα διάνυσμα για κάθε μία έννοια ("king", "man", "woman", "queen"). Ενδεικτικά:

- $\overrightarrow{u_{king}}$ = "king",
- $\overrightarrow{u_{man}}$ = "man",
- $\overrightarrow{u_{woman}}$ = "woman" και
- $\overrightarrow{u_{queen}}$ = "queen"

Νοηματικά υπάρχει σύνδεση μεταξύ των λέξεων "king" και "man" και αντίστοιχα για τις λέξεις "woman" και "queen". Οπότε αφαιρώντας την ιδιότητα "man" ο αλγόριθμος κατανοεί ότι αφαιρείται το αρσενικό φύλο στην ιδιότητα "king" αλλά ταυτόχρονα παραμένει η ιδιότητα του αξιώματός του (Εικόνα 2.11).



Εικόνα 2.11: Αναπαράσταση διανυσμάτων παραδείγματος $king - man + woman = queen$ [21].

Στο επόμενο βήμα προστίθεται το "woman", δηλαδή προστίθεται το γυναικείο φύλο που έχει την ιδιότητα του "king". Έτσι, το αποτέλεσμα που αναμένεται να βγάλει η τεχνική ως το πιο κοντινό διάνυσμα είναι η λέξη "queen". Ο λόγος που προκύπτει το συγκεκριμένο αποτέλεσμα είναι ότι η τεχνική της ενσωμάτωσης λέξεων εμβαθύνει στη νοηματική ανάλυση των λέξεων εκτός του νοήματος της λέξης μέσα σε ένα κείμενο.

2.5.3 Word2Vec - Τρόποι εκπαίδευσης της τεχνικής

Προκειμένου να δουλέψει και να προκύψουν αποτελέσματα, η τεχνική word2vec πρέπει να εκπαιδευτεί με κάποιον τρόπο. Ο τρόπος αυτός δεν είναι μοναδικός. Συγκεκριμένα υπάρχουν δύο μέθοδοι μέσω των οποίων δουλεύει το word2vec. Η πρώτη είναι το CBOW (Common Bag of Words) και η δεύτερη είναι το Skip Gram. Το κοινό τους στοιχείο είναι ότι και οι δύο αξιοποιούν νευρωνικά δίκτυα. Στην παρακάτω υπό-ενότητα 2.5.3.1 θα αναλυθούν οι μέθοδοι αυτές και θα αναδειχθούν οι βασικές μεταξύ τους διαφορές.

2.5.4 Word2Vec - CBOW (Common Bags of Words)

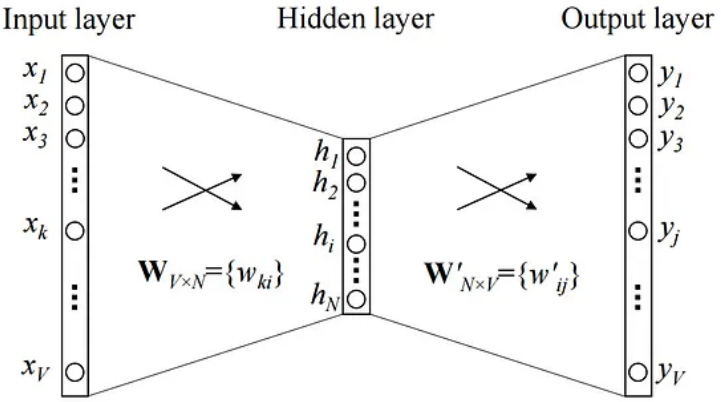
Η πρώτη μέθοδος είναι το CBOW, το οποίο αποτελεί αρκτικόλεξο του Common Bag of Words. Η συγκεκριμένη τεχνική έχει ως στόχο να προβλέψει τη "λέξη-στόχο" κοινώς την target word σύμφωνα με το κείμενο που υπάρχει στην αντίστοιχη πρόταση ως συμφραζόμενα.

Εξαιτίας του ότι η CBOW αποτελεί μέθοδο του word2vec και κατ' επέκταση DL, όπως αναλύθηκε σε προηγούμενες ενότητες, υπάρχει άμεση σύνδεση του μοντέλου με τα νευρωνικά δίκτυα. Στην περίπτωση αυτή, ως input layer θα θεωρηθούν τα συμφραζόμενα και ως output layer η target word. Στο hidden layer υποβόσκει το πιο περίπλοκο στάδιο, καθώς στο αυτό το σημείο λειτουργεί η τεχνική των word embeddings.

Παρακάτω θα αναλυθούν με περισσότερες λεπτομέρειες το κάθε ένα layer, ξεκινώντας από το input layer. Τα πορίσματα που ορίζονται εξαρτώνται άμεσα από τον αριθμό των συνολικών λέξεων που υπάρχουν στο αντίστοιχο "λεξιλόγιο". Παραδείγματος χάριν, αν το λεξιλόγιο αποτελείται από 100 λέξεις, τότε ως input layer θα υπήρχαν 100 διαφορετικά one-hot encoded διανύσματα. Ενώ θεωρείται ότι κάθε λέξη αποτελείται από ένα μοναδικό διάνυσμα. Το μέγεθος του κάθε one-hot encoded διανύσματος είναι V (όπου V = vocabulary το σύνολο των λέξεων), το οποίο είναι το ίδιο και με το μέγεθος των διανυσμάτων στο output layer. Η διαφορά μεταξύ τους είναι ότι στο output layer οι τιμές είναι σε κλίμακα από το μηδέν (0) έως και το ένα (1). Το αντίστοιχο μέγεθος που περιέχει τους νευρώνες στο hidden layer είναι N (όπου N = neurons ο αριθμός των νευρώνων).

Από την Εικόνα 2.12 προκύπτει ότι δημιουργούνται δύο πίνακες:

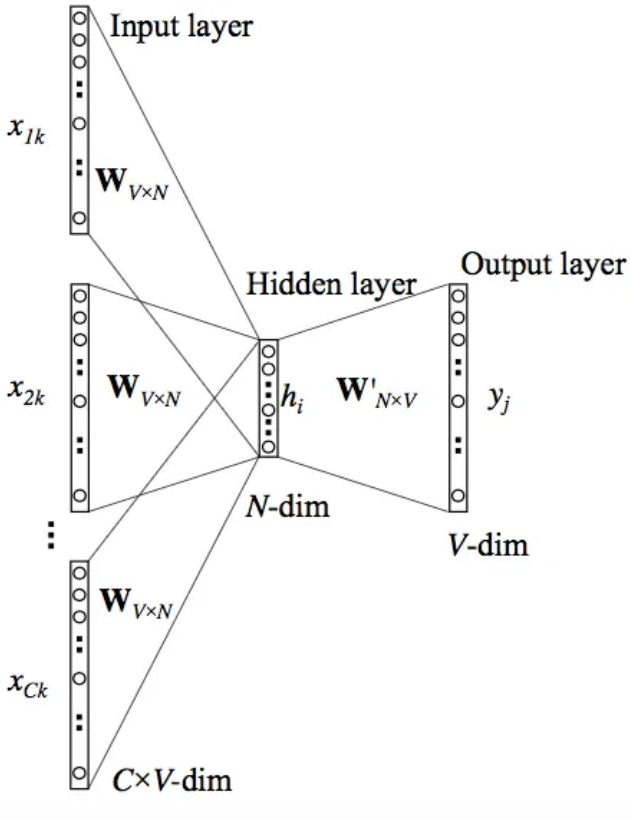
- Ο $W_{V\times N}$ πίνακας είναι υπεύθυνος για τα στοιχεία που αφορούν το input και hidden layer με διάσταση $V \times N$.
- Ο $W'_{N\times V}$ πίνακας είναι υπεύθυνος για τα στοιχεία που αφορούν το hiden και output layer με διάσταση $N \times V$.



Εικόνα 2.12: Ένα απλό μοντέλο CBOW με μία μόνο περιεχόμενη λέξη [22].

Το παραπάνω μοντέλο υπενθυμίζεται ότι χρησιμοποιείται για να προβλέψει την target word μόνο με μία λέξη ως λεξιλόγιο. Η αντίστοιχη διαδικασία μπορεί να ακολουθηθεί και με πολλαπλές λέξεις (κείμενα, προτάσεις κ.ο.κ.). Όταν συμβαίνει αυτό, στο hidden layer αλλάζουν μερικά στοιχεία συγκριτικά με το one word κείμενο. Πριν γίνει ο πίνακας $W_{V\times N}$, το μοντέλο παίρνει τον μέσο όρο των διανυσμάτων των λέξεων του input layer. Ως νέος πίνακας παράγεται το γινόμενο του πίνακα βάρους του input → hidden με τον μέσο όρο που υπολογίστηκε στο προηγούμενο βήμα, Εικόνα 2.13 [22].

