Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Αυτόματη Παραγωγή Περίληψης Κειμένου στην Ελληνική Γλώσσα με Αναδρομικά Νευρωνικά Δίκτυα Βαθιάς Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΗΤΡΟ ΝΙΚΟΣ

Επιβλέπων: Ιάκωβος Στ. Βενιέρης,

Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Αυτόματη Παραγωγή Περίληψης Κειμένου στην Ελληνική Γλώσσα με Αναδρομικά Νευρωνικά Δίκτυα Βαθιάς Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΗΤΡΟ ΝΙΚΟΣ

Επιβλέπων: Ιάκωβος Στ. Βενιέρης,

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14η Απριλίου 2021.

.....	.....	.....
Ι. Στ. Βενιέρης Καθηγητής Ε.Μ.Π.	Δ.-Θ. Κακλαμάνη Καθηγήτρια Ε.Μ.Π.	Γ. Ματσόπουλος Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2021

.....

Μήτρο Νίκος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών, Ε.Μ.Π.

Περίληψη

Σύστημα αυτόματης παραγωγής περίληψης ονομάζεται ένα σύστημα, το οποίο με τη χρήση κάποιου λογισμικού, παράγει ένα κείμενο μικρότερης έκτασης του αρχικού, και ταυτοχρόνως διατηρεί το αρχικό και βασικό νόημα του. Ο ολοένα και αυξανόμενος όγκος των δεδομένων που συναντάται στη σημερινή εποχή, καθιστά την ανάγκη ενός τέτοιου αυτόματου συστήματος ακόμα πιο αισθητή και ταυτοχρόνως ενθαρρύνει την ανάπτυξη λογισμικού για την προσπέλαση αυτού του μεγάλου όγκου δεδομένων.

Ο σχεδιασμός ενός εύρωστου συστήματος ικανό να παράγει νοηματικά, συντακτικά και γραμματικά ορθές περιλήψεις συγκεντρώνει μεγάλο ερευνητικό ενδιαφέρον, με αποτέλεσμα διάφοροι μηχανισμοί και εργαλεία λογισμικού να έχουν υλοποιηθεί για την επίλυση αυτού του προβλήματος. Κανένα όμως από αυτά τα εργαλεία δεν έχουν επιτύχει μέχρι σήμερα τη δημιουργία ενός γενικευμένου και ιδανικού συστήματος αυτόματης παραγωγής περίληψης, λόγω της δυσκολίας και της πολυπλοκότητας που εμφανίζει το συγκεκριμένο πρόβλημα. Ωστόσο, η ραγδαία ανάπτυξη της μηχανικής και ιδιαιτέρως της βαθιάς μάθησης, έχει συστήσει νέες τεχνικές αντιμετώπισης του προβλήματος, επιφέροντας βελτιωμένα αποτελέσματα.

Στη συγκεκριμένη εργασία, πραγματοποιείται μια πρώτη απόπειρα υλοποίησης ενός αποδοτικού συστήματος αυτόματης παραγωγής περίληψης για την ελληνική γλώσσα. Η υλοποίηση βασίζεται σε μηχανισμούς βαθιάς μάθησης και συγκεκριμένα σε αναδρομικά νευρωνικά δίκτυα και στην αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή. Για την προσαρμογή του συστήματος στην ελληνική γλώσσα, πραγματοποιείται σύγκρισή της με την αγγλική και μελέτη των ιδιομορφιών της σε συντακτικό και γραμματικό επίπεδο. Οι ιδιομορφίες αυτές αντιμετωπίζονται μέσω του λογισμικού στο στάδιο της προ-επεξεργασίας και στο στάδιο της παραγωγής των αριθμητικών αναπαραστάσεων των λέξεων.

Για την αξιολόγηση των σχεδιαστικών επιλογών που πάρθηκαν κατά την υλοποίηση του συστήματος και την εύρεση του βέλτιστου συνδυασμού τους, διερευνάται η επίδραση διάφορων βασικών παραμέτρων στην επίδοση του. Οι μετρήσεις της επίδοσης του συστήματος βασίζονται στις καθιερωμένες σε προβλήματα παραγωγής περίληψης μετρικές ROUGE. Από τα πειραματικά αποτελέσματα εξάγονται χρήσιμα συμπεράσματα για την λειτουργία και την αποτελεσματικότητα του συστήματος και αναδεικνύονται κάποιες ορθές κατευθύνσεις για το σχεδιασμό παρόμοιων συστημάτων. Τέλος, παρουσιάζεται ένας πιθανός τρόπος ενσωμάτωσης του υλοποιημένου συστήματος σε μια έξυπνη εφαρμογή για κινητά και προτείνονται κάποιες μελλοντικές κατευθύνσεις προώθησης της έρευνας πάνω στο αντικείμενο.

Λέξεις κλειδιά

Αυτόματη Περίληψη Κειμένων, Περίληψη Ελληνικών Κειμένων, Επεξεργασία Φυσικής Γλώσσας, Βαθιά Μάθηση, Αναδρομικά Νευρωνικά Δίκτυα, Αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή, Μετρικές ROUGE

Abstract

An automatic text summarization system refers to a system that uses software to produce a shortened version of a text document, preserving its initial and basic topic. Nowadays, the rapidly increasing volume of data makes the need of such automatic systems even more urgent and at the same time encourages the software development for the processing of all this information.

The design of a robust system capable of producing semantically, syntactically and grammatically correct summaries is a subject of great research interest. As a result, various software mechanisms and tools have been implemented to solve this problem. But none of these tools have so far succeeded in creating a general and ideal system of automatic summarization, due to the difficulty and complexity of this problem. However, the rapid development of machine learning and especially deep learning, has recommended new techniques to tackle the problem, providing improved results.

In this thesis, a first attempt is made to implement an efficient automatic summarization system for the Greek language. The implementation of this system is based on deep learning mechanisms, and specifically on recurrent neural networks and the encoder-decoder architecture. For the adaptation of the system to the Greek language, a comparison between the Greek and the English language is made on a syntactical and grammatical level and the specific features of the Greek language are highlighted. The problems that arise from these features are addressed through the software at the pre-processing of the Greek text data and at the production of the word embeddings.

In order to evaluate the design choices made during the implementation of the system, the impact of various key parameters is explored. The performance of the system is measured using the established metrics ROUGE, that have been used extensively on similar systems. Via analyzing the experimental results, useful conclusions about the functionality and the efficiency of the system are extracted and some good guidelines are highlighted for the design of similar systems. Finally, a way of integrating the implemented system into a smart mobile application is presented and some future directions are proposed, in order to promote the research on the subject.

Keywords

Automatic Text Summarization, Greek Text Summarization, Natural Language Processing, Deep Learning, Recurrent Neural Networks, Encoder-Decoder Architecture, ROUGE metrics

Ευχαριστίες

Η εκπόνηση της παρούσας διπλωματικής εργασίας πραγματοποιήθηκε στα πλαίσια των προπτυχιακών σπουδών μου στη σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Πριν προχωρήσω στην περιγραφή της εργασίας και των αποτελεσμάτων που προέκυψαν, θα ήθελα να ευχαριστήσω θερμά τους ανθρώπους που συνέβαλαν στην ολοκλήρωση της συγκεκριμένης εργασίας.

Αρχικά θα ήθελα να απευθύνω τις ευχαριστίες μου στον καθηγητή του Ε.Μ.Π. κύριο Ιάκωβο Στ. Βενιέρη, για την δυνατότητα που μου προσέφερε να εκπονήσω τη διπλωματική μου εργασία πάνω σε ένα ιδιαίτερα ενδιαφέρον αντικείμενο και να διευρύνω με αυτό το τρόπο τις επιστημονικές και τεχνολογικές μου γνώσεις. Επίσης, θέλω να ευχαριστήσω ιδιαίτερα το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας (ΕΔΥΤΕ), για τη προσφορά υπολογιστικών πόρων από το υπολογιστικό του σύστημα υψηλών επιδόσεων ARIS, χωρίς το οποίο δε θα ήταν δυνατή η ολοκλήρωση των πειραμάτων μας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου, η οποία στάθηκε δίπλα μου και με στήριξε όλα αυτά τα χρόνια, καθώς και τους φίλους και συμφοιτητές μου, οι οποίοι μέσω της συνεργασίας μας συνέβαλαν στην ολοκλήρωση της ακαδημαϊκής μου σταδιοδρομίας.

Αθήνα, Απρίλιος 2021

Μήτρο Νίκος

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1. Εισαγωγή

1.1 Περίληψη και η χρησιμότητά της

Περίληψη ή αλλιώς σύνοψη ενός κειμένου, είναι ένα νέο κείμενο μικρότερης έκτασης από το αρχικό, το οποίο αποτελεί μια περιεκτική και συνοπτική απόδοση του περιεχομένου του [1]. Μια περίληψη για να θεωρείτε ορθή, πρέπει να αποδίδει το βασικό μήνυμα του πρωτότυπου κειμένου, συμπεριλαμβάνοντας τα σημαντικότερα σημεία του, και να είναι νοηματικά και συντακτικά άρτια [2]. Επιπλέον, είναι σημαντικό ότι ενώ η περίληψη αποτελεί προσωπική δημιουργία του εκάστοτε συντάκτη της, πρέπει να είναι πληροφοριακή και αντικειμενική, χωρίς ίχνος σχολιασμού ή κριτικής. Τέλος, σημειώνεται ότι μια περίληψη μπορεί να προέρχεται από πολλά κείμενα τα οποία έχουν κοινό θεματικό υπόβαθρο.

Τα βασικότερα είδη περίληψης που συναντάμε είναι η ενδεικτική/περιγραφική (indicative) και η πληροφοριακή (informative) περίληψη. Η πληροφοριακή περίληψη παράγεται από τα σημαντικότερα σημεία ανά παράγραφο και καλύπτει σε έκταση περίπου το ένα τρίτο του πρωτότυπου κειμένου. Ενώ η περιγραφική περίληψη παράγεται από τα σημαντικότερα σημεία ανά νοηματική ενότητα, περιγράφει τα βασικά μηνύματα και καλύπτει προσεγγιστικά το ένα έκτο του πρωτότυπου κειμένου [3].

Ο κύριος στόχος της περίληψης και το βασικό πλεονέκτημα που προσφέρει είναι η μείωση του όγκου της πληροφορίας και κατά συνέπεια του χρόνου διαβάσματος. Το γεγονός αυτό τη καθιστά πολύ σημαντική, ιδιαίτερα στη σημερινή εποχή όπου ο όγκος πληροφοριών είναι τεράστιος. και για αυτό το λόγο χρησιμοποιείται ευρέως. Πλέον, κάθε είδος κειμένου που δημοσιεύεται (άρθρα, βιβλία, ακαδημαϊκές δημοσιεύσεις κ.τ.λ.), συνοδεύονται ως επί το πλείστον από τη περίληψή τους.

Τέλος, αξιοσημείωτο είναι το γεγονός ότι η παραγωγή της περίληψης είναι επωφελής και για τον συγγραφέα της. Η παραγωγή της περίληψης αποτελεί μια αφαιρετική διαδικασία, κατά την οποία ο συγγραφέας αναπτύσσει τη κριτική του σκέψη, τη λογική και τη μνήμη του, για να διακρίνει τα σημαντικά από τα περιττά σημεία του κειμένου και να τα παρουσιάσει κατάλληλα [4]. Για αυτό το λόγο η παραγωγή της περίληψης διδάσκεται στα πρώτα έτη της σχολικής ηλικίας.

1.2 Αυτόματη παραγωγή περίληψης

Η παραγωγή περίληψης κειμένου χρησιμοποιούταν ανέκαθεν από τους ανθρώπους ως εργαλείο ελαχιστοποίησης του όγκου της πληροφορίας, συγκεκριμένα γραπτού λόγου, με στόχο την ερμηνεία και κατανόηση περισσότερης πληροφορίας γρηγορότερα. Δεδομένου όμως ότι η παραγωγή περίληψης είναι μια διαδικασία χρονοβόρα και αρκετές φορές δύσκολη, η αυτοματοποίησή της καθίσταται πολύ σημαντική και αποτελεί ένα ισχυρό κίνητρο για ακαδημαϊκή έρευνα.