Γενικώς, το μοντέλο του CBOW αποτελεί ένα δυνατό εργαλείο για την τεχνική του word2vec, καθώς τα αποτελέσματα του είναι μεγάλης ακρίβειας. Η χρήση του είναι ευρεία τόσο σε text classification όσο και σε άλλους κλάδους του NLP.



Εικόνα 2.13: Λειτουργία του CBOW μοντέλου [22]

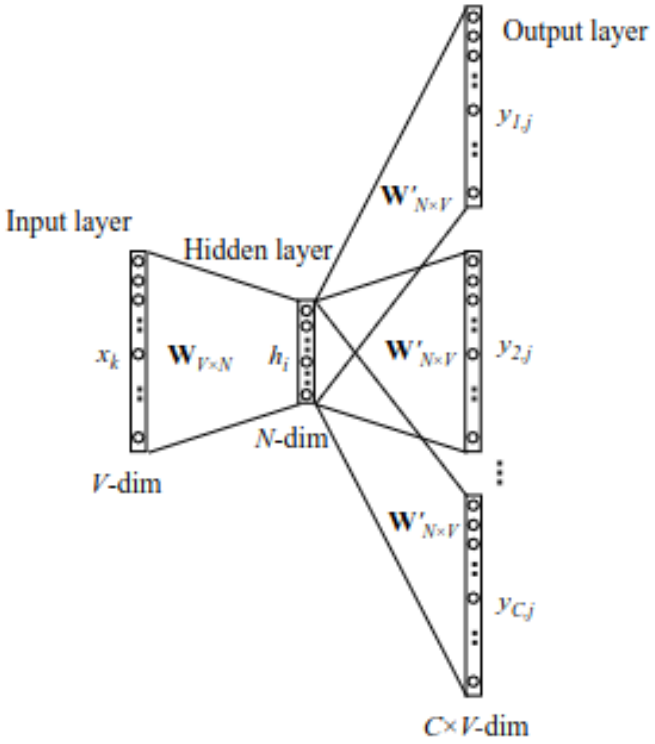
2.5.5 Word2Vec - Skip-Gram

Η δεύτερη μέθοδος εκπαίδευσης του word2vec είναι το Skip-Gram. Το Skip-Gram αποτελεί μία unsupervised τεχνική, η οποία οπτικά μοιάζει με το αντίστροφο μοντέλο του CBOW όταν αυτό δέχεται πολλές λέξεις (context) ως όρισμα. Η τεχνική αυτή έχει, δηλαδή, ως στόχο να προβλέψει τις λέξεις που περιβάλλουν (context) την target word, το οποίο ορίζεται στο input, γεγονός που το καθιστά υπολογιστικά πιο απαιτητικό από το CBOW.

Στο input layer, όπως ειπώθηκε και προηγουμένως, εισάγεται μία λέξη, η target word ($w(t)$). Οπότε στο στάδιο της εισαγωγής το μοντέλο ξεκινάει όπως το CBOW, στην περίπτωση της μίας λέξης ως input. Κατ' επέκταση ίδιο μένει και το διάνυσμα h στο hidden layer. Η κύρια διαφορά με τον τρόπο που λειτουργεί το Skip-Gram εισάγεται στο σημείο ενδιάμεσα στο hidden και output layer. Στο στάδιο αυτό υπολογίζεται το εσωτερικό γινόμενο μεταξύ του πίνακα βάρους ($W_{V\times N}$) και του διανύσματος εισαγωγής (input). Το αποτέλεσμα θα μεταφερθεί στο output layer, στο οποίο το εσωτερικό γινόμενο θα χρησιμοποιηθεί για την πρόβλεψη των συμφραζομένων (context words) με βάση τα διανύσματα που υπολογίστηκαν στο hidden layer [22].

Τέλος, θα εφαρμοστεί και η συνάρτηση softmax για να μετατρέψει τα αποτελέσματα του output layer σε πιθανότητες, με σκοπό να μεγιστοποιηθούν οι πιθανότητες για τις σωστές context λέξεις. Δεδομένου ότι το αποτέλεσμα θα είναι ένα σύνολο από λέξεις και όχι μία μοναδική, η συνάρτηση softmax υπολογίζει τις πιθανότητες για κάθε context λέξη. Το σύνολο των λέξεων που θα προβλέψει το μοντέλο είναι C , όπου το C είναι ο αριθμός του window size που θα οριστεί. Παραδείγματος χάριν, αν το window size είναι 1, το μοντέλο θα προσπαθήσει να προβλέψει τις λέξεις ακριβώς πριν και μετά η target word (δηλαδή τις λέξεις στις θέσεις $(t-1)$ και $(t+1)$) [22].

Στην Εικόνα 2.14 φαίνεται και πρακτικά πώς λειτουργεί το μοντέλο.



Εικόνα 2.14: Σχηματική αναπαράσταση του μοντέλου Skip-Gram [22].

2.5.6 Word2Vec – Διαφορές CBOW και Skip-Gram

Παρακάτω θα αναφερθούν μερικές από τις βασικές διαφορές των δύο μεθόδων εκπαίδευσης της τεχνικής του word2vec.

- Μία από τις βασικές διαφορές είναι το τι προβλέπει το κάθε μοντέλο. Στο CBOW δίνονται τα παρεμφερή συμφραζόμενα και αναζητείται η target word. Αντιθέτως, στο Skip-Gram δίνεται η target word και αναζητείται το κείμενο που ταιριάζει με αυτήν.

- Ένα από τα πιο δυνατά χαρακτηριστικά του CBOW είναι η ακρίβεια του μοντέλου. Δεδομένου ότι λαμβάνει ως όρισμα ένα σύνολο λέξεων και αναζητά μία μόνο για να προβλέψει, μπορεί να εκπαιδεύσει μεγαλύτερα dataset από το μοντέλο του Skip-Gram, το οποίο θέλει μία λέξη για να προβλέψει πολλές [23].

- Το Skip-Gram είναι λειτουργεί καλύτερα όταν το dataset είναι μικρό, σύμφωνα με τον Mikolov.

- Το CBOW λειτουργεί πιο γρήγορα και μπορεί να αναπαραστήσει καλύτερα λέξεις που χρησιμοποιούνται περισσότερο στην καθομιλουμένη. Το Skip-Gram, αντίστοιχα, μπορεί να αναπαραστήσει καλά λέξεις σπάνιες στη χρήση [24].

Κλείνοντας την ενότητα με μία μικρή ανασκόπηση, τα word embeddings δημιουργούν πολυδιάστατα διανύσματα με σκοπό να αποτυπωθεί η νοηματική και συντακτική τους σύνδεση. Για να επιτευχθεί αυτό υπάρχουν πολλές μέθοδοι αναπαράστασης. Μία από αυτές είναι και η τεχνική του word2vec. Το word2vec, συγκεκριμένα, μπορεί να εφαρμοστεί με δύο διαφορετικούς τρόπους, τα μοντέλα CBOW και Skip-Gram. Αν και έχουν και τα δύο αρκετά θετικά στοιχεία είναι σημαντικό να τονιστεί ότι η μέθοδος του word2vec είναι ηλικίας τουλάχιστον μίας δεκαετίας, γεγονός που την καθιστά πλέον παλιό αλγόριθμο στο χώρο του ML. Μία εξελιγμένη και σχετικά νέα τεχνική είναι το BERT που θα αναλυθεί στην επόμενη παράγραφο.

2.6 Αρχιτεκτονικές εφαρμογής της NLP

Στην ενότητα αυτή θα αναλυθεί άλλη μία αρχιτεκτονική, η οποία είναι πιο καινούργια συγκριτικά με τα word embeddings (2003), καθώς αναπτύχθηκε από το 2016 και έπειτα. Η αρχιτεκτονική αποτελεί κλάδο του AI και συγκεκριμένα της NLP, όπως και τα word embeddings. Ειδικότερα, θα γίνει μία εισαγωγή στα Transformer και έπειτα θα αναπτυχθεί και το μοντέλο BERT.

Πριν ξεκινήσει η ανάλυση της αρχιτεκτονικής του Transformer, ωστόσο, και για να κατανοηθεί καλύτερα η έννοια αυτή θα δοθεί ένας μικρό ορισμός για τα seq2seq (sequence to sequence) μοντέλα. Αποτελούν ένα μοντέλο νευρωνικών δικτύων, τα οποία έχουν ως στόχο να μετατραπεί μία σειρά (sequence) δεδομένων σε μία άλλη. Σε αντίθεση με τα νευρωνικά δίκτυα, τα μοντέλα seq2seq δουλεύουν εύκολα σε δεδομένα όπου τα input και output έχουν διαφορετικό μήκος. Αυτό τα καθιστά ένα πολύ καλό εργαλείο για εφαρμογές όπως η περίληψη κειμένων, η αναγνώριση και μετατροπή προφορικού και γραπτού λόγου, καθώς και η δημιουργία chatbots [25].

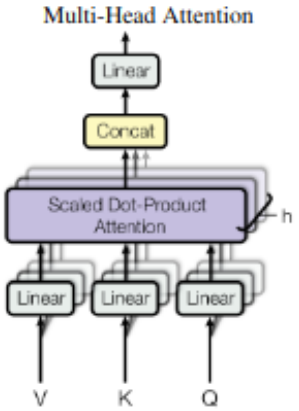
2.6.1 Transformers – Ορισμός της έννοιας

Ο μετασχηματιστής, γνωστός και ως Transformer, είναι μία αρχιτεκτονική νέα στην ηλικία και καινοτόμα που αξιοποιείται από τη NLP με σκοπό να πραγματοποιήσει μία εργασία seq2seq (sequence to sequence). Βασίζεται εξ ολοκλήρου στο self attention (αυτό-προσοχή) προκειμένου να υπολογίσει αναπαραστάσεις των δεδομένων εισαγωγής και εξαγωγής, χωρίς να χρησιμοποιούνται RNNs (Recurrent neutral networks)[1] με αλληλουχία (sequence-aligned) ή συνελικτικές τεχνικές [26]. Αυτό είναι ένα άκρως σημαντικό στοιχείο των Transformer, καθώς έτσι τα δεδομένα μπορούν να επεξεργαστούν με μεγαλύτερη ταχύτητα.

Αλλά τι είναι το self attention που στηρίζεται σε αυτό ολόκληρη η αρχιτεκτονική; Σύμφωνα με το άρθρο "Attention is all you need" που γράφτηκε από μία ομάδα ερευνητών το 2017 [27], το self attention αποτελεί έναν μηχανισμό προσοχής για να συνδέσει διαφορετικές θέσεις μιας ενιαίας ακολουθίας προκειμένου να υπολογιστεί μια αναπαράσταση της ακολουθίας. Εκτός από αυτά, χρησιμοποιούνται και κάποια είδη multi-head attention. Τα multi-head attention ουσιαστικά είναι πολλά self attention μαζεμένα σε ένα. Το βασικό τους στοιχείο είναι ότι δουλεύουν ταυτόχρονα και παράλληλα, με το κάθε ένα να συγκεντρώνεται σε ένα διαφορετικό κομμάτι της σχέσης μεταξύ των στοιχείων.

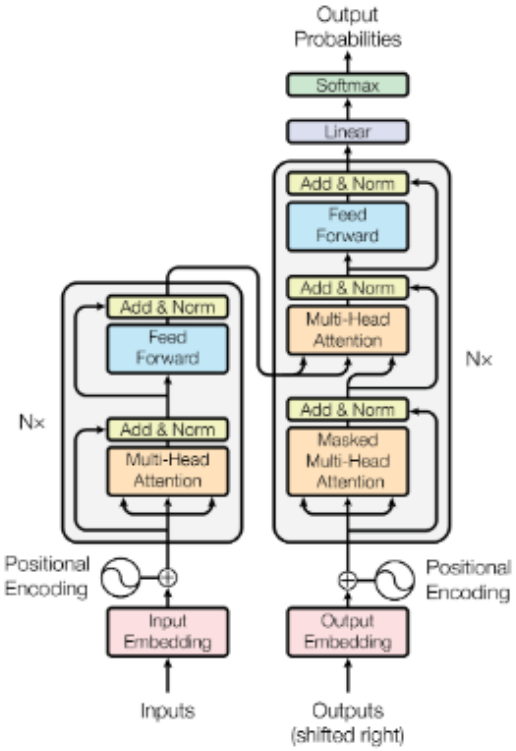
Ένα παράδειγμα της διαφοράς των δύο τύπων προσοχής είναι ότι το self συγκεντρώνεται σε κάθε μία λέξη μεμονωμένα, ενώ το multi-head θα παρακολουθήσει ολόκληρη την πρόταση και θα συγκεντρωθεί σε στοιχεία όπως η σύνταξη, γραμματική κ.ο.κ. [28]. Με τη χρήση και των δύο τύπων προσοχής γίνεται μία πιο ουσιαστική κατανόηση του κειμένου που εξετάζεται.

Η αρχιτεκτονική των δύο τύπων δίνεται σχηματικά στις Εικόνες 2.15 και 2.16.



Εικόνα 2.15: Μηχανισμός multi-head attention [27].

[1] Τα RNNs είναι DNNs, τα οποία εκπαιδεύονται με διαδοχικά δεδομένα ή δεδομένα χρονοσειρών. Στόχος τους είναι να δημιουργήσουν ένα ML μοντέλο για να πάρουν διαδοχικές προβλέψεις ή συμπεράσματα σύμφωνα με τα διαδοχικά στοιχεία εισόδου [29].



Εικόνα 2.16: Αρχιτεκτονική Transformer [27].

Πριν ολοκληρωθεί η εισαγωγή στα Transformers, δε γίνεται να μην αναφερθούν και τα τρία είδη αρχιτεκτονικής αυτών [30]:

- **Vanilla Transformer:** *σύμφωνα με αυτήν την αρχιτεκτονική το μοντέλο έχει και encoder και decoder.*
- **Decoder only Transformer:** *το μοντέλο έχει μόνο decoders.*
- **Encoder only Transformer:** *το μοντέλο έχει μόνο encoders.*

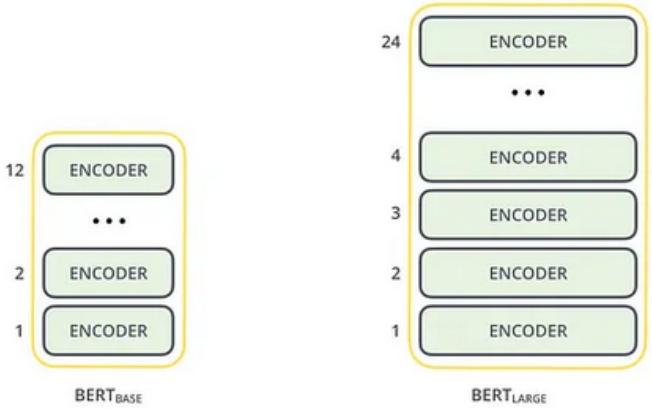
2.6.2 BERT – Ορισμός και Αρχιτεκτονική της έννοιας

Μία εφαρμογή των Transformers είναι και η τεχνική BERT. Συγκεκριμένα BERT ονομάζεται το αρκτικόλεξο του Bidirectional Encoder Representations from Transformers, δηλαδή Αναπαραστάσεις Κωδικοποιητή Διπλής Κατεύθυνσης από Μετασχηματιστές. Η ίδια αποτελεί μία σχετικά καινούργια τεχνική, καθώς βγήκε στην κυκλοφορία στα τέλη του 2018 από μία ομάδα ερευνητών της Google που ειδικεύονταν στο τομέα του AI. Αποτελεί μία καινοτόμα τεχνική για τον τομέα της NLP και κατ' επέκταση και του ML και οδήγησε στην εφαρμογή πολλών NLP λειτουργιών που θα αναφερθούν παρακάτω.

Γιατί είναι όμως καινοτόμα η τεχνική BERT; Μέχρι την κυκλοφορία του BERT, τα Transformers ήταν εκπαιδευμένα να εξετάζουν μία πρόταση είτε από αριστερά προς τα δεξιά είτε από δεξιά προς τα αριστερά. Αυτό σήμαινε ότι το μοντέλο μπορούσε να δει μόνο τις λέξεις που ακολουθούσαν ή που προηγούνταν της λέξης που εξεταζόταν, αλλά ποτέ και τις δύο κατευθύνσεις ταυτόχρονα. Το καινούργιο και καινοτόμο στοιχείο που εισήγαγε η τεχνική BERT είναι η διπλής κατεύθυνσης εκπαίδευση των μετασχηματιστών. Δηλαδή, η τεχνική πλέον παρατηρεί και τις λέξεις που ακολουθούν, αλλά και τις λέξεις που προηγούνται από τη λέξη που εξετάζεται. Με αυτόν τον τρόπο, δίνεται μια σφαιρική άποψη για τη σημασία των λέξεων και θα αποφευχθούν προβλήματα όπως η πολυσημία μίας έννοιας.

Για να επιτευχθεί αυτό, αξιοποιείται συγκεκριμένα η αρχιτεκτονική που έχουν τα Transformers του BERT. Η αρχιτεκτονική τους ανήκει στην κατηγορία των Encoder only Transformer, η οποία βοηθάει στην ανάλυση κειμένων, σύνταξης και δημιουργίας embeddings [31]. Αυτό είναι άκρως σημαντικό, καθώς εξαιτίας της αρχιτεκτονικής αυτής, το BERT μπορεί να λάβει πληροφορία για το περιεχόμενο της κάθε λέξης, που εξετάζει, σύμφωνα και με τα συμφραζόμενά της. Το γεγονός αυτό του δίνει προβάδισμά συγκριτικά με την τεχνική του word2vec, που μπορεί να παρερμηνεύσει κάποιες έννοιες ανεξαρτήτου του δοθέντος κείμενου.