Η ανάγκη της αυτόματης παραγωγής περίληψης είναι ακόμα πιο εμφανής στην εποχή μας, όπου λόγω της ραγδαίας ανάπτυξης της τεχνολογίας και του διαδικτύου, υπάρχει ένας τεράστιος και αυξανόμενος όγκος κειμένων από διάφορες πηγές. Ο όγκος της πληροφορίας είναι τόσο μεγάλος πού καθίσταται πλέον αδύνατο για τους ανθρώπους η ελαχιστοποίηση του σε λογικό χρόνο, για αυτό το λόγο επιστρατεύονται οι υπολογιστές και εφαρμογές της τεχνητής νοημοσύνης (artificial intelligence), όπως η μηχανική μάθηση (machine learning).

Η αυτόματη παραγωγή περίληψης κειμένου ορίζεται λοιπόν ως η μοντελοποίηση ενός συστήματος αυτόματων κανόνων, για την παραγωγή ενός μικρότερου και άρτιου κειμένου που αποδίδει νοηματικά το αρχικό κείμενο, με τη χρήση λογισμικού και χωρίς τη βοήθεια του ανθρώπου. Ο σχεδιασμός και η υλοποίηση εύρωστων συστημάτων αυτόματης παραγωγής περίληψης ανήκει στο τομέα της επεξεργασίας φυσικής γλώσσας (Natural Language Processing - NLP) και αποτελεί ένα πρόβλημα που συγκεντρώνει όλο και περισσότερη δημοτικότητα και ερευνητικό ενδιαφέρον.

Τα συστήματα αυτόματης παραγωγής περίληψης χρησιμοποιούνται καθημερινά και αρκετές φορές ενσωματώνονται σε παρόμοια προβλήματα της επεξεργασίας φυσικής γλώσσας, όπως η κατηγοριοποίηση κειμένων (text categorization) [5], η αυτόματη εξαγωγή τίτλων σε κείμενα (headline generation) [6], η αυτόματη εξαγωγή απαντήσεων σε ερωτήσεις (questioning answering - QA) [7] και η ανάλυση συναισθημάτων από κείμενα (sentiment analysis) [8], όπου εφαρμόζεται σαν ενδιάμεσο στάδιο για τη μείωση της έκτασης του κειμένου.

Λαμβάνοντας υπόψιν τη συνήθη διαδικασία που ακολουθεί ένας άνθρωπος για την παραγωγή της περίληψης ενός κειμένου, όπου αρχικά διαβάζει ολόκληρό το κείμενο για να το κατανοήσει και μετά εντοπίζει τα σημαντικότερα σημεία του, ευκολά μπορεί κανείς να συμπεράνει ότι η αυτόματη παραγωγή περίληψης αποτελεί μια περίπλοκη και μη ντετερμινιστική διαδικασία. Κατά συνέπεια, ο υπολογιστής ως ντετερμινιστική μηχανή που λειτουργεί με κανόνες και όχι με την ανθρώπινη λογική, αδυνατεί να διαχωρίσει τις ιδιομορφίες κάθε κειμένου και να αναπτύξει ένα νοηματικά και συντακτικά ορθό κείμενο. Για αυτό το λόγο σε μια προσπάθεια προσομοίωσης της ανθρώπινης λογικής χρησιμοποιούνται μέθοδοι τεχνητής νοημοσύνης (artificial intelligence), και συγκεκριμένα βαθιάς μάθησης(deep learning).

1.3 Σχετιζόμενη ερευνητική δραστηριότητα

Το αντικείμενο της αυτόματης παραγωγής περίληψης συγκεντρώνει μεγάλο ερευνητικό ενδιαφέρον τα τελευταία χρόνια, λόγω της χρησιμότητας και αναγκαιότητας των εφαρμογών του. Επιπλέον, αποτελεί ένα αντικείμενο το οποίο εξαιτίας της πολυπλοκότητας του, μπορεί να προσεγγιστεί με πολλούς διαφορετικούς τρόπους. Το γεγονός αυτό οδηγεί στην πραγματοποίηση πολλών διαφορετικών μεταξύ τους ερευνητικών εργασιών, οι οποίες ακολουθούν διαφορετικά μονοπάτια επίλυσης του προβλήματος. Τη μερίδα του λέοντος φαίνεται να συγκεντρώνουν οι τεχνικές της βαθιάς μάθησης, οι οποίες εμφανίζουν τη μεγαλύτερη αποτελεσματικότητα και είναι αυτές στις οποίες βασίζεται η παρούσα εργασία. Στη συνέχεια θα πραγματοποιηθεί μια σύντομη επισκόπηση των αξιοσημείωτων ερευνών που έχουν υλοποιηθεί τα τελευταία χρόνια, πάνω στο αντικείμενο της αυτόματης παραγωγής περίληψης.

Οι πρώτες έρευνες πάνω στο αντικείμενο βασίστηκαν σε μεθόδους επιλογής και συμπίεσης του αρχικού κειμένου. Για παράδειγμα, η αναφορά [9] πρότεινε ένα σύστημα, το οποίο αρχικά εξάγει φράσεις, αποτελούμενες από ουσιαστικά και ρήματα, από τις πρώτες προτάσεις ενός άρθρου και χρησιμοποιεί έναν επαναληπτικό αλγόριθμό για να τις συμπιέσει. Η έρευνα αυτή, όπως και η πλειοψηφία των ερευνών πριν την επέκταση της βαθιάς μάθησης, αποτελεί εξαγωγική παραγωγή περίληψης (extractive summarization), η οποία υλοποιείται με την εξαγωγή σημαντικών προτάσεων από το αρχικό κείμενο.

Με την άνθηση της βαθιάς μάθησης, και την αποτελεσματική χρήση των τεχνικών της σε πολλά προβλήματα της επεξεργασίας της φυσικής γλώσσας [10], δημιουργήθηκε μεγάλο ερευνητικό ενδιαφέρον στο σχεδιασμό συστημάτων αυτόματης παραγωγής περίληψης με παραγωγή αυτούσιων προτάσεων (abstractive summarization). Η αρχή έγινε με την αναφορά [11], όπου εφαρμόστηκε ένα μοντέλο αρχιτεκτονικής ακολουθία σε ακολουθία (sequence-to-sequence) με μηχανισμό προσοχής [12], το οποίο αποτελείται από έναν κωδικοποιητή συνελικτικών νευρωνικών δικτύων (CNN encoder) και ένα feed-forward νευρωνικό δίκτυο ως αποκωδικοποιητή (decoder). Η αναφορά [13], αντικατέστησε τον αποκωδικοποιητή με ένα δίκτυο αναδρομικών νευρωνικών δικτύων (RNN) βελτιώνοντας την επίδοση του συστήματος. Στη συνέχεια, στην έρευνα [14] πραγματοποιήθηκε η υλοποίηση ενός μοντέλου βασισμένο πλήρως σε αναδρομικά νευρωνικά δίκτυα. Η αναφορά [15] ήταν η πρώτη που απέδειξε ότι ένας μηχανισμός αντιγραφής, μπορεί να συνδυάσει τα πλεονεκτήματα της εξαγωγικής και της αφηρημένης παραγωγής περίληψης, αντιγράφοντας σε μεμονωμένες περιπτώσεις κάποιες λέξεις από το αρχικό κείμενο. Η αναφορά [16] παρατηρώντας ότι το μοντέλο τείνει πολλές φορές να επαναλαμβάνει κάποιες φράσεις, πρόσθεσε ένα μηχανισμό κάλυψης (coverage mechanism) για να αντιμετωπίσει αυτό το φαινόμενο. Πρόσφατα, κάποιες προσεγγίσεις βασισμένες στην ενισχυτική μάθηση (reinforcement learning) [17],[18] δείχνουν υποσχόμενα αποτελέσματα, όμως η εκπαίδευσή τέτοιων συστημάτων είναι πιο αργή και δυσκολότερα ρυθμιζόμενη.

Εύκολα παρατηρεί κανείς ότι η πλειοψηφία των συστημάτων βασίζεται σε μοντέλα αρχιτεκτονικής sequence-to-sequence συνδυασμένα με διάφορους μηχανισμούς και μετατροπές για την βελτιστοποίηση του τελικού συστήματος. Για αυτό λόγω, αντίστοιχη κατεύθυνση ακολούθησε και η παρούσα εργασία, όπως περιγράφεται λεπτομερώς στο Κεφάλαιο 2.

1.4 Αντικείμενο εργασίας

Το αντικείμενο της διπλωματικής εργασίας αποτελεί ο σχεδιασμός και η υλοποίηση ενός συστήματος αυτόματης παραγωγής περίληψης ελληνικών κειμένων με χρήση τεχνικών βαθιάς μάθησης. Συγκεκριμένα, το τελικό σύστημα, παίρνοντας ως είσοδο ένα κείμενο γραμμένο στα ελληνικά, θα παράγει στην έξοδο ένα μικρότερο σε έκταση κείμενο, το οποίο θα διατηρεί το νοηματικό περιεχόμενο του αρχικού και θα έχει συνάφεια. Από τη προηγούμενη ενότητα γίνεται σαφές ότι η συντριπτική πλειοψηφία των συστημάτων που έχει υλοποιηθεί ερευνητικά, αφορούν κείμενα γραμμένα στην αγγλική γλώσσα, με κάποιες εξαιρέσεις όπως οι [19], [20] που αφορούν κινέζικα κείμενα και οι [21], [22] που απευθύνονται στην ινδική γλώσσα. Συνεπώς, στα πλαίσια αυτής της διπλωματικής γίνεται μια πρώτη απόπειρα υλοποίησης ενός αντίστοιχου μοντέλου πάνω σε ελληνικά δεδομένα, διερευνώντας τις ήδη υπάρχουσες υλοποιήσεις σε ξένες γλώσσες.

Θα μπορούσε λοιπόν κανείς να διακρίνει ότι οι κύριες συνεισφορές της παρούσας εργασίας στην έρευνα αποτελούν οι εξής:

- Η επιλογή των κατάλληλων μεθόδων προ-επεξεργασίας και ‘’καθαρισμού’’ των ελληνικών δεδομένων, για την σωστή μορφοποίηση και ενσωμάτωσή τους στο σύστημα.
- Η διερεύνηση των γλωσσικών, λεξιλογικών και συντακτικών ιδιομορφιών της ελληνικής γλώσσας σε σχέση με την αγγλική, καθώς και η διερεύνηση των σχεδιαστικών επιλογών και μεθόδων για την αντιμετώπιση τους.
- Η αξιολόγηση του συστήματος, κατά την οποία γίνεται διερεύνηση των παραμέτρων μέσω πειραμάτων για την μεγιστοποίηση της απόδοσής του. Για την αξιολόγηση του συστήματος χρησιμοποιούνται οι μετρικές Rouge.
- Τα συμπεράσματα που προκύπτουν από την ανάλυση και αξιολόγηση του μοντέλου, τα οποία έχουν ως στόχο την επίδειξη κάποιων σωστών σχεδιαστικών κατευθύνσεων και τη γενικότερη συμπεριφορά του μοντέλου στον αναγνώστη.
- Η επεξήγηση της χρήσης του μοντέλου και η παρουσίαση τρόπων επέκτασής του, όπως η ενσωμάτωσή του σε μια εφαρμογή για έξυπνα κινητά. Και η προώθηση της έρευνας πάνω στο αντικείμενο, και σε αντίστοιχα προβλήματα της επεξεργασίας φυσικής γλώσσας.