Όσον αφορά τα μοντέλα αρχιτεκτονικής του BERT, γενικά ως τεχνική, χωρίζονται σε δύο κατηγορίες, Εικόνα 2.17. Η διαφορά τους προκύπτει σύμφωνα με τον αριθμό των encoders που μπορούν να δεχτούν, δηλαδή των τμημάτων των Transformers που αναλύουν τις σχέσεις μεταξύ των λέξεων. Συγκεκριμένα, το BERT Base δέχεται 12 encoders και το BERT LARGE δέχεται 24 encoders, όπου και τα δύο στοιβάζουν τα encoders το ένα πάνω από το άλλο [30]. Με την αύξηση των encoders αυτό που επιτυγχάνεται είναι η καλύτερη κατανόηση των λέξεων και συμφραζομένων είτε είναι σύνθετα είτε όχι.



Εικόνα 2.17: Μοντέλα αρχιτεκτονικής BERT [30].

2.6.3 BERT - Τρόποι εκπαίδευσης της τεχνικής

Όπως και το μοντέλο του word2vec, προκειμένου να εξαχθούν αποτελέσματα από την τεχνική BERT πρέπει να εκπαιδευτούν τα μοντέλα αρχιτεκτονικής του. Οι τρόποι και πάλι ποικίλουν και το εξίσου σημαντικό είναι ότι δεν εκπαιδεύονται όπως ένα κλασικό Transformer. Ο ένας τρόπος είναι το μοντέλο Masked Language Modelling (MLM) και ο άλλος είναι το μοντέλο Next Sentence Prediction (NSP).

Πριν αναλυθούν οι δύο τρόποι, να επισημανθεί ότι υπάρχουν και κάποια κοινά στοιχεία στα μοντέλα. Για αρχή, και το MLM και το NSP χρησιμοποιούν το WordPiece tokenization. Εκτός από τον διαχωρισμό των κειμένων σε προτάσεις και κατ' επέκταση σε λέξεις, ο συγκεκριμένος αλγόριθμος διαχωρίζει και τις λέξεις με τις "περίεργες καταλήξεις" τους. Παραδείγματος χάριν, χωρίζει τη λέξη "raining" σε "rain" και "ing". Αυτό γίνεται προκειμένου να μην μπερδευτεί η τεχνική σε σύνθετες ή/και μακροσκελείς λέξεις.

Το επόμενο κοινό στοιχείο των δύο μοντέλων είναι ότι το BERT με στόχο να δουλέψει άρτια, προτιμά να έχει όση περισσότερη πληροφορία είναι δυνατή. Για αυτόν τον λόγο προσθέτει κάποια ειδικά σύμβολα. Μερικά από αυτά είναι το [CLS] το οποίο μπαίνει στην αρχή του κειμένου και συμβολίζει την ταξινόμιση (classification) και το [SEP], το οποίο χρησιμοποιείται για να φανεί που διαχωρίζονται οι προτάσεις μεταξύ τους και συμβολίζει τον διαχωρισμό (separation) [32].

2.6.4 BERT - Masked Language Modelling (MLM)

Αφού έχουν προηγηθεί τα δύο παραπάνω βήματα, το MLM (Masked Language Modelling) επιλέγει τυχαία ένα ποσοστό λέξεων και το κρύβει. Συγκεκριμένα επιλέγει το 15% των λέξεων που υπάρχει και τις εμφανίζει με νέο συμβολισμό, [MASK]. Με βάση τις λέξεις που δεν είναι κρυμμένες και το

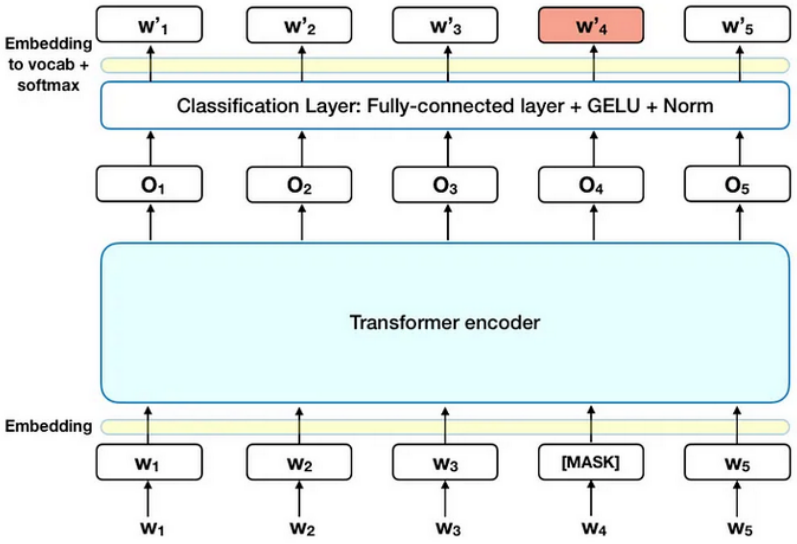
νόημα αυτών, το μοντέλο προσπαθεί να προβλέψει ποιες είναι οι λέξεις που έχουν κρυφτεί, τελικά. Αυτή η διαδικασία απεικονίζεται ενδεικτικά στην Εικόνα 2.18 και βοηθά το BERT να κατανοήσει τον τρόπο σύνδεσης των λέξεων.

Αξίζει να σημειωθεί, ωστόσο, ότι σύμφωνα με τον Jacob Devlin (ερευνητή της Google) το ποσοστό των λέξεων που εμφανίζεται με το [MASK] δεν είναι τυχαίο. Αν χρησιμοποιηθεί μικρότερο ποσοστό του 15 ($< 15\%$), το μοντέλο θα είναι πολύ δύσκολο να εκπαιδευτεί. Αντίστοιχα, αν το ποσοστό είναι μεγαλύτερο του 15 ($> 15\%$) δε δίνονται αρκετά συμφραζόμενα για να μπορέσει το μοντέλο να πάρει αποτελέσματα [33]. Το ποσοστό 15% θεωρείται καλός συμβιβασμός, καθώς επιτρέπει στο μοντέλο να έχει επαρκή συμφραζόμενα για να μάθει τη σύνδεση μεταξύ των λέξεων.

Παρ' όλα αυτά, ο Jacob Devlin τόνισε ότι ακόμα και το 15% δεν είναι το τέλειο ποσοστό. Πρακτικά αξιοποιείται αυτό, όντως ως το ποσοστό που προαναφέρθηκε, ωστόσο από αυτό το ποσοστό δεν είναι τελικά όλες οι λέξεις "κρυμμένες" με το [MASK]. Στο 80% των περιπτώσεων, οι λέξεις κρύβονται όντως με τη λέξη [MASK]. Παρ' όλα αυτά στο υπόλοιπο 20% συμβαίνει κάτι διαφορετικό. Το πρώτο μισό του υπόλοιπου (10%) λαμβάνει ως "όρισμα" μία τυχαία λέξη και το άλλο μισό (10%) κρατάει τη λέξη που έχει εξ αρχής.

Παραδείγματος χάριν, έστω η πρόταση "Η θάλασσα είναι μπλε".

- **Στο** 80% **των περιπτώσεων θα προκύψει:** *"Η θάλασσα είναι [MASK]".*
- **Στο** 10% **των περιπτώσεων θα προκύψει:** *"Η θάλασσα είναι τραίνο".*
- **Στο άλλο** 10% **των περιπτώσεων θα προκύψει** *"Η θάλασσα είναι μπλε".*



Εικόνα 2.18: Τρόπος λειτουργίας του μοντέλου MLM [34].

Ένα πιθανό μειονέκτημα του MLM είναι ότι το BERT δίνει έμφαση μόνο στις λέξεις που έχουν καλυφτεί με το σύμβολο του [MASK]. Αγνοεί, δηλαδή, να προβλέψει τις λέξεις που εμφανίζονται κανονικά [34]. Αυτό αποτελεί αρνητικό, καθώς το BERT δεν έχει, πιθανώς, κατανοήσει πλήρως το συνολικό κείμενο, κατά τη διάρκεια που προσπαθεί να προβλέψει τις κρυμμένες λέξεις.

2.6.5 BERT – Next Sentence Prediction (NSP)

Το μοντέλο NSP αποτελεί αρκτικόλεξο του Next Sentence Prediction και κάνει ό,τι αναφέρει το όνομα του. Συγκεκριμένα, το NSP λαμβάνει ως ορίσματα δύο (ένα ζεύγος) προτάσεις και ο στόχος του είναι να καταλάβει αν η δεύτερη πρόταση είναι η φυσική συνέχεια της πρώτης. Μαντεύει, δηλαδή, αν η δεύτερη πρόταση είναι η επόμενη στο δοθέν έγγραφο σε σχέση με την πρώτη από το ζεύγος που εισήχθηκε αρχικά.

Με σκοπό να εκπαιδευτεί, το μοντέλο χωρίζει τα ζευγάρια ανάλογα με το από τι αποτελείται η δεύτερη πρόταση. Πιο αναλυτικά, τα μισά (50%) ζευγάρια που δίνονται ως input είναι όντως δύο διαδοχικές προτάσεις που υπάρχουν στο αρχικό κείμενο. Στις άλλες μισές (50%) input προτάσεις,

ωστόσο, η δεύτερη πρόταση είναι μία τυχαία σε σχέση με τη νοηματική συνέχεια του κειμένου. Η υπόθεση είναι ότι στη δεύτερη περίπτωση οι δύο προτάσεις θα διαχωριστούν.

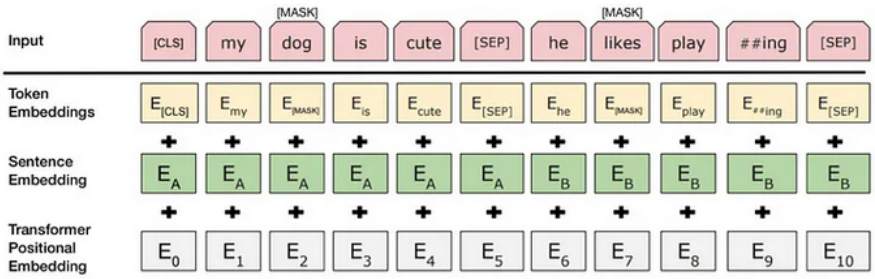
Προκειμένου να μπορέσει να διακρίνει το μοντέλο το πού ξεκινά και πού ολοκληρώνεται η κάθε πρόταση, ανά ζευγάρι μελέτης, ακολουθεί μία σειρά βημάτων. Αρχικά, όπως αναφέρθηκε και στην αρχή των τρόπων εκπαίδευσης, θα προστεθούν τα σύμβολα [CLS] και [SEP] στην αρχή του input και στο τέλος κάθε πρότασης αντιστοίχως. Έπειτα γίνεται ο διαχωρισμός των δύο προτάσεων σε πρόταση Α και πρόταση Β. Και τέλος, θα δημιουργηθεί ένα embedding σε κάθε πρόταση για να υποδείξει τη θέση τους στην ακολουθία.

Προκειμένου να προβλεφθεί αν η υπόθεση δουλεύει ή όχι, εκτελούνται μερικά ακόμα βήματα. Αναλυτικότερα, σύμφωνα με τον Rani Horev στο άρθρο "BERT Explained: State of the art language model for NLP" [34]:

- Το input διέρχεται σε ένα Transformer.
- Η έξοδος του συμβόλου [CLS] μετασχηματίζεται σε ένα διάνυσμα σχήματος 2×1, χρησιμοποιώντας ένα απλό στρώμα ταξινόμησης.
- Τελικώς υπολογίζεται η πιθανότητα του IsNextSequence με τη συνάρτηση Softmax.

Ενδεικτικά ο σχηματικός τρόπος λειτουργίας του μοντέλου NSP εμφανίζεται στην Εικόνα 2.19.

Κλείνοντας, το NSP είναι σημαντικό για την εκπαίδευση του BERT, καθώς βοηθά στην καλύτερη κατανόηση του κειμένου. Το BERT, με αυτόν τον τρόπο, μπορεί να διαχειριστεί σύνθετα κείμενα και να δίνει ακριβή αποτελέσματα στις προβλέψεις του. Για αυτόν τον λόγο είναι σημαντικό και για αρκετές εφαρμογές του NLP.



Εικόνα 2.19: Τρόπος λειτουργίας του μοντέλου NSP [34].

2.6.6 BERT – Εφαρμογές στην καθημερινότητα

Το εντυπωσιακό με την τεχνική BERT είναι ότι η ίδια έχει πολλές παραλλαγές. Δηλαδή ανάλογα με το είδος του κειμένου που λαμβάνει ως όρισμα, μπορεί να επιλεχθεί η καταλληλότερη BERT παραλλαγή. Αυτό γίνεται, καθώς όπως ειπώθηκε και παραπάνω η τεχνική επιζητά τη μεγαλύτερη ακρίβεια. Μερικές από τις παραλλαγές και οι εφαρμογές του BERT είναι οι εξής, κλείνοντας το θεωρητικό κομμάτι της πτυχιακής αυτής:

- **BERTSUM ή αλλιώς Text Summarization:** *Η τεχνική μπορεί να δημιουργήσει περιλήψεις κειμένων*
- **Google Smart Search:** *Το BERT βοήθησε την αναζήτηση της μηχανής της Google ούτως ώστε να προκύπτουν ακόμα πιο σχετικά αποτελέσματα σύμφωνα με την αναζήτηση του χρήστη.*
- **ScieBERT:** *Δημιουργήθηκε από 1,4 εκατομμύρια δημοσιεύσεις, εκ των οποίων το* 18% *είναι σχετικό με τον κλάδο των υπολογιστών και το υπόλοιπο* 82% *με τον κλάδο της βιοϊατρικής. Στόχος του ScieBERT είναι να συγκεντρωθεί σε έργα που είναι συσχετισμένα με την επιστήμη [35].*
- **BioBERT:** *Η κατηγορία αυτή υπερτερεί συγκριτικά με το γενικό BERT, καθώς αφορά συγκεκριμένα τη χρήση της NLP για βιοϊατρικά κείμενα. Το κοινό χαρακτηριστικό με το ScieBERT είναι ότι και οι δύο παραλλαγές αναπτύχθηκαν για εξειδικευμένους τομείς που δεν καλύπτονται επαρκώς από το γενικό μοντέλο του BERT.*

- **ClinicalBERT:** Σύμφωνα με το άρθρο "ClinicalBERT: Modeling Clinical Notesand Predicting Hospital Readmission" [36] αυτή η παραλλαγή του BERT μοντελοποιεί τις κλινικές σημειώσεις και προβλέπει την επανεισαγωγή στο νοσοκομείο με την ενσωμάτωση κλινικών κειμένων/σημειώσεων σε πλαίσιο.

Παρατηρώντας μερικές μόνο από τις παραλλαγές του BERT, γίνεται κατανοητό ότι το BERT αποτελεί ένα πολύ σημαντικό εργαλείο. Με τις πολλές παραλλαγές του, εξυπηρετεί ακόμα καλύτερα εξειδικευμένες ανάγκες σε πολλούς τομείς, γεγονός που δε θα ήταν δυνατόν με το γενικό μοντέλο του BERT. Αυτή η ποικιλία κατατάσσει το BERT ως ένα από τα πιο δυνατά εργαλεία που εξυπηρετεί η NLP.

2.7 Σύγκριση τεχνικών word2vec και BERT

Κλείνοντας το θεωρητικό κομμάτι της εργασίας, θα ακολουθήσει μία σύγκριση των τεχνικών word2vec και BERT:

Κοινά στοιχεία των τεχνικών word2vec και BERT:

- Και τα δύο είναι ευρέως διαδεδομένες τεχνικές αναπαράστασης διανυσμάτων λέξεων που χρησιμοποιούνται στη NLP.
- Και οι δύο τεχνικές προέρχονται από ερευνητές της εταιρείας της Google.

Διαφορές των τεχνικών word2vec και BERT:

- Το BERT είναι πιο σύγχρονη τεχνική (2018) σε σύγκριση με το word2vec (2013), αν και οι δύο είναι σχετικά νέες τεχνικές.
- Στο word2vec χρησιμοποιούνται Shallow NNs, σε αντίθεση με το BERT που βασίζεται στην αρχιτεκτονική των Transformers, που αποτελείται από 12 στρώματα (DNNs).
- Το BERT είναι και language representation model, δηλαδή μοντέλο αναπαράστασης γλώσσας εκτός από μοντέλο ενσωμάτωσης όπως το word2vec, καθώς προσφέρει και προτασιακή αναπαράσταση.
- Ως αφορά τα διανύσματα, το word2vec αποθηκεύει ένα διάνυσμα ως αναπαράσταση μίας λέξης, ενώ το BERT παράγει ένα διάνυσμα για μία λέξη συλλέγοντας πληροφορίες από τις υπόλοιπες λέξεις που ανήκουν στην πρόταση που ανήκουν [37].
- Το BERT λαμβάνει δεδομένα από τα συμφραζόμενα των λέξεων, για να κατανοήσει τη σωστή σημασία της λέξης. Έτσι μειώνει την πιθανότητα λάθους, στην περίπτωση που μία λέξη έχει πολλές έννοιες.
- Το word2vec χρησιμοποιείται συχνότερα σε μικρές και απλές εργασίες, όπως η ταξινόμηση εγγράφων. Αντίθετα, το BERT χρησιμοποιείται σε πιο σύνθετες εργασίες, όπως εξαγωγή συμπερασμάτων φυσικής γλώσσας.