Σε αυτό το σημείο πρέπει να τονιστεί πως το υλοποιήσιμο μοντέλο δεν αποτελεί ένα σύστημα παραγωγής περίληψης γενικού σκοπού, αλλά περιορίζεται στην ελληνική γλώσσα και στο συγκεκριμένο λεξιλόγιο και συντακτικό ύφος που προκύπτει από τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευσή του. Η δημιουργία ενός καθολικού συστήματος παραγωγής περίληψης αποτελεί μια πάρα πολύ περίπλοκη και δύσκολη διαδικασία, καθώς ο αριθμός των διαφορετικών φυσικών γλωσσών, διαλέκτων και υφών γραψίματος είναι τόσο μεγάλος, που είναι σχεδόν αδύνατο να μοντελοποιηθεί από ένα μοναδικό σύστημα. Επίσης, ο όγκος των δεδομένων που απαιτείται για την εκπαίδευση ενός τέτοιου συστήματος είναι τεράστιος, όπως αντίστοιχα μεγάλη είναι και η απαιτούμενη υπολογιστική ισχύς. Παρ' όλα αυτά, ο βασικός κορμός και η φιλοσοφία του συστήματος που υλοποιείται από τη παρούσα εργασία, μπορεί να επεκταθεί και να ενσωματωθεί με τις κατάλληλες αλλαγές σε αντίστοιχα συστήματα διαφορετικού σκοπού.

1.5 Οργάνωση κειμένου

Η παρούσα εργασία δομείται από πέντε κεφάλαια , που το κάθε ένα περιγράφει ένα σημαντικό κομμάτι της υλοποίησης της, με κατεύθυνση από τη θεωρία στη πράξη. Συγκεκριμένα, τα επόμενα τέσσερα κεφάλαια που ακολουθούν περιγράφονται ως εξής. Στο κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο στο οποίο στηρίζεται το μοντέλο αυτόματης περίληψης που υλοποιήθηκε. Για να γίνει πλήρως κατανοητό το μοντέλο, η λειτουργία κάθε στοιχείου του περιγράφεται αναλυτικά.

Στο κεφάλαιο 3 γίνεται αναφορά στη μορφή και στη προ-επεξεργασία των ελληνικών δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση, δοκιμή και αξιολόγηση του συστήματος. Η προ-επεξεργασία των δεδομένων είναι ένα σημαντικό στάδιο, για αυτό κάθε βήμα επεξεργασίας που εφαρμόστηκε αναλύεται και αιτιολογείται. Επίσης, στο συγκεκριμένο κεφάλαιο δίνεται μεγάλη προσοχή στις ιδιομορφίες και διαφορές της ελληνικής γλώσσας με την αγγλική και στις μεθόδους αντιμετώπισης τους. Στο τέλος γίνεται μια περιγραφή των αριθμητικών αναπαραστάσεων των λέξεων.

Στη συνέχεια, στο κεφάλαιο 4 παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του λογισμικού. Επίσης, περιγράφονται οι διαδικασίες εκπαίδευσης, αξιολόγησης και δοκιμής του μοντέλου μαζί με τις μετρικές αξιολόγησης ROUGE. Έπειτα, παρουσιάζεται η διερεύνηση διάφορων παραμέτρων του συστήματος, για την εύρεση της βέλτιστης απόδοσής του. Τα βέλτιστα αποτελέσματα που προκύπτουν συγκρίνονται με τα αντίστοιχα άλλων παρόμοιων ερευνητικών εργασιών.

Τέλος, στο κεφάλαιο 5, παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπόνηση αυτής της διπλωματικής εργασίας, καθώς και κάποιες μελλοντικές κατευθύνσεις για τη περαιτέρω έρευνα του αντικειμένου.

2. Θεωρητικό υπόβαθρο

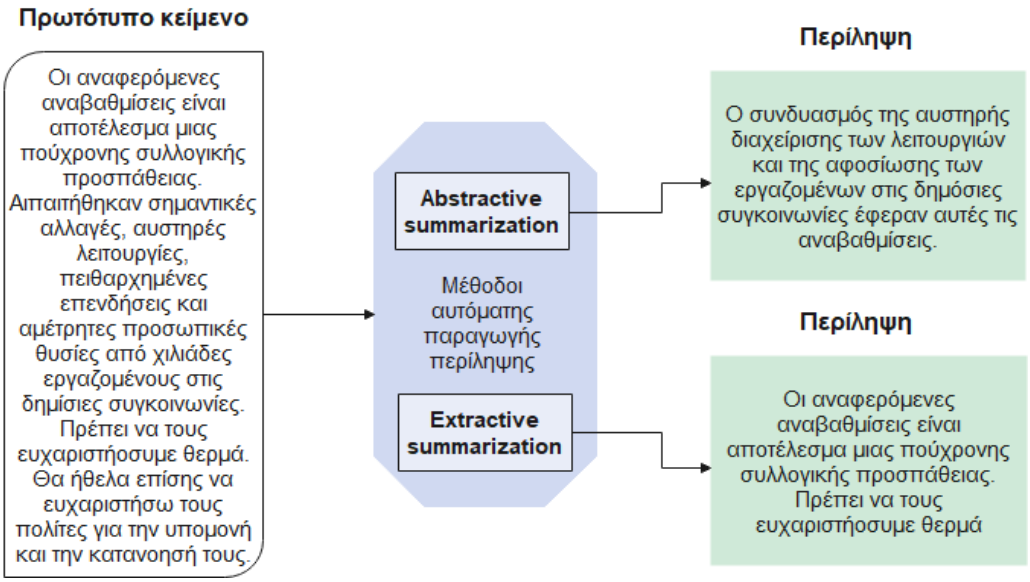
2.1 Τεχνικές αυτόματης παραγωγής περίληψης (extractive – abstractive)

Υπάρχουν δύο βασικές προσεγγίσεις που ακολουθούνται για την αντιμετώπιση του προβλήματος της αυτόματης παραγωγής περίληψης κειμένου. Η πρώτη ονομάζεται εξαγωγική παραγωγή περίληψης (extractive summarization) και όπως προδίδει η ονομασία της, αφορά την εύρεση και εξαγωγή σημαντικών προτάσεων ή φράσεων από το αρχικό κείμενο για τη σύνταξη της περίληψης. Στη πραγματικότητα όμως, ο άνθρωπος όταν πραγματοποιεί τη περίληψη ενός κειμένου τείνει να το παραφράζει με δικό του τρόπο. Στην ίδια μεθοδολογία στηρίζεται η δεύτερη προσέγγιση, γνωστή ως αφηρημένη παραγωγή περίληψης (abstractive summarization), κατά την οποία γίνεται παραγωγή νέων προτάσεων που συντάσσουν περίληψης, διατηρώντας την βασική αρχική πληροφορία. Πιο αναλυτικά, οι δύο βασικές προσεγγίσεις περιγράφονται ως εξής:

- **Extractive Summarization:** Κατά την παραγωγή περίληψης με εξαγωγή προτάσεων, χρησιμοποιούνται αλγόριθμοι μηχανικής μάθησης, οι οποίοι εντοπίζουν και επιλέγουν τις σημαντικότερες προτάσεις στο αρχικό κείμενο, για να συντάξουν τη τελική περίληψη. Η διάκριση των αξιοσημείωτων προτάσεων προς εξαγωγή, πραγματοποιείται βαθμονομώντας τη κάθε λέξη ή πρόταση του αρχικού κειμένου με βάση διάφορα κριτήρια, όπως η συχνότητα εμφάνισης τους. Ένας γνωστός τέτοιος αλγόριθμος αποτελεί ο αλγόριθμος TextRank, ο οποίος βασίζεται στον αλγόριθμο PageRank που χρησιμοποιεί η Google στη μηχανή αναζήτησής της για να βαθμονομήσει τα σχετιζόμενα sites σε κάθε αναζήτηση. Η διαδικασία παρομοιάζει τη συμπεριφορά των ανθρώπων, όπου πολλές φορές όταν διαβάζουμε ένα κείμενο υπογραμμίζουμε τα αξιοσημείωτα σημεία του.
  Αξίζει να σημειωθεί ότι, η διατήρηση ορθής σύνταξης και γραμματικής στη περίληψη που παράγεται από την εξαγωγή προτάσεων είναι δεδομένη, καθώς οι προτάσεις αντιγράφονται αυτούσιες από το κείμενο. Επίσης, η διαδικασία είναι αρκετά γρήγορη καθώς βασίζεται σε μηχανισμούς μη-επιβλεπόμενης μηχανικής μάθησης. Είναι όμως αρκετά περιοριστική όσον αφορά τις δυνατότητες επέκτασής της και δεν παράγει νέα πληροφορία.

- **Abstractive Summarization**: Κατά την παραγωγή περίληψης με παραγωγή νέων προτάσεων, χρησιμοποιούνται αλγόριθμοι μηχανικής μάθησης οι οποίοι αναλύοντας τις ακολουθίες λέξεων που προκύπτει από το κείμενο, βρίσκουν σχετιζόμενες νοηματικά λέξεις και φράσεις με τις οποίες συντάσσουν τη περίληψη. Σε αυτή τη περίπτωση για να παραχθεί ένα αξιόλογο αποτέλεσμα, ο υπολογιστής χρειάζεται μεγάλο όγκο δεδομένα και αρκετό χρόνο εκπαίδευσης. Όπως αναφέρθηκε και προηγουμένως, αρκετή επιτυχία σε αυτή τη προσέγγιση έχουν επιδείξει μηχανισμοί βαθιάς μάθησης, όπως τα αναδρομικά νευρωνικά δίκτυα. Η αφηρημένη παραγωγή περίληψης προσομοιάζει

αρκετά τη καθιερωμένη διαδικασία που ακολουθούν οι άνθρωποι για την παραγωγή περίληψης, όπου αναλύοντας το κείμενο παράγουν νέες προτάσεις βασιζόμενοι σε αυτό. Σε αντίθεση με την προηγούμενη περίπτωση, η σωστή γραμματική και σύνταξη δεν είναι δεδομένες και αποτελούν ένα πρόβλημα προς επίλυση. Επιπλέον, αν και οι μηχανισμοί που χρησιμοποιούνται στη συγκεκριμένη διαδικασία είναι πιο περίπλοκοι και παράγουν χειρότερα αποτελέσματα από την εξαγωγή προτάσεων, είναι πιο υποσχόμενοι για την δημιουργία ενός ιδανικού και καθολικού συστήματος αυτόματης περίληψης, καθώς προσφέρουν μεγάλη αφαιρετικότητα.



Εικόνα 2.1: *Παράδειγμα extractive vs abstractive αυτόματης παραγωγής περίληψης.*

Σημειώνεται ότι όλα τα συστήματα αυτόματης παραγωγής περίληψης που έχουν υλοποιηθεί ακολουθούν μία από τις δύο παραπάνω προσεγγίσεις, ή τον συνδυασμό τους [23]. Στη παρούσα εργασία, υλοποιείται σύστημα αυτόματης παραγωγής περίληψης με παραγωγή νέων προτάσεων (abstractive summarization), καθώς η προσέγγιση αυτή είναι πιο υποσχόμενη και λιγότερο περιορίσιμη. Περιέχει βέβαια, μικρή επιρροή από τη μέθοδο της εξαγωγικής περίληψης (extractive summarization), μέσω του μηχανισμού αντιγραφής άγνωστων λέξεων που θα αναλυθεί παρακάτω. Στη συνέχεια, πραγματοποιείται μια αναφορά στη βαθιά μάθηση και στη συνεισφορά της στην αντιμετώπιση του προβλήματος της αυτόματης παραγωγής περίληψης.