Κεφάλαιο 3

Πρακτικό μέρος

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστούν μερικά παραδείγματα γραμμένα σε κώδικα στην προγραμματιστική γλώσσα python. Θα φανεί πώς εφαρμόζεται πρακτικά η τεχνική word2vec και η τεχνική BERT.

3.1 Παραδείγματα στην τεχνική Word2vec

3.1.1 Παράδειγμα word2vec χωρίς tokenization

Στο πρώτο παράδειγμα θα παρατηρηθεί ο τρόπος με τον οποίο βρίσκονται λέξεις με κοντινό νόημα, με την τεχνική word2vec. Στην αρχή θα δηλωθούν, όπως φαίνεται στον Κώδικα 3.1, τα πακέτα που πρέπει να ενταχθούν για να δουλέψει η τεχνική word2vec, τα οποία θα αξιοποιηθούν και στα επόμενα παραδείγματα.

```
from gensim.models import Word2Vec
```

Κώδικας 3.1: Εισαγωγή πακέτων word2vec

Το ιδιαίτερο στο συγκεκριμένο παράδειγμα είναι ότι το corpus ("το σώμα των κειμένων") θα πρέπει να το ορίσει ο προγραμματιστής εξ αρχής "λέξη-προς-λέξη". Στην προκείμενη περίπτωση θα εισαχθούν οι προτάσεις "The cat is my pet" και "The dog barked at my cat" όπως φαίνεται στον Κώδικα 3.2.

```
corpus = [
      ["the", "cat", "is", "my", "pet"],
      ["the", "dog", "barked", "at", "my", "cat"]
    ]
```

Κώδικας 3.2: Ορισμός του corpus ανά λέξη

Αφού έχει οριστεί επιτυχώς και το corpus, θα οριστεί το μοντέλο που θα προετοιμάσει το word2vec σύμφωνα με το corpus και θα οριστεί συνάρτηση για να βρεθούν λέξεις με κοντινό νόημα σύμφωνα με το corpus για μία συγκεκριμένη λέξη, εδώ όπως φαίνεται και στον Κώδικα 3.3 τη λέξη pet. Τέλος, αφού έχουν υπολογιστεί οι κοντινές σχέσεις των λέξεων με τη μία που έχει τεθεί, θα εμφανιστούν οι λέξεις με τη μεγαλύτερη συγγένεια, ο αριθμός των οποίων θα οριστεί από των προγραμματιστή όπως φαίνεται στον Κώδικα 3.3.

```
model = Word2Vec(corpus, min_count=1)

similar_words = model.wv.most_similar("pet")

for word, score in similar_words[:3]:

    print(word, score)
```

Κώδικας 3.3: Προετοιμασία, εύρεση και εμφάνιση των λέξεων με κοντινό νόημα

Ενδεικτικά, τα αποτελέσματα που προκύπτουν παρατίθενται παρακάτω. Όπως παρατηρείται οι λέξεις που αποτυπώνονται με τη μεγαλύτερη συγγένεια είναι οι λέξεις the, at, barked.

- the 0.1459505707025528
- at 0.041577354073524475
- barked 0.03476494178175926

3.1.2 Παράδειγμα word2vec με tokenization

Τι γίνεται στην περίπτωση, όμως, που το κείμενο που πρέπει να αναλυθεί είναι λίγο μεγαλύτερο και το σενάριο του να περαστούν όλες οι λέξεις μία-μία είναι αδύνατο; Εδώ εισέρχεται το πακέτο της NLP που διαχωρίζει τα κείμενα σε προτάσεις και τις προτάσεις σε λέξεις, όπως φαίνεται και στον Κώδικα 3.5.

```
from gensim.models import Word2Vec

from nltk.tokenize import sent_tokenize, word_tokenize
```

Κώδικα 3.5: Εισαγωγή πακέτων tokenization

Αφού εισαχθούν τα πακέτα, ο προγραμματιστής αρκεί να διαχωρίσει τις προτάσεις μέσα στη λίστα "sentences" και ορίζοντας την εντολή word lists που αναπαριστάται στον Κώδικα 3.6, θα χωριστούν οι προτάσεις σε λέξεις αυτόματα. Το αποτέλεσμα παρατίθεται στον Κώδικα 3.7

```
sentences = [

    "The cat is my pet",

    "The dog barked at my cat"

]

word_lists = [sentence.lower().split() for sentence in sentences]

print (word_lists)
```

Κώδικας 3.6: Εντολή sentence και tokenizing κειμένο

```
[['the', 'cat', 'is', 'my', 'pet'], ['the', 'dog', 'barked', 'at', 'my', 'cat']]
```

Κώδικας 3.7: Αποτελέσματα word2vec με tokenization

3.1.3 Παράδειγμα king – man + woman

Ένα παράδειγμα που αναλύθηκε στο θεωρητικό μέρος της πτυχιακής. Προκειμένου να υπολογιστεί το διάνυσμα, χωρίς να δοθεί κανένα κείμενο, θα αξιοποιηθεί ένα pre-trained μοντέλο ενσωμάτωσης λέξεων το οποίο παράχθηκε από την εταιρεία της Google. Το μοντέλο με όνομα "word2vec-google-news-300" εμπεριέχει διανυσματικές λέξεις για περισσότερες από τρεις εκατομμύρια λέξεις. Σύμφωνα με τον Binod Suman στο αρχείο κώδικά του για το word2vec στην πλατφόρμα του GitHub, τα

διανύσματα του μοντέλου της Google έχουν τριακόσιες (300) διαστάσεις και έχουν προετοιμαστεί στο μεγαλύτερο dataset με άρθρα [38].

Όπως αναπαριστάται στον Κώδικα 3.8, το μοντέλο της Google δεν εισάγεται ως πακέτο. Αυτήν τη φορά το καινούργιο πακέτο εισάγει τον υπολογισμό του συνημιτόνου των διανυσμάτων των λέξεων.

```
import gensim.downloader as api

from sklearn.metrics.pairwise import cosine_similarity

model = api.load("word2vec-google-news-300")
```

Κώδικας 3.8: Εισαγωγή μοντέλου “word2vec-google-news-300” και του πακέτου υπολογισμού του συνημιτόνου

Αφού έχουν οριστεί τα βασικά, θα οριστούν οι έννοιες king, man, woman, queen και το result. Τα πρώτα τέσσερα θα λάβουν τα διανύσματα σύμφωνα με το μοντέλο της Google, ενώ το result θα οριστεί από τις πράξεις king – man + woman, όπως φαίνεται και στον κώδικα 3.9.

```
king = model['king']

man = model['man']

woman = model['woman']

queen = model['queen']

result = king - man + woman
```

Κώδικας 3.9: Ορισμός εννοιών king, man, woman, queen, result

Τέλος, θα δοθεί η συνάρτηση υπολογισμού του συνημιτόνου του result και του queen, για να ερευνηθεί πόσο ταυτίζονται οι έννοιες στον Κώδικα 3.10 και τα αποτελέσματα αυτού στον Κώδικα 3.11. Στον Κώδικα 3.10 φαίνεται ότι η συνάρτηση του king – man + woman υπολογίζεται και με δεύτερο τρόπο. Με το συγκεκριμένο μοντέλο, όπως απεικονίζεται στον Κώδικα 3.11, το αποτέλεσμα είναι μία σειρά λέξεων, οι οποίες σύμφωνα με το αποτέλεσμα του συνημιτόνου κατατάσσονται σε νοηματικά πιο κοντινές.

```
similarity = cosine_similarity(result.reshape(1,-1), queen.reshape(1,-
1))[0][0]

print("Cosine similarity : ", similarity)

model.most_similar(positive=['king','woman'],negative=['man'])
```

Κώδικας 3.10: Υπολογισμός συνημιτόνου result, queen και king – man + queen

```
Cosine similarity :  0.7300518

[('queen', 0.7118193507194519),

 ('monarch', 0.6189674139022827),

 ('princess', 0.5902431011199951),

 ('crown_prince', 0.5499460697174072),

 ('prince', 0.5377321839332581),

 ('kings', 0.5236844420433044),

 ('Queen_Consort', 0.5235945582389832),

 ('queens', 0.5181134343147278),

 ('sultan', 0.5098593831062317),

 ('monarchy', 0.5087411999702454)]
```

Κώδικας 3.11: Αποτελέσματα παραδείγματος king – man + woman

3.1.4 Παράδειγμα CBOW και Skip-Gram

Στο παράδειγμα αυτό ο κώδικας θα γίνει ακόμα πιο σύνθετος. Για αρχή, όπως παρατηρείται στον Κώδικα 3.12, θα ενταχθούν όλα τα πακέτα που χρησιμοποιήθηκαν στα παραπάνω παραδείγματα. Τα πακετά είναι υπεύθυνα για την υλοποίηση του word2vec, του tokenization και αυτήν τη φορά θα εισαχθεί και μία εντολή σύμφωνα με την οποία ο κώδικας θα αγνοήσει τα warnings που ενδεχεται να συναντήσει.

```
from gensim.models import Word2Vec

import gensim

from nltk.tokenize import sent_tokenize, word_tokenize

import nltk

nltk.download('punkt')

import warnings

warnings.filterwarnings(action='ignore')
```