2.2 Βαθιά μάθηση και επεξεργασία φυσικής γλώσσας

Όπως ήδη αναφέρθηκε η αυτόματη παραγωγή περίληψης αποτελεί μια περίπλοκη και μη ντετερμινιστική διαδικασία που εξαρτάται από πολυάριθμες παραμέτρους. Η υλοποίηση λοιπόν ενός αποτελεσματικού συστήματος αυτόματης παραγωγής περίληψης δε μπορεί να βασιστεί σε απλούς υπολογιστικούς μηχανισμούς, αλλά σε πιο εξελιγμένους μηχανισμούς που προσομοιάζουν την ανθρώπινη λογική, όπως είναι αυτοί της τεχνητής νοημοσύνης και συγκεκριμένα της βαθιάς μάθησης.

Η τεχνητή νοημοσύνη είναι ένας ευρύς και αναπτυσσόμενος κλάδος της επιστήμης των υπολογιστών, ο οποίος ασχολείται με την ανάπτυξη εξελιγμένων υπολογιστικών συστημάτων, ικανών να πραγματοποιήσουν διεργασίες που απαιτούν ανθρώπινη νοημοσύνη. Μερικοί τομείς στους οποίους εφαρμόζεται η τεχνητή νοημοσύνη είναι η ρομποτική, η όραση υπολογιστών, η επεξεργασία φυσικής γλώσσας, τα μέσα κοινωνικής δικτύωσης, η κυβερνοασφάλεια και πολλά άλλα. Ένας σημαντικός κλάδος της τεχνητής νοημοσύνης είναι η μηχανική μάθηση. Η μηχανική μάθηση μελετά αλγορίθμους και μηχανισμούς για την υλοποίηση αυτόνομων μοντέλων. Τα μοντέλα αυτά, αφού εκπαιδευτούν από ένα σύνολο δεδομένων, μπορούν να προβλέψουν και να πάρουν αποφάσεις, αυτοματοποιώντας την εκάστοστε λειτουργία.

Η βαθιά μάθηση αποτελεί ένα υποσύνολο της μηχανικής μάθησης, και χρησιμοποιεί πολυσύνθετα νευρωνικά δίκτυα, ικανά να εκπαιδευτούν από ένα τεράστιο όγκο μη κατηγοριοποιημένων δεδομένων και να πάρουν αποφάσεις με την ελάχιστη ανθρώπινη παρέμβαση. Η σημαντικότερη διαφορά των μηχανισμών της βαθιάς μάθησης με των υπόλοιπων μηχανισμών μηχανικής μάθησης είναι η επεκτασιμότητα. Στη περίπτωση των βαθιών νευρωνικών δικτύων, λόγω του μεγέθους και της πολυπλοκότητας τους, έχουν τη δυνατότητα να δεχτούν ένα τεράστιο όγκο μη κατηγοριοποιημένων δεδομένων εκπαίδευσης, αυξάνοντας αναλόγως την επίδοση του τελικού μοντέλου. Αντιθέτως, απλούστερες υλοποιήσεις νευρωνικών δικτύων μηχανικής μάθησης δεν είναι ικανά να προσπελάσουν τόσο μεγάλο όγκο δεδομένων και η επίδοσή τους έχει συνήθως κάποιο ‘’ταβάνι’’. Συνεπώς, οι μηχανισμοί της βαθιάς μάθησης χρησιμοποιούνται σε προβλήματα με μεγάλη πολυπλοκότητα και απαιτούν μεγάλο όγκο δεδομένων για την εκπαίδευσή τους. Τέτοια προβλήματα είναι αυτά της επεξεργασίας φυσικής γλώσσας, και συγκεκριμένα ένα από αυτά είναι η αυτόματη παραγωγή περίληψης με την οποία ασχολείται η παρούσα εργασία.

Εικόνα 2.2: *Τεχνητή νοημοσύνη/Μηχανική μάθηση/Βαθιά μάθηση*

Η ραγδαία ανάπτυξη της βαθιάς μάθησης και των μηχανισμών της, ώθησε την υλοποίηση διάφορων ερευνών πάνω σε προβλήματα ακολουθιακής παραγωγής πληροφορίας, όπως είναι αυτά της επεξεργασίας φυσικής γλώσσας, υποδεικνύοντας πολύ καλά αποτελέσματα. Προβλήματα όπως η αυτόματη περιγραφή εικόνας με κείμενο [24]**,** η αυτόματη παραγωγή κειμένου από φωνή [12] και η αυτόματη μετάφραση κειμένου [25]**,** αντιμετωπίστηκαν για πρώτη φορά σε έναν ικανοποιητικό βαθμό. Τα μοντέλα που χρησιμοποιήθηκαν για την επίλυση των παραπάνω προβλημάτων, επιλέχθηκαν ως βάση για την αντιμετώπιση του προβλήματος της αυτόματης παραγωγής περίληψης. Συγκεκριμένα, το πρόβλημα της αυτόματης μετάφρασης είναι το πιο κοντινό στην αυτόματη παραγωγή περίληψης, καθώς και στις δύο περιπτώσεις πραγματοποιείται η μετατροπή μιας μορφής κειμένου σε μία άλλη.

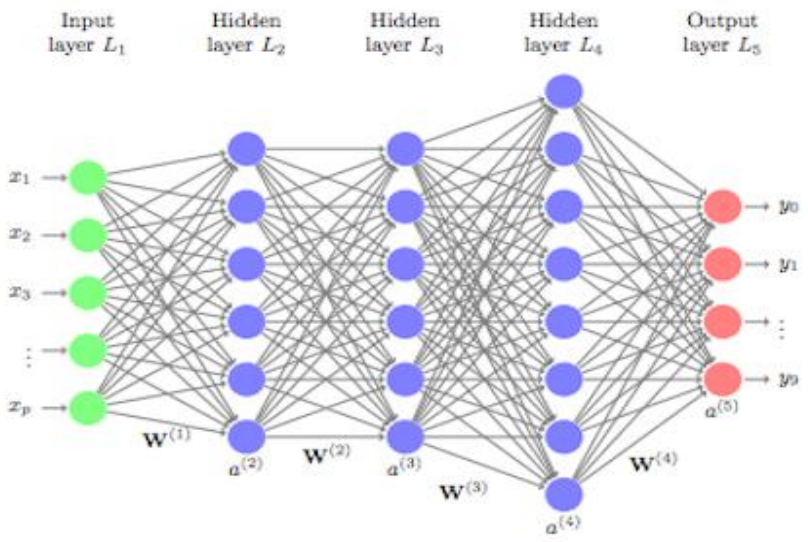
Όλα τα παραδείγματα που αναφέρθηκαν παραπάνω, βασίζονται σε μια συγκεκριμένη αρχιτεκτονική βαθιάς μάθησης που ονομάζεται ακολουθία σε ακολουθία (seq2seq) και αποτελείται από έναν κωδικοποιητή και ένα αποκωδικοποιητή. Στη συγκεκριμένη αρχιτεκτονική στηρίζεται και το μοντέλο της παρούσας εργασίας, για αυτό το λόγο θα αναλυθεί λεπτομερώς στη παρακάτω ενότητα.

2.3 Περιγραφή μοντέλου

2.3.1 Αναδρομικά νευρωνικά δίκτυα (RNNs)

Τα αναδρομικά νευρωνικά δίκτυα (recurrent neural networkς) αποτελούν έναν από τους κυριότερους μηχανισμούς της βαθιάς μάθησης, και το βασικό δομικό στοιχείου της αρχιτεκτονικής του μοντέλου που περιγράφεται. Ένα αναδρομικό νευρωνικό δίκτυο μοιάζει αρκετά με το κλασσικό εμπρόσθιο νευρωνικό δίκτυο (feedforward neural network).

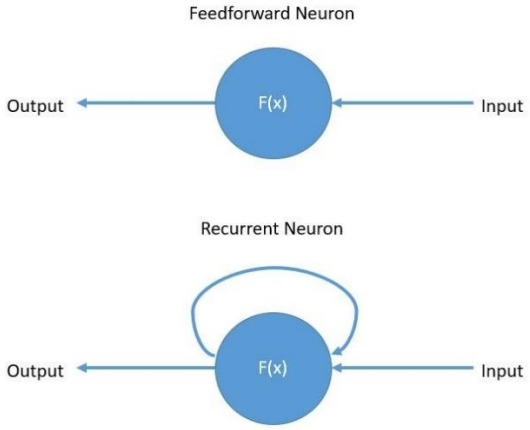
Ένα κλασσικό νευρωνικό δίκτυο αποτελείται από τεχνητούς νευρώνες, οι οποίοι συνδέονται μεταξύ τους, και χωρίζεται σε τρία τμήματα: το επίπεδο εισόδου (input layer), τα κρυφά επίπεδα (hidden layers) και το επίπεδο εξόδου (output layer). Μέσω των τεχνητών νευρώνων η πληροφορία ρέει από το επίπεδο εισόδου προς το επίπεδο εξόδου, αλλάζοντας τιμές στα ενδιάμεσα στάδια. Πιο συγκεκριμένα, στο νευρωνικό δίκτυο υπάρχουν συγκεκριμένες τιμές που ονομάζονται βάρη, βάσει των οποίων μεταδίδεται η πληροφορία. Αυτά τα βάρη είναι τιμές που προκύπτουν από την συσχέτιση μεταξύ των δεδομένων, και προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης, ώστε στο τέλος να προκύψουν οι επιθυμητοί έξοδοι για κάθε είσοδο. Όσο καλύτερα εκπαιδευμένο είναι το δίκτυο, τόσο πιο ακριβείς είναι τα υπολογισμένα βάρη και οι προβλέψεις του πάνω σε νέα άγνωστα δεδομένα.



Εικόνα 2.3*: Απεικόνιση ενός κλασσικού νευρωνικού δικτύου με 3 κρυφά επίπεδα.*

Σε ένα κλασσικό νευρωνικό δίκτυο, υπάρχει μια έξοδο ανά είσοδο και συνεπώς δεν είναι δυνατό εισάγοντας στο δίκτυο το ένα δεδομένο μετά το άλλο, να παραχθεί μια έξοδο με βάση όλα αυτά τα δεδομένα, αλλά θα παραχθούν ξεχωριστές εξόδους για κάθε δεδομένο. Συνεπώς, ένα κλασσικό νευρωνικό δίκτυο δε μπορεί να επεξεργαστεί διαδοχικά δεδομένα, που ουσιαστικά συγκροτούν μια ακολουθία δεδομένων.

Αυτό το πρόβλημα αντιμετωπίζεται με τα αναδρομικά νευρωνικά δίκτυα. Η κύρια διαφορά του αναδρομικού νευρωνικού δικτύου με το κλασσικό νευρωνικό δίκτυο, είναι ότι η έξοδος του δικτύου ανατροφοδοτείται στην είσοδό του, επιτρέποντας κάθε έξοδός του να αποτελεί συνάρτηση της τωρινής αλλά και των προηγούμενων εισόδων του. Η πληροφορία που ανατροφοδοτείται ονομάζεται κρυφή κατάσταση (hidden state). Μια πιο μαθηματική περιγραφή είναι η εξής : Έστω ότι έχουμε μια ακολουθία N εισόδων $X = \{x_i, i=0,..N\}$, μια έξοδο y_i και μια κρυφή κατάσταση h_i που αντιστοιχούν σε είσοδο x_i σε μια δεδομένη χρονική στιγμή. Στον επόμενο γύρο εκπαίδευσης θα τροφοδοτηθούν στο δίκτυο τόσο η επόμενη είσοδο της ακολουθίας x_{i+1}, όσο και η κρυφή κατάσταση h_i, με αποτέλεσμα η επόμενη έξοδος να υπολογίζεται συναρτήσει αυτών και να περιέχει πληροφορία σχετικά με την προηγούμενη είσοδο. Η κρυφή κατάσταση λειτουργεί σαν ένα είδος εσωτερικής μνήμης, καθώς απομνημονεύει πληροφορίες προηγούμενων δεδομένων.



Εικόνα 2.4: *Διαφορές κλασσικού και αναδρομικού νευρωνικού νευρώνα*

Με αυτή την ιδιότητα της ανατροφοδότησης, το αναδρομικό νευρωνικό δίκτυο είναι ικανό να χειριστεί πολύ αποτελεσματικά εισόδους που αποτελούν εξαρτημένες ακολουθίες, δηλαδή ακολουθίες στις οποίες κάθε είσοδος της ακολουθίας εξαρτάται από τις προηγούμενες εισόδους. Η ιδιότητα αυτή είναι ο κύριος λόγος για τον οποίο τα αναδρομικά νευρωνικά δίκτυα λειτουργούν αποτελεσματικά σε προβλήματα επεξεργασίας φυσικής γλώσσας, καθώς κάθε πρόταση φυσικής γλώσσας αποτελείται από διαδοχικές λέξεις, οι οποίες εξαρτώνται η μια από την άλλη. Η κάθε λέξη σε μία πρόταση καθορίζει ποιες λέξεις μπορούν να την ακολουθήσουν, τόσο από άποψη γραμματικής και συντακτικού, όσο και νοηματικού περιεχομένου. Έτσι το αναδρομικό δίκτυο, έχοντας την ικανότητα να παράγει μια ακολουθία πληροφορίας, μπορεί να λειτουργήσει αποτελεσματικά στο συγκεκριμένο πρόβλημα της αυτόματης παραγωγής περίληψης. Κάθε λέξη της περίληψης που εξάγεται από το δίκτυο παράγεται συνυπολογίζοντας τις ήδη παραγόμενες λέξεις της περίληψης, επιτρέποντας έτσι τη λήψη σωστότερων

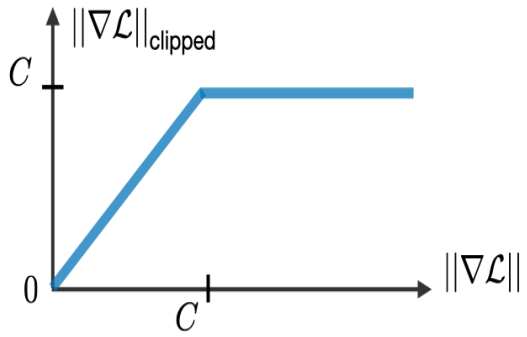
αποφάσεων από το δίκτυο, όσον αφορά τη γραμματική, τη σύνταξη και το νόημα των τελικών προτάσεων.

Αν και το αναδρομικό νευρωνικό δίκτυο είναι μια μορφή νευρωνικού δικτύου που μπορεί να χειριστεί ακολουθίες δεδομένων, πάσχει αρκετές φορές από προβλήματα, τα οποία προκύπτουν από το πολλαπλασιαστικό τρόπο διάδοσης του σφάλματος της εξόδου κατά τη διάρκεια της εκπαίδευσης. Συγκεκριμένα, το σφάλμα της εξόδου του δικτύου υπολογίζεται βάσει της συνάρτησης κόστους, η οποία μετράει την απόκλιση της τιμής της εξόδου του δικτύου από την αναμενόμενη τιμή. Κατά τη διάδοση του σφάλματος, υπολογίζονται παράγωγοι της συνάρτησης κόστους, οι οποίες χρησιμοποιούνται για την ενημέρωση των βαρών του δικτύου σε κάθε βήμα εκπαίδευσης. Σύμφωνα με το κανόνα της αλυσίδας, όσο το σφάλμα διαδίδεται προς τα πίσω (backpropagation), τόσο αυξάνονται οι πολλαπλασιαστικοί όροι από τους οποίους υπολογίζονται αυτές οι παράγωγοι. Αυτή η δραματική αύξηση των πολλαπλασιαστικών όρων μπορεί να οδηγήσει σε πολύ μικρούς αριθμούς ενημέρωσης των αρχικών βαρών, αν πολλοί από αυτούς τους όρους προκύψουν μικροί (vanishing gradient) ή σε πολύ μεγάλους αν οι όροι προκύψουν μεγάλοι (exploding gradient). Η εξίσωση **2.1** περιγράφει τον υπολογισμό της παραγώγου για την ενημέρωση τυχαίου βάρους w1 σε ένα νευρωνικό δίκτυο με τρία κρυφά επίπεδα.

$$\frac{\partial error}{\partial w1} = \frac{\partial error}{\partial output} * \frac{\partial output}{\partial hidden3} * \frac{\partial hidden3}{\partial hidden2} * \frac{\partial hidden2}{\partial hidden1} * \frac{\partial hidden1}{\partial w1} \quad (2.1)$$

Η αντιμετώπιση των παραπάνω προβλημάτων σε περίπτωση εμφάνισης τους είναι σημαντική καθώς μειώνουν την επίδοση και την ακρίβεια του μοντέλου και αυξάνουν τον απαιτούμενο χρόνο εκπαίδευσης.

Στη περίπτωση εμφάνισης πολύ μεγάλων τιμών των βαρών , χρησιμοποιείται συχνά η μέθοδος αποκοπής (gradient clipping) για την αντιμετώπιση της. Με αυτό το τρόπο τίθεται ένα κατώφλι το οποίο αποκόβει τις τιμές των παραγώγων όταν αυτές το υπερβαίνουν, εμποδίζοντας την συσσώρευση μεγάλων τιμών.



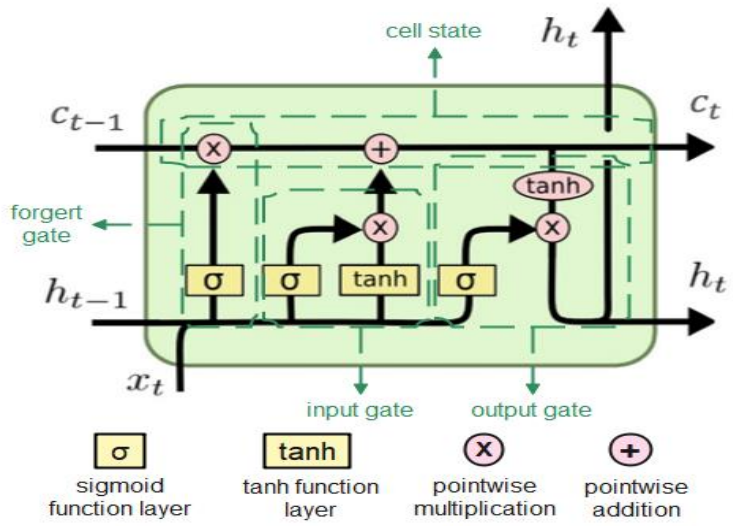
Εικόνα 2.5: *Μέθοδος αποκοπής*

Το αντίστοιχο φαινόμενο για πολύ μικρές τιμές, εμφανίζεται συχνότερα στα αναδρομικά νευρωνικά δίκτυα και προκαλεί το πρόβλημα της βραχυπρόθεσμης μνήμης (short term memory). Συγκεκριμένα, όσο αυξάνεται το μήκος της ακολουθίας εισόδου στο δίκτυο, τόσο αυξάνονται οι πολλαπλασιαστικοί όροι των παραγώγων και αντίστοιχα μικραίνουν οι τιμές τους. Ως αποτέλεσμα το δίκτυο αδυνατεί να συγκρατήσει πληροφορία από όλο το μήκος της ακολουθίας και "μνήμη" του γίνεται βραχυπρόθεσμη.

Συνεπώς, αρκετή πληροφορία που προκύπτει από την αρχή της ακολουθίας χάνεται και άρα μια απόφαση/πρόβλεψη του δικτύου, θα πραγματοποιηθεί λαμβάνοντας υπόψιν περισσότερο τις τελευταίες και κοντινές εισόδους και λιγότερο έως καθόλου τις αρχικές. Στη περίπτωση της αυτόματης παραγωγής περίληψης, οι πιο πρόσφατες λέξεις παίζουν σημαντικότερο ρόλο στον υπολογισμό της εξόδου από άποψη γραμματικής και σύνταξης. Όμως πρέπει να ληφθούν υπόψιν και οι παλαιότερες, καθώς είναι εξίσου σημαντικές για την συνάφεια και το συνολικό νόημα του τελικού κειμένου.

Το πρόβλημα της βραχυπρόθεσμης μνήμης αντιμετωπίζεται με τη χρήση κάποιων παραλλαγών του αναδρομικού νευρωνικού δικτύου, τα οποία σχεδιάστηκαν να είναι αποτελεσματικά ανεξαρτήτως του μεγέθους της ακολουθίας εισόδου. Τα πιο διαδεδομένα από αυτά είναι το δίκτυο LSTM (Long Short Term Memory) και το δίκτυο GRU (Gated Recurrent Unit), τα οποία αναλύονται παρακάτω.

Long Short Term (LSTM) δίκτυο: Το κύριο χαρακτηριστικό που διαφοροποιεί το LSTM δίκτυο από το κλασσικό αναδρομικό και το κάνει ικανό να χειριστεί μεγάλες ακολουθίες πληροφορίας, είναι η κατάσταση κυττάρου (cell state). Η κατάσταση κυττάρου αποτελεί ένα είδος μνήμης για το δίκτυο, καθώς σε κάθε βήμα εκπαίδευσης ανατροφοδοτείται στην είσοδό του δικτύου, αποθηκεύοντας και μεταφέροντας σχετικές πληροφορίες σε όλο το μήκος της ακολουθίας. Με αυτό το τρόπο, ακόμα και πληροφορίες που προκύπτουν από τα αρχικά βήματα, δηλαδή την αρχή της ακολουθίας εισόδου, μπορεί να φτάσουν μέχρι το τέλος, περιορίζοντας σε μεγάλο βαθμό το φαινόμενο της βραχυπρόθεσμης μνήμης. Καθώς η κατάσταση κυττάρου μεταδίδεται, σχετικές πληροφορίες προστίθενται ή αφαιρούνται από αυτήν σε κάθε βήμα μέσω ειδικών πυλών του δικτύου. Οι πύλες αυτές είναι μηχανισμοί που μαθαίνουν, μέσω της εκπαίδευσης του δικτύου, να διακρίνουν ποια πληροφορία είναι χρήσιμη και πρέπει να αποθηκευτεί και ποια είναι άχρηστη και πρέπει να "ξεχαστεί".



Εικόνα 2.6: *Αρχιτεκτονική μονάδας LSTM δικτύου.*

Όπως απεικονίζεται στην **Εικόνα 2.6**, κάθε μονάδα του LSTM δικτύου περιέχει τρείς εισόδους και δύο εξόδους. Όπου x_t είναι η είσοδος της ακολουθίας για τη τρέχων μονάδα του δικτύου, c_{t-1} είναι η "μνήμη" ή αλλιώς η κατάσταση κυττάρου από τη προηγούμενη μονάδα, h_{t-1} είναι η έξοδος της προηγούμενης μονάδας, c_t είναι η τρέχουσα αποθηκευμένη πληροφορία και h_t είναι η παραγόμενη έξοδος. Επίσης, χρησιμοποιούνται δυο συναρτήσεις ενεργοποίησης, η σιγμοειδή συνάρτηση (sigmoid) που προσαρμόζει τις τιμές μέσα στα όρια [0,1], και η συνάρτηση υπερβολικής εφαπτομένης που προσαρμόζει τις τιμές μέσα στα όρια [-1,1]. Το LSTM περιέχει τέσσερις βασικές δομές: τη κατάσταση κυττάρου και τρεις πύλες. Οι λειτουργίες τους περιγράφονται παρακάτω.