Κώδικας 3.12: Εισαγωγή πακέτων παραδείγματος CBOW και Skip-Gram

Αφού έχουν εισαχθεί τα πακέτα, θα περαστεί το κείμενο που θα επεξεργαστεί. Συγκεκριμένα με τον Κώδικα 3.13, παρατηρείται ότι μπορούν να δοθούν ολόκληρα κείμενα, μορφοποιημένα και μη. Στην περίπτωση του παραδείγματος αυτού έχει δοθεί το βιβλίο "Η Αλίκη στη χώρα των θαυμάτων". Ταυτόχρονα θα αντικατασταθεί ο χαρακτήρας escape με το κενό (space).

```
sample = open("alice.txt")

f = s.replace("\n", " ")

data = []
```

Κώδικας 3.13: Εισαγωγή του κειμένου στο κώδικα

Από αυτό το σημείο ξεκινάει η διαδικασία μετατροπής του κειμένου σε μεμονωμένες λέξεις. Αρχικά ο αλγόριθμος θα πάρει όλο το κείμενο και θα το μετατρέψει σε προτάσεις και στη συνέχεια θα μετατρέψει κάθε πρότασης σε μεμονωμένες λέξεις. Όπως αποτυπώνεται και στον Κώδικα 3.14 γίνεται χρήση του tokenize, ενώ πλέον οι λέξεις αποτελούνται και από πεζά γράμματα μόνο.

```
for i in sent_tokenize(f):

  temp = []

  for j in word_tokenize(i):

    temp.append(j.lower())

  data.append(temp)
```

Κώδικας 3.14: Διαδικασία tokenize του κειμένου

Το επόμενο στάδιο είναι το πιο σημαντικό. Σε αυτό το σημείο θα υπολογιστούν τα δύο μοντέλα CBOW και Skip-Gram. Για να οριστούν τα μοντέλα όπως διαπιστώνεται και από τον Κώδικα 3.15 εμπεριέχουν τέσσερα-πέντε στοιχεία. Τα κοινά στοιχεία των CBOW, Skip-Gram είναι οι τιμές:

- **Data:** *τα δεδομένα του κειμένου όπως αναλύθηκαν στον Κώδικα 3.14.*
- **min count:** *η ελάχιστη ποσότητα εμφάνισης μίας λέξης που χρειάζεται για να ενταχθεί στο λεξιλόγιο της τεχνικής. Εδώ προκύπτει το συμπέρασμα ότι στο λεξιλόγιο θα ενταχθούν όλες οι λέξεις του κειμένου που θα παρθούν από το data, αφού πρέπει να εμφανιστούν τουλάχιστον μία φορά.*
- **vector size:** *το μέγεθος/διάσταση του διανύσματος που θα έχει κάθε λέξη. Εδώ δηλαδή το διάνυσμα κάθε λέξης θα έχει διάσταση 100.*
- **window:** *το παράθυρο λέξεων που βοηθά στην εκπαίδευση του μοντέλου. Δηλαδή, εδώ, με window = 5 η τεχνική αυτή θα κοιτάξει τις 5 προηγούμενες και 5 επόμενες λέξεις από αυτήν που μελετάται εκείνη τη στιγμή.*

- **sg=1:** *όσο η τιμή sg είναι 1 τότε το word2vec υλοποιείται με την τεχνική του Skip-Gram. Αν είναι 0 χρησιμοποιείται η τεχνική του CBOW.*

```
model1 = gensim.models.Word2Vec(data, min_count=1,
                                vector_size=100, window=5)
print("Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' " +
      "και 'rabbit' σύμφωνα με την τεχνική CBOW : ",
      model1.wv.similarity('alice', 'rabbit'))
print("Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' " +
      "και 'alice' σύμφωνα με την τεχνική CBOW : ",
      model1.wv.similarity('alice', 'alice'))
model2 = gensim.models.Word2Vec(data, min_count=1, vector_size=100,
                                window=5, sg=1)
```

```
print("Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' " +
      "και 'rabbit' σύμφωνα με την τεχνική Skip Gram : ",
      model2.wv.similarity('alice', 'rabbit'))
print("Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' " +
      "και 'alice' σύμφωνα με την τεχνική Skip Gram : ",
       model2.wv.similarity('alice', 'alice'))
```

Κώδικας 3.15: Υπολογισμός μοντέλων CBOW και Skip-Gram

Στο Κώδικα 3.16 αποτυπώνονται τα αποτελέσματα των CBOW και Skip-Gram. Παρατηρείται ότι στη σύγκριση “alice”, “alice”, το αποτέλεσμα και στα δύο μοντέλα είναι το αναμενόμενο, δηλαδή 1 (πλήρη ταύτιση). Αντίστοιχα, με το μοντέλο CBOW η σύνδεση των “alice”, ”rabbit” είναι μεγαλύτερη συγκριτικά με το μοντέλο Skip-Gram.

```
Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' και 'rabbit' σύμφωνα με την
τεχνική CBOW :  0.9991814

Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' και 'alice' σύμφωνα με την
τεχνική CBOW :  1.0

Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' και 'rabbit' σύμφωνα με την
τεχνική Skip Gram :  0.9230797

Η ομοιότητα συνημιτόνου μεταξύ των λέξεων 'alice' και 'alice' σύμφωνα με την
τεχνική Skip Gram :  1.0
```

Κώδικας 3.16: Αποτελέσματα παραδείγματος CBOW και Skip-Gram

Κεφάλαιο 4

Σύνοψη

4.1 Συμπεράσματα

Στην παρούσα πτυχιακή εργασία μελετήθηκαν δύο από τις πιο σημαντικές και διαδεδομενές τεχνικές αυτόματης εξαγωγής λέξεων και φράσεων από κείμενα, συγκεκριμένα το Word2Vec και το BERT. Αρχικά, η τεχνική Word2Vec, με μοντέλα εκτέλεσης τα CBOW και Skip-Gram, αποδείχθηκε αποτελεσματική στην ταχεία αναπαράσταση των λέξεων μέσω διανυσμάτων. Αντίστοιχα, το BERT, βασισμένο στους μετασχηματιστές (Transformers), πρόσφερε μεγαλύτερη ακρίβεια και κατανόηση του περιεχομένου των λέξεων, με την τεχνική να εφαρμόζεται καλύτερα σε πιο σύνθετες γλωσσικές δομές.

Τα αποτελέσματα της ανάλυσης των τεχνικών αυτών έδειξαν ότι, ενώ το Word2Vec είναι ταχύτερο και πιο απλό στην υλοποίηση, το BERT υπερτερεί σε θέματα ακρίβειας, ιδιαίτερα σε εφαρμογές που απαιτούν καλύτερη κατανόηση του συμφραζόμενου των λέξεων. Για το λόγο αυτό, η επιλογή της κατάλληλης τεχνικής εξαρτάται από τις απαιτήσεις της κάθε εφαρμογής. Το Word2Vec είναι ιδανικό για εφαρμογές που απαιτούν ταχύτητα και απόδοση, και το BERT είναι προτιμητέο για εργασίες που απαιτούν βαθύτερη ανάλυση των γλωσσικών συνδέσεων των λέξεων της εφαρμογής.

Μέσω της εργασίας αυτής αποτυπώνεται η σημασία της αυτόματης εξαγωγής λέξεων και φράσεων. Με μερικές εφαρμογές να καταγράφονται στη βελτίωση της αναζήτησης πληροφοριών, της δημιουργίας περιλήψεων και της ανάλυσης συναισθημάτων, προσφέροντας νέες δυνατότητες στην επεξεργασία μεγάλου όγκου δεδομένων. Η υλοποίηση των τεχνικών αυτών μπορεί να προσφέρει και πολύτιμες λύσεις σε ένα ευρύ φάσμα εφαρμογών, από τη βελτίωση των μηχανών αναζήτησης μέχρι την ανάλυση κειμένων σε επιχειρηματικό ή επιστημονικό πλαίσιο.

4.2 Μελλοντικές επεκτάσεις

Η παρούσα εργασία αφήνει ανοιχτό το πεδίο για μελλοντικές επεκτάσεις και βελτιώσεις. Περαιτέρω έρευνα θα μπορούσε να εστιάσει στη βελτιστοποίηση των τεχνικών αυτών για συγκεκριμένες γλώσσες ή κλάδους, όπως για παράδειγμα η ελληνική γλώσσα, η οποία παρουσιάζει ιδιαιτερότητες στη μορφολογία και τη σύνταξή της. Επιπλέον, η ενσωμάτωση πιο σύγχρονων τεχνικών, όπως οι GPT (Generative Pretrained Transformers), θα μπορούσε να εξεταστεί για την περαιτέρω βελτίωση της κατανόησης των γλωσσικών δεδομένων και την παραγωγή πιο ακριβών αποτελεσμάτων.