- **Πύλη διαγραφής (Forget gate)** : Η πύλη διαγραφής αποφασίζει ποια πληροφορία είναι σημαντική και πρέπει να αποθηκευτεί και ποια να διαγραφεί. Παίρνει ως εισόδους την έξοδο της προηγούμενης μονάδας h_{t-1} και την είσοδο της τρέχουσας μονάδας x_t, Αυτές αφού πρώτα ενωθούν, περνάνε από τη σιγμοειδή συνάρτηση, που φράζει τις τιμές τους στο διάστημα από 0 έως 1 και ουσιαστικά καθορίζει τη σημαντικότητα τους. Στη συνέχεια, η έξοδος που προκύπτει από τη συνάρτηση πολλαπλασιάζεται ανά στοιχείο με την αποθηκευμένη πληροφορία, που αποτελεί την προηγούμενη κατάσταση κυττάρου. Μέσω αυτού του πολλαπλασιασμού, όση πληροφορία προσεγγίζει το 0 διαγράφεται, ενώ όση είναι κοντά στο 1 αποθηκεύεται. Η λειτουργία της πύλης διαγραφής περιγράφεται από τη συνάρτηση **2.2.**

- **Πύλη εισόδου (Input gate)** : Η πύλη εισόδου είναι υπεύθυνη στο να αποφασίσει ποια πληροφορίά θα προστεθεί στην κατάσταση κυττάρου, που έχει προκύψει μετά την πύλη διαγραφής. Όπως και στην πύλη διαγραφής, η έξοδο της προηγούμενης μονάδας h_{t-1} και η τωρινή είσοδος x_t ενώνονται και περνάνε από τη σιγμοειδή συνάρτηση, η οποία φράζει

τις τιμές τους στο διάστημα από 0 έως 1. Ταυτόχρονα όμως, περνάνε και από τη συνάρτηση υπερβολικής εφαπτομένης, η οποία με τη σειρά της περιορίζει τις τιμές τους στο διάστημα από -1 έως 1. Η πρώτη συνάρτηση προσδιορίζει ποιες τιμές θα προστεθούν στη "μνήμη" και η δεύτερη λειτουργεί σαν ένα μέτρο κανονικοποίησης των τιμών. Τα αποτελέσματα που προκύπτουν από αυτές τις δύο συναρτήσεις πολλαπλασιάζονται ανά σημείο μεταξύ τους για να προκύψει η πληροφορία που θα προστεθεί. Η λειτουργία της πύλης εισόδου εκφράζεται μαθηματικά με την συνάρτηση **2.3**.

- **Κατάσταση κυττάρου (Cell state)** : Η κατάσταση κυττάρου αποτελείται από την έξοδο της πύλης διαγραφής συν την έξοδο της πύλης εισόδου. Η κατάσταση κυττάρου περιέχει όλη τη σχετική και σημαντική πληροφορία και όπως ήδη αναφέρθηκε, λειτουργεί ως μνήμη της μονάδας του δικτύου. Προκύπτει από το άθροισμα της την εξόδου της πύλης διαγραφής με την έξοδο της πύλης εισόδου, οπώς περιγράφεται και στην εξίσωση **2.4**.
- **Πύλη εξόδου (Output gate)** : Μέσω της πύλης εξόδου προσδιορίζεται η πληροφορία που θα βγεί σαν έξοδο από την τρέχουσα μονάδα του δικτύου. Η προηγούμενη έξοδος ενώνεται με την τωρινή είσοδο, όπως και στις υπόλοιπες πύλες, και περνάνε μαζί από τη συνάρτηση υπερβολικής συνάρτησης, που κανονικοποιεί τις τιμές τους στο διάστημα [-1,1]. Το αποτέλεσμα πολλαπλασιάζεται ανά σημείο με την κατάσταση κυττάρου, η οποία έχει προηγουμένως περάσει από τη σιγμοειδή συνάρτηση. Το αποτέλεσμα της συνάρτησηςαποτελεί την έξοδο της μονάδας. Η λειτουργία της πύλης εισόδου εκφράζεται μαθηματικά με τις συναρτήσεις **2.5** και **2.6**.

Η λειτουργία των πυλών και συνεπώς του LSTM δικτύου περιγράφεται μαθηματικά με τις παρακάτω συναρτήσεις:

$$f_t = \sigma(W_f * x_t + U_f * h_{t-1} + b_f) \quad (2.2)$$

$$i_t = \sigma(W_i * x_t + U_i * h_{t-1} + b_i) \quad (2.3)$$

$$c_t = f_t * c_{t-1} + i_t * tanh(W_c * x_t + U_c * h_{t-1} + b_c) \quad (2.4)$$

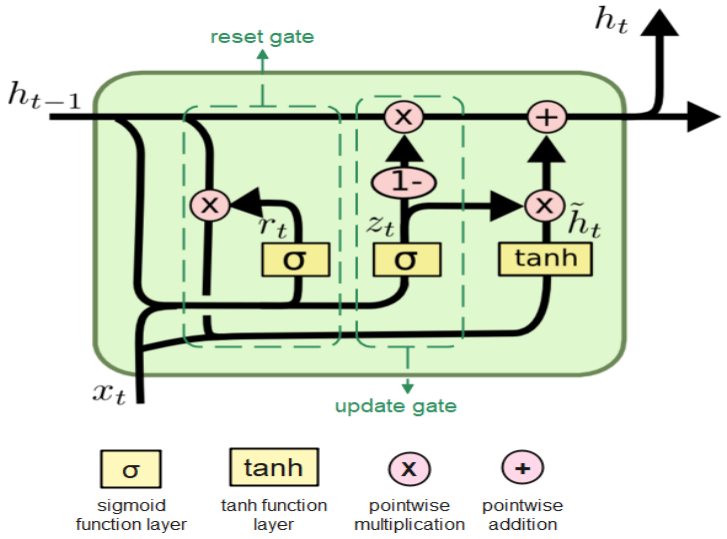
$$o_t = \sigma(W_o * x_t + U_o * h_{t-1} + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh(c_t) \quad (2.6)$$

Όπου σ είναι η σιγμοειδής συνάρτηση, tanh η συνάρτηση παραβολικής εφαπτομένης, W και U είναι τα βάρη που προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης και b κάποιες σταθερές τιμές.

Gated Recurrent Unit (GRU) δίκτυο: Το GRU δίκτυο είναι μια ακόμη γνωστή παραλλαγή του αναδρομικού νευρωνικού δικτύου που αντιμετωπίζει αποτελεσματικά το πρόβλημα της βραχυπρόθεσμης μνήμης. Η αρχιτεκτονική του είναι παρόμοια με αυτή του LSTM αλλά λίγο πιο απλουστευμένη, καθώς περιέχει λιγότερους όρους. Συγκεκριμένα, το GRU δίκτυο έχει δυο εισόδους, τη x_t που είναι η είσοδος της ακολουθίας για τη τρέχων μονάδα του δικτύου και τη

h_{t-1} που είναι η έξοδος της προηγούμενης μονάδας. Υπάρχει όμως μόνο μια έξοδος η h_t , η οποία ανατροφοδοτείται στην επόμενη μονάδα του δικτύου σε κάθε κύκλο αναδρομής, σε αντίθεση με το LSTM που έχει δύο εξόδους. Η μοναδική έξοδος του δικτύου h_t , ονομάζεται κρυφή κατάσταση (hidden state) και έχει αντίστοιχη λειτουργία με την κατάσταση κυττάρου στο LSTM, καθώς μέσω αυτής αποθηκεύεται και μεταδίδεται η σχετική πληροφορία.



Εικόνα 2.7: *Αρχιτεκτονική μονάδας GRU δικτύου*

Το GRU δίκτυο περιέχει 2 πύλες, τη πύλη επαναφοράς και τη πύλη ενημέρωσης και χρησιμοποιεί επίσης τη σιγμοειδή συνάρτηση και τη συνάρτηση παραβολικής εφαπτομένης για να επιτελέσει τις λειτουργίες της. Όπως και στο LSTM δίκτυο, αυτές οι πύλες εκπαιδεύονται να φιλτράρουν τη πληροφορία σε κάθε κύκλο αναδρομής, κρατώντας μόνο ότι είναι χρήσιμο, Η λειτουργία κάθε πύλης περιγράφεται παρακάτω:

- **Πύλη επαναφοράς (reset gate)** : Μέσω της πύλης επαναφοράς υπολογίζεται πόση και ποια αποθηκευμένη πληροφορία από τις προηγούμενες μονάδες του δικτύου θα διαγραφεί. Η πύλη παίρνει ως εισόδους την έξοδο της προηγούμενης μονάδας h_{t-1} και την είσοδο της τρέχουσας μονάδας $x_{t,}$, οι οποίες αφού πρώτα ενωθούν, περνάνε από τη σιγμοειδή συνάρτηση, η οποία προσαρμόζει τις τιμές τους στο διάστημα από 0 έως 1. Έπειτα το αποτέλεσμα που προκύπτει από τη σιγμοειδή πολλαπλασιάζεται ανά σημείο με την προηγούμενη κρυφή κατάσταση h_{t-1} και προκύπτει η έξοδος της πύλης. Μέσω αυτού του πολλαπλασιασμού, όσες τιμές θεωρούνται ασήμαντες και είναι κοντά στο 0 χάνονται, ενώ αυτές που είναι κοντά στο 1 συνεχίζουν καθώς είναι χρήσιμες. Η λειτουργία της πύλης εισόδου εκφράζεται μαθηματικά με την συνάρτηση **2.7**.

- **Πύλη ενημέρωσης (update gate)** : Η πύλη ενημέρωσης βοηθά το δίκτυο να καθορίσει πόση και ποια αποθηκευμένη πληροφορία από το παρελθόν είναι χρήσιμη και πρέπει να μεταδοθεί και στις επόμενες χρονικά μονάδες. Όπως και στη πύλη επαναφοράς, η έξοδος της προηγούμενης μονάδας h_{t-1} και η είσοδος της τρέχουσας μονάδας x_t ενώνονται και περνάνε μαζί από τη σιγμοειδή συνάρτηση, η οποία φράζει τις τιμές τους στο διάστημα [0,1]. Στη συνέχεια όμως, το αποτέλεσμα της συνάρτησης δεν πολλαπλασιάζεται αυτούσιο με την προηγούμενη κρυφή κατάσταση h_{t-1} όπως στη πύλη επαναφοράς, αλλά πρώτα οι τιμές του αφαιρούνται από μοναδιαίες τιμές. Έπειτα, το αποτέλεσμα αυτού του πολλαπλασιασμού προστίθενται με την έξοδο της πύλης επαναφοράς αφού αυτή περάσει πρώτα από τη συνάρτηση tanh, η οποία κανονικοποιεί τις τιμές της στο διάστημα [-1,1]. Το άθροισμα αυτό αποτελεί τη κρυφή κατάσταση της τρέχουσας μονάδας και συνεπώς την έξοδό της, καθώς είναι η πληροφορία που θα τροφοδοτηθεί στα επόμενα βήματα. Η λειτουργία της πύλης εισόδου εκφράζεται μαθηματικά με την συνάρτηση **2.8**.

Η λειτουργία των πυλών και συνεπώς του GRU δικτύου περιγράφεται μαθηματικά με τις παρακάτω συναρτήσεις:

$$r_t = \sigma(W_r * x_t + U_r * h_{t-1} + b_r) \quad (2.7)$$

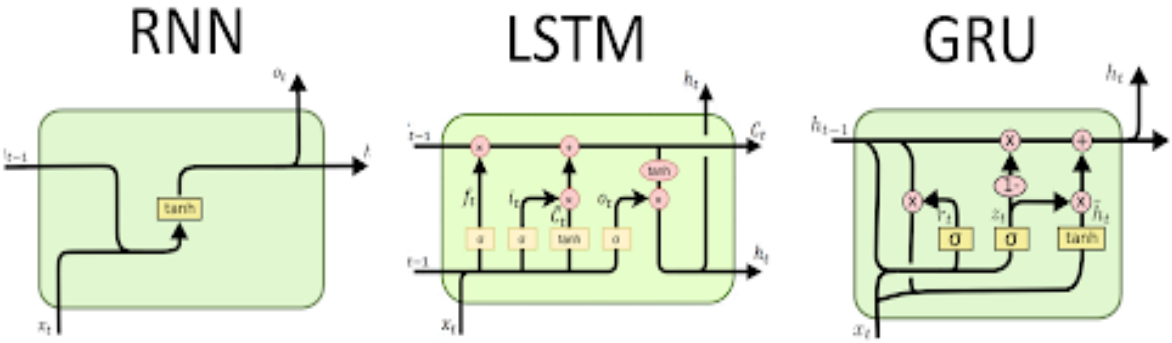
$$z_t = \sigma(W_z * x_t + U_z * h_{t-1} + b_z) \quad (2.8)$$

$$\tilde{h}_t = tanh(W_h * x_t + U_h * (r_t * h_{t-1}) + b_h) \quad (2.9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2.10)$$

Όπου σ είναι η σιγμοειδής συνάρτηση, tanh η συνάρτηση παραβολικής εφαπτομένης, W και U είναι τα βάρη που προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης και b κάποιες σταθερές τιμές.

Οι αρχιτεκτονικές των LSTM και GRU δικτύων είναι έτσι σχεδιασμένες, ώστε να μπορούν να αντιμετωπίσουν το πρόβλημα της βραχυπρόθεσμης μνήμης, καθώς κατά τη διάδοση της πληροφορίας χρησιμοποιούν μηχανισμούς που διακρίνουν το σημαντικό κομμάτι της και το "θυμούνται"΄. Συγκρίνοντας κάποιος τις δύο αρχιτεκτονικές μεταξύ τους, συμπεραίνει ότι το GRU δίκτυο, έχοντας μικρότερη πολυπλοκότητα και λιγότερα στοιχεία, είναι ευκολότερα διαχειρίσιμο και εκπαιδεύεται γρηγορότερα. Από την άλλη όμως, το LSTM δίκτυο παρουσιάζει καλύτερα αποτελέσματα όταν υπάρχουν πολλά διαθέσιμα δεδομένα εκπαίδευση και επαρκή υπολογιστική ισχύς, ιδιαίτερα σε περίπλοκα προβλήματα όπως αυτό της αυτόματης παραγωγής περίληψης. Για αυτό το λόγο, στο παρών μοντέλο προτιμάται η χρήση του LSTM δικτύου.



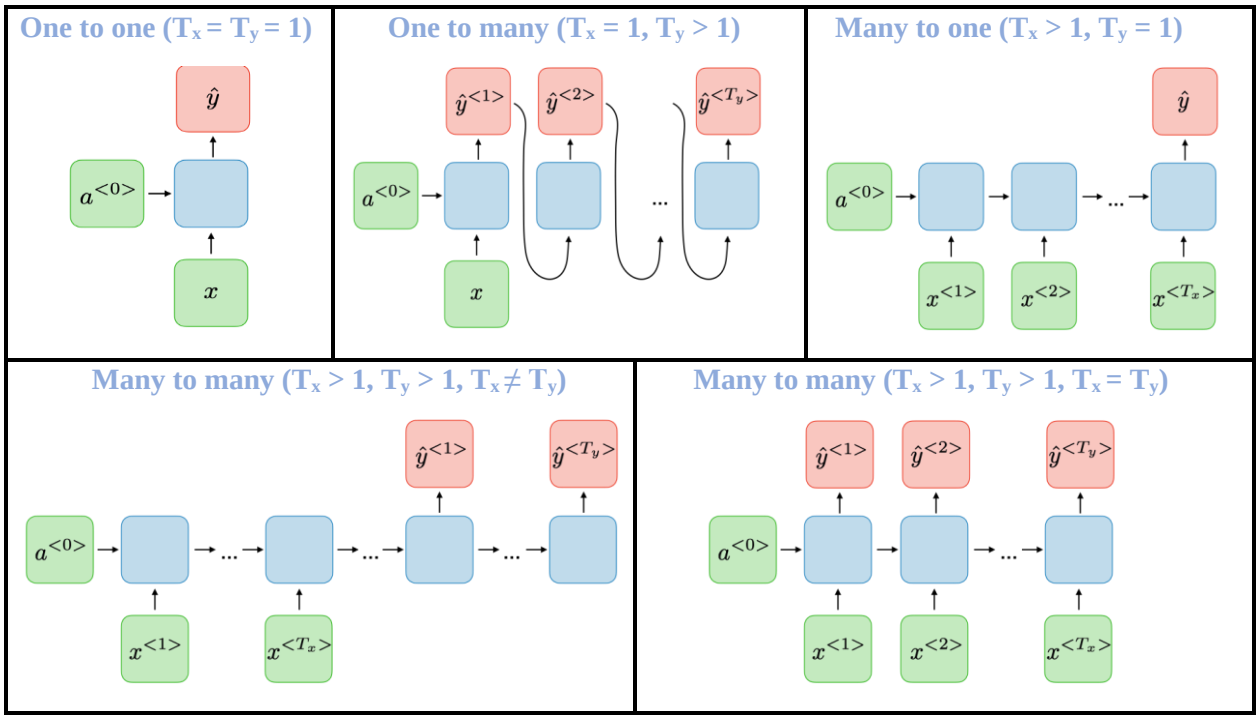
Εικόνα 2.8: *Σύγκριση κλασσικού RNN , LSTM και GRU δικτύου.*

2.3.2 Αρχιτεκτονική ακολουθία σε ακολουθία (Seq2Seq)

Μετά το προσδιορισμό του πυρήνα ή αλλιώς της βασικής μονάδας του μοντέλου, που είναι το LSTM αναδρομικό νευρωνικό δίκτυο, είναι αναγκαίο να καθοριστεί η αρχιτεκτονική του. Τα αναδρομικά νευρωνικά δίκτυα χρησιμοποιούνται σε διάφορες εφαρμογές της επεξεργασίας φυσικής γλώσσας και φωνής. Για αυτό το λόγο, έχουν σχεδιαστεί διαφορετικές αρχιτεκτονικές που προσαρμόζονται αναλόγως τη φύση του προβλήματος που αντιμετωπίζουν. Οι βασικές αυτές αρχιτεκτονικές παρουσιάζονται παρακάτω:

- **Ένα σε Ένα (one to one):** Είναι η πιο απλή περίπτωση, όπου υπάρχει μια είσοδος και μια έξοδος. Αυτή η αρχιτεκτονική είναι όμοια με αυτή ενός κλασσικού εμπρόσθιου νευρωνικού δικτύου.
- **Ένα σε πολλά (one to many):** Σε αυτή τη περίπτωση υπάρχει μια είσοδο και πολλές έξοδοι. Συνεπώς, το νευρωνικό δίκτυο παράγει από ένα στοιχείο εισόδου μια ακολουθία στοιχείων ως έξοδο. Μια γνωστή εφαρμογή αυτή της αρχιτεκτονικής, αποτελεί η αυτόματη παραγωγή κειμένου από εικόνα (image captioning), όπου παίρνοντας μια εικόνα ως είσοδο, παράγεται ένα σύντομο κείμενο που τη περιγράφει.
- **Πολλά σε ένα (many to one):** Σε αυτή τη περίπτωση υπάρχουν πολλές είσοδοι και μία έξοδος. Συνεπώς, το νευρωνικό δίκτυο παράγει από μια ακολουθία στοιχείων στην είσοδο, μια έξοδο. Μια γνωστή εφαρμογή αυτή της αρχιτεκτονικής, αποτελεί η εξαγωγή συναισθήματος από κείμενο (sentiment analysis), όπου από ένα κείμενο το δίκτυο εξάγει συνήθως έναν αριθμό που αντιπροσωπεύει το συναίσθημα που προκαλεί το κείμενο.
- **Πολλά σε πολλά (many to many):** Ονομάζεται αλλιώς ακολουθία σε ακολουθία και όπως προδίδει η ονομασία της, το νευρωνικό δίκτυο παράγει από μια ακολουθία στοιχείων στην είσοδο, μια ακολουθία στοιχειών στην έξοδο. Αναλόγως της εφαρμογής, το πλήθος στοιχείων των δυο ακολουθιών μπορεί να είναι ίσο ή διαφορετικό. Η αυτόματη

παραγωγή μετάφρασης κειμένου και η αυτόματη παραγωγή περίληψης αποτελούν παραδείγματα εφαρμογής της συγκεκριμένης αρχιτεκτονικής.



Εικόνα 2.9: *Αναπαραστάσεις των αρχιτεκτονικών των RNN δικτύων.*

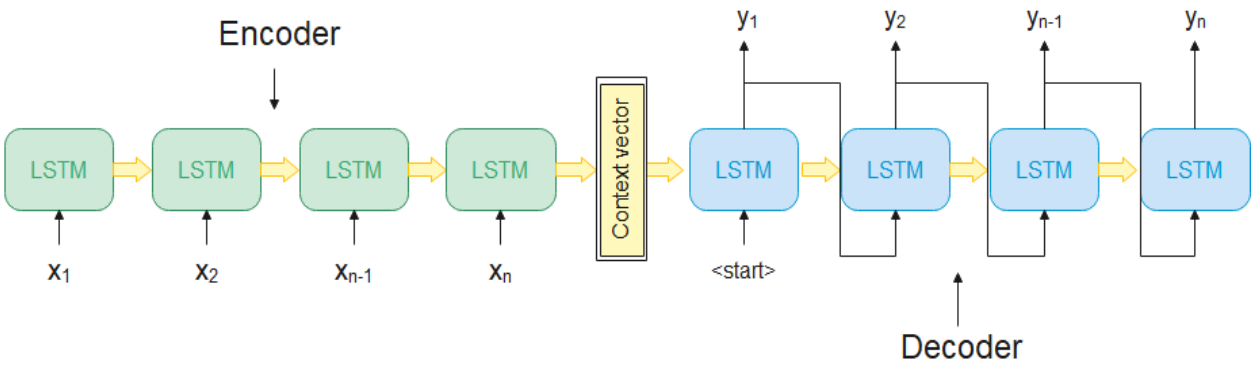
Όπως ήδη έχει αναφερθεί, στο πρόβλημα της αυτόματης παραγωγής περίληψης εφαρμόζεται η αρχιτεκτονική ακολουθία σε ακολουθία ή όπως είναι γνωστή η αρχιτεκτονική seq2seq. Ο λόγος είναι ευκατανόητος, καθώς το μοντέλο τροφοδοτείται με ένα κείμενο που αποτελεί μια ακολουθία λέξεων και παράγει ένα διαφορετικό κείμενο, δηλαδή μια άλλη ακολουθία λέξεων. Το παραγόμενο κείμενο είναι μικρότερης έκτασης από το αρχικό, όποτε ισχύει ότι το πλήθος των στοιχείων της ακολουθίας εξόδου είναι διαφορετικό από αυτό της εισόδου ($T_x > T_y$).

Όταν όμως ένας άνθρωπος συντάσσει τη περίληψη ενός κειμένου, δεν διαβάζει το κείμενο ανά πρόταση ως μια ακολουθία λέξεων και ταυτόχρονα παράγει τη περίληψη. Αντιθέτως, η διαδικασία που ακολουθεί είναι ότι αρχικά διαβάζει ολόκληρο το κείμενο, το κατανοεί, διακρίνει τα σημαντικά σημεία του και στη συνέχεια με βάση αυτά συντάσσει τη περίληψη του. Αντίστοιχη πρακτική πρέπει να ακολουθηθεί και στη περίπτωση αυτόματης παραγωγής περίληψης μέσω υπολογιστή για να είναι αποτελεσματικό το μοντέλο. Συγκεκριμένα το μοντέλο πρώτα επεξεργάζεται ολόκληρο το αρχικό κείμενο κρατώντας τις χρήσιμες πληροφορίες και στη συνέχεια παράγει ένα κείμενο που τις περιέχει, εκφράζοντάς τες με διαφορετικό τρόπο.

Για την υλοποίηση της παραπάνω διαδικασίας, έχει σχεδιαστεί μια συγκεκριμένη αρχιτεκτονική seq2seq που ονομάζεται αρχιτεκτονική κωδικοποιητή – αποκωδικοποιητή (encoder – decoder).

Η συγκεκριμένη αρχιτεκτονική χρησιμοποιείται ευρέως σε προβλήματα φυσικής επεξεργασίας γλώσσας πέρα από την αυτόματη παραγωγή περίληψης. Όπως προδίδει το όνομά της, αποτελείται από δυο κομμάτια τον κωδικοποιητή και τον αποκωδικοποιητή, τα οποία είναι δυο αναδρομικά νευρωνικά δίκτυα με ένα ή παραπάνω στρώματα. Η λειτουργία τους περιγράφεται παρακάτω:

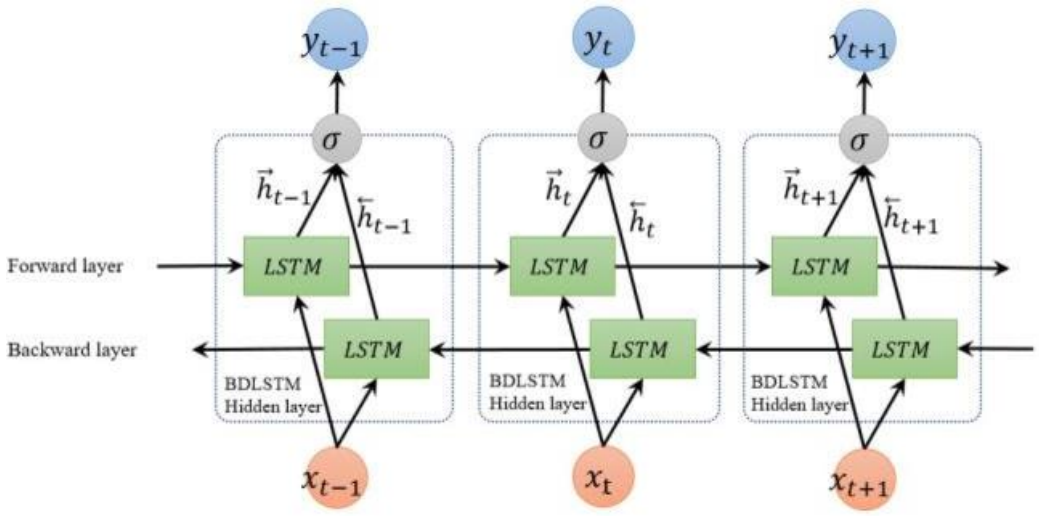
- **Κωδικοποιητής:** Ο ρόλος του κωδικοποιητή είναι να "διαβάσει" όλη την ακολουθία εισόδου, στη περίπτωσή μας το αρχικό κείμενο, και να βγάλει στην έξοδο όλη τη σημαντική πληροφορία από ολόκληρη την ακολουθία. Όλη αυτή η πληροφορία συμπυκνώνεται και εξάγεται από τον κωδικοποιητή με τη μορφή ενός διανύσματος που ονομάζεται διάνυσμα συμφραζομένων (context vector). Η πληροφορία από κάθε στοιχείο της ακολουθίας αποθηκεύεται και μεταδίδεται σε κάθε χρονικό βήμα μέσω των κρυφών καταστάσεων του δικτύου. Η κρυφή κατάσταση $\mathrm{h_t}$ για κάθε χρονικό βήμα t, προκύπτει από τη προηγούμενη κρυφή κατάσταση $\mathrm{h_{t-1}}$ και από την είσοδο που αντιστοιχεί στο τρέχων βήμα $\mathrm{x_t}$. Μαθηματικά περιγράφεται ως εξής:
$$h_t = f(W_{hh} * h_{t-1} + W_{hx} * x_t) \quad (2.11)$$
Η συνάρτηση - αλγόριθμος με τον οποίο υπολογίζεται η κρυφή κατάσταση εξαρτάται από τη μορφή του δικτύου (π.χ. LSTM ή GRU) και αναλύθηκε στο προηγούμενο κεφάλαιο.
- **Αποκωδικοποιητής:** Ο ρόλος του αποκωδικοποιητή είναι να "διαβάσει" τη πληροφορία που εξήγαγε ο κωδικοποιητής και με βάση αυτή, να παραγάγει μια ακολουθία στην έξοδο. Ο αποκωδικοποιητής τροφοδοτείται με το διάνυσμα συμφραζομένων, που έχει εξάγει ο κωδικοποιητής, και το οποίο αποτελεί την αρχική του κατάσταση, ενώ ως αρχικό στοιχείο εισόδου του παίρνει κάποιο στοιχείο που να υποδουλώνει την εκκίνησή του (π.χ. <START>). Η κρυφή κατάσταση του αποκωδικοποιητή υπολογίζεται σε κάθε χρονικό βήμα με αντίστοιχο τρόπο με του κωδικοποιητή, με τη διαφορά ότι χρησιμοποιείται η προηγούμενη έξοδος $\mathrm{y_{t-1}}$ του αποκωδικοποιητή σαν τρέχουσα είσοδο x_t. Δηλαδή :
$$y_t = f(W_s * h_{t-1} + W_{hy} * y_{t-1}) \quad (2.12)$$
Παρομοίως, η συνάρτηση - αλγόριθμος με τον οποίο υπολογίζεται η κρυφή κατάσταση εξαρτάται από τη μορφή του δικτύου (π.χ. LSTM ή GRU).
Η έξοδος του δικτύου σε κάθε χρονικό βήμα y_t, είναι ένα διάνυσμα πιθανότητας με βάση το οποίο αποφασίζεται τελικά η τελική έξοδος του μοντέλου. Και υπολογίζεται ως εξής:
$$h_t = softmax(W_{hh} * h_t) \quad (2.13)$$
Όπου softmax είναι μια κανονικοποιημένη εκθετική συνάρτηση, που χρησιμοποιείται για τη δημιουργία πιθανοτικών τιμών στο διάστημα [0,1].



Εικόνα 2.10: *Απεικόνιση αρχιτεκτονικής κωδικοποιητή - αποκωδικοποιητή LSTM δικτύων.*

Στη παρούσα εργασία, για την υλοποίηση του συστήματος αυτόματης παραγωγής περίληψης, χρησιμοποιούνται κωδικοποιητής και αποκωδικοποιητής αρχιτεκτονικής LSTM. Σημειώνεται όμως ότι για την καλύτερη αντιμετώπιση του προβλήματος και την υλοποίηση ενός πιο αποτελεσματικού μοντέλου, χρησιμοποιείται αμφίδρομος κωδικοποιητής (bidirectional encoder) LSTM αντί του κλασσικού μονόδρομου (unidirectional encoder). Ο λόγος για τον οποίον έγινε αυτή η επιλογή αναλύεται παρακάτω.

Ένας κλασσικός κωδικοποιητής επεξεργάζεται την ακολουθία στην είσοδό του σειριακά, δηλαδή με τη σειρά κάθε στοιχείο της ακολουθίας από την αρχή ως το τέλος της, και στην έξοδό του παράγει ένα διάνυσμα με πληροφορία από όλη την ακολουθία. Με αυτό το τρόπο όμως η πληροφορία που προκύπτει είναι μοντελοποιημένη να δείχνει εξαρτήσεις μόνο από τα προηγούμενα στοιχεία της ακολουθίας. Στη πραγματικότητα όμως, στο γραπτό λόγο υπάρχουν νοηματικές, συντακτικές και γραμματικές εξαρτήσεις όχι μόνο από τις προηγούμενες λέξεις, αλλά και από τις επόμενες. Με αντίστοιχο τρόπο, όταν ο άνθρωπος διαβάζει ένα κείμενο για την παραγωγή της περίληψής του, χρειάζεται πολλές φορές να κοιτάξει και παρακάτω από το σημείο που εστιάζει κάθε στιγμή. Έτσι προέκυψε η ανάγκη για χρήση ενός αμφίδρομου κωδικοποιητή, ο οποίος λειτουργεί όπως ακριβώς ο κλασσικός κωδικοποιητής αλλά και προς την άλλη κατεύθυνση. Συγκεκριμένα, επεξεργάζεται την ακολουθία εισόδου και από το τέλος προς την αρχή, παράγοντας ένα δεύτερο διάνυσμα πληροφορίας με εξαρτήσεις από επόμενα στοιχεία στη σειρά της ακολουθίας. Τα δύο διανύσματα συνδυάζονται και τροφοδοτούν τον αποκωδικοποιητή για την παραγωγή της περίληψης.



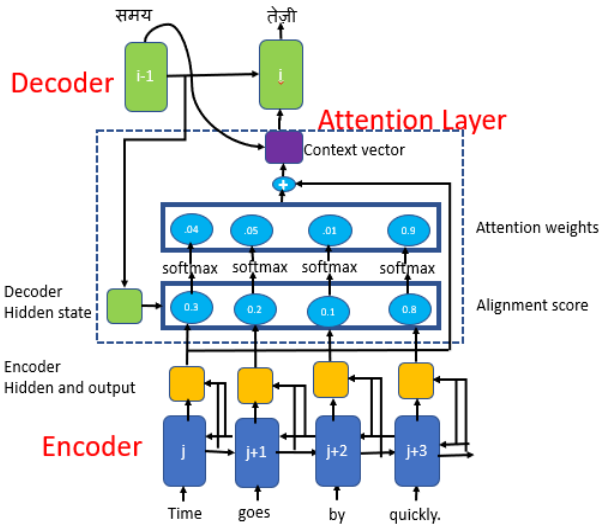
Εικόνα 2.11: *Απεικόνιση αμφίδρομου κωδικοποιητή LSTM.*

2.3.3 Μηχανισμός Προσοχής

Όπως αναφέρθηκε στη προηγούμενη ενότητα, στην αρχιτεκτονική κωδικοποιητή – αποκωδικοποιητή ο κωδικοποιητής κωδικοποιεί την πληροφορία από ολόκληρη την ακολουθία εισόδου σε ένα διάνυσμα συμφραζομένων (context vector), που αποτελεί και την τελευταία κρυφή κατάσταση του. Το διάνυσμα συμφραζομένων τροφοδοτείται στον αποκωδικοποιητή, ο οποίος αρχίζει τη παραγωγή της περίληψης λέξη προς λέξη, βασιζόμενος στο διάνυσμα αυτό. Έτσι η περίληψη παράγεται συνυπολογίζοντας ολόκληρο το κείμενο εισόδου. Πέραν όμως από τη συνολική επίγνωση του αρχικού κειμένου, πρέπει το μοντέλο να μπορεί να διακρίνει τα κρίσιμα σημεία του κειμένου και να εστιάζει σε αυτά κατά την παραγωγή της περίληψης. Ακολουθώντας ουσιαστικά το παράδειγμα του ανθρώπου, ο οποίος ενώ διαβάζει ολόκληρο το κείμενο όταν παράγει την περίληψή του, εστιάζει κάθε στιγμή σε συγκεκριμένα σημεία (φράσεις ή λέξεις) τις οποίες θεωρεί σημαντικότερες.

Είναι λοιπόν απαραίτητη η χρήση ενός μηχανισμού ο οποίος θα προσθέσει στο context vector πληροφορία σχετικά με τη σημαντικότητα κάθε λέξης, έτσι ώστε ο αποκωδικοποιητής να εστιάζει σε συγκεκριμένες σχετικές λέξεις ή φράσεις της ακολουθίας εισόδου κάθε φορά που παράγει μια έξοδο. Ο μηχανισμός αυτός ονομάζεται μηχανισμός προσοχής (attention mechanism) και αντιπροσωπεύει τη “προσοχή” που αποδίδει ο αποκωδικοποιητής σε κάθε στοιχείο της ακολουθίας. Ο πιο γνωστός μηχανισμός προσοχής είναι ο μηχανισμός προσοχής Bahdanau, ο οποίος παρουσιάστηκε πρώτη φορά από τον Bahdanau για την υλοποίηση ενός συστήματος αυτόματης μετάφρασης κειμένου [26]. Από