Τέλος, οι τεχνικές της αυτόματης εξαγωγής θα μπορούσαν να συνδυαστούν με άλλες τεχνολογίες, όπως η ανάλυση εικόνας ή βίντεο, ώστε να επιτευχθεί ένας πολυμεσικός τρόπος κατανόησης και εξαγωγής πληροφορίας από διάφορες πηγές δεδομένων.

Κεφάλαιο 5

Βιβλιογραφία

[1] Manning, C. (2020). Artificial Intelligence Definitions. Stanford University. https://hai.stanford.edu/sites/default/files/2020-09/AI-Definitions-HAI.pdf (Προσπέλαση Μάρτιος 1, 2024)

[2] Gulley, A. & Hilliard, A. (2024). Lost in Transl(A)t(I)on: Differing Definitions of AI [Updated]. https://www.holisticai.com/blog/ai-definition-comparison (Προσπέλαση Μάρτιος 1, 2024)

[3] Brown, E. C. & O'Leary, E. D. (1995). Introduction to artificial intelligence and expert systems. https://msbfile03.usc.edu/digitalmeasures/doleary/intellcont/Brown-Oleary-es_tutor -1.htm (Προσπέλαση Μάρτιος 5, 2024)

[4] Turing, A. M. (1950). Computing Machinery and Intelligence. Mind 49: 433-460.

[5] SAP (χχ). Τι είναι η τεχνητή νοημοσύνη; https://www.sap.com/greece/products/artificial-intelligence/what-is-artificial-intelligence.html (Προσπέλαση Ιούλιος 7, 2024)

[6] Tableau (χχ). The Ultimate Guide To Artificial Intelligence (AI): Definition, How It Works, Examples, History, & More. https://www.tableau.com/data-insights/ai/what-is (Προσπέλαση Ιούλιος 24, 2024)

[7] Ευρωπαϊκό Κοινοβούλιο (2020). Τι είναι η τεχνητή νοημοσύνη και πώς χρησιμοποιείται; https://www.europarl.europa.eu/topics/el/article/20200827STO85804/ti-einai-i-techniti-noimosuni-kai-pos-chrisimopoieitai (Προσπέλαση Απρίλιος 4, 2024)

[8] Wikipedia (χχ). Machine Learning. https://en.wikipedia.org/wiki/Machine_learning (Προσπέλαση Απρίλιος 6, 2024)

[9] IBM (χχ). What is machine learning (ML)? https://www.ibm.com/topics/machine-learning (Προσπέλαση Απρίλιος 6, 2024)

[10] Gupta, N. (2024). A guide to Supervised Learning. https://medium.com/@ngneha090/a-guide-to-supervised-learning-f2ddf1018ee0 (Προσπέλαση Απρίλιος 8, 2024)

[11] Kozan, M. (2021). Supervised and Unsupervised Learning (an Intuitive Approach). https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f 64b644 (Προσπέλαση Απρίλιος 8, 2024)

[12] Sharma, G. (2023). A gentle Introduction to Semi Supervised Learning. https://medium.com/@gayatri_sharma/a-gentle-introduction-to-semi-supervised-learning-7afa5539beea (Προσπέλαση Απρίλιος 15, 2024)

[13] Saini, S. (2021). Supervised vs. Unsupervised Learning: What's the Difference? https://www.linkedin.com/pulse/supervised-vs-unsupervised-learning-whats-difference-smriti-saini (Προσπέλαση Απρίλιος 15, 2024)

[14] IBM (χχ). What is deep learning? https://www.ibm.com/topics/deep-learning (Προσπέλαση Ιούνιος 15, 2024)

[15] Ronaghan, S. (2018). Introduction to Deep Learning: What do I need to know...? https://srnghn.medium.com/introduction-to-deep-learning-what-do-i-need-to-know-75794ebc4a62 (Προσπέλαση Ιούνιος 15, 2024)

[16] Greeco (χχ). Επεξεργασία Φυσικής Γλώσσας (NLP): Τι είναι και τι πλεονεκτήματα έχει. https://greeco.gr/business/techniti-noimosyni/epexergasoa-fusikis-glwssas/ (Προσπέλαση Ιούνιος 20, 2024)

[17] NLP Cloud (2021). Natural Language Processing Introduction: what is Natural Language Processing (NLP)? https://nlpcloud.com/introduction-what-is-nlp-natural-language-processing.html (Προσπέλαση Ιούνιος 20, 2024)

[18] IBM (χχ). What are word embeddings? https://www.ibm.com/topics/word-embeddings (Προσπέλαση Ιούλιος 20, 2024)

[19] Elastic (χχ). What are word embeddings? https://www.elastic.co/what-is/word-embedding (Προσπέλαση Ιούλιος 20, 2024)

[20] Logunova, I. (2023). Word2Vec: Why Do We Need Word Representations? https://serokell.io/blog/word2vec#what%E2%80%99s-the-difference-between-word-representation%2C-word-vectors%2C-and-word-embeddings%3F (Προσπέλαση Ιούλιος 25, 2024)

[21] Thorn, J. (2020). Deep Learning for NLP: Word Embeddings. https://towardsdatascience.com/deep-learning-for-nlp-word-embeddings-4f5c90bcdab5 (Προσπέλαση Ιούλιος 25, 2024)

[22] Xin, R. (2016). word2vec Parameter Learning Explained. https://arxiv.org/pdf/1411.2738

[23] Thulo, C. (2023). Understanding the Continuous Bag of Words (CBOW) Model: Architecture, Working Mechanism and Math Behind It — Natural language processing. https://medium.com/@codethulo/understanding-the-continuous-bag-of-words-cbow-model-architecture-working-mechanism-and-math-78c7284a8d5a (Προσπέλαση Ιούλιος 28, 2024)

[24] Karani, D. (2018). Introduction to Word Embedding and Word2Vec. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa (Προσπέλαση Ιούλιος 28, 2024)

[25] Analytics Vedhya (2024). Introduction to Seq2Seq Models. https://www.analyticsvidhya.com/blog/2020/08/a-simple-introduction-to-sequence-to-sequence-models/ (Προσπέλαση Ιούλιος 28, 2024)

[26] Kulshrestha, R. (2020). Transformers. https://towardsdatascience.com/transformers-89034557de14 (Προσπέλαση Ιούλιος 30, 2024)

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, N.A., Kaiser, L. & Polusukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems 30 (NIPS 2017). Ed. Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., S. Vishwanathan and Garnett, R. Curran Accosiates, Inc.

[28] Punyakeerthi BL (2024). Difference between Self-Attention and Multi-head Self-Attention. https://medium.com/@punya8147_26846/difference-between-self-attention-and-multi-head-self-attention-e33ebf4f3ee0 (Προσπέλαση Ιούλιος 30, 2024)

[29] IBM (χχ). What is a recurrent neural network (RNN)? https://www.ibm.com/topics/recurrent-neural-networks (Προσπέλαση Ιούλιος 30, 2024)

[30] Pickleprat (2024). Encoder Only Architecture: BERT. Bidirectional Encoder Representation Transformer. https://medium.com/@pickleprat/encoder-only-architecture-bert-4b27f9c76860 (Προσπέλαση Ιούλιος 30, 2024)

[31] Kumar, A. (2024). Transformer Architecture Types: Explained with Examples. https://vitalflux.com/transformer-architecture-types-explained-with-examples/ (Προσπέλαση Αύγουστος 1, 2024)

[32] Shaikh, R. (2023). Guide from Beginner to Advanced in Natural Language Processing (NLP). https://medium.com/@shaikhrayyan123/a-comprehensive-guide-to-understanding-bert-from-beginners-to-advanced-2379699e2b51 (Προσπέλαση Αύγουστος 1, 2024)

[33] Devlin, J. (χχ). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Bidirectional Encoder Representations from Transformers). https://computational-linguistics-class.org/slides/old/90-guest_lecture_jacob_devlin_bert_presentations.pdf (Προσπέλαση Αύγουστος 1, 2024)

[34] Horev, R. (2018). BERT Explained: State of the art language model for NLP. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 (Προσπέλαση Αύγουστος 1, 2024)

[35] Chakraborty, S. (2020). Practical Uses of BERT. https://sayanchak.medium.com/practical-uses-of-bert-c384ae3a5c2a (Προσπέλαση Αύγουστος 4, 2024)

[36] Huang, K., Altosaar, J. & Ranganath, R. (2020). ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. https://arxiv.org/pdf/1904.05342

[37] Ankiit. (2022). Word2vec vs BERT. https://medium.com/@ankiit/word2vec-vs-bert-d04ab3ade4c9 (Προσπέλαση Αύγουστος 10, 2024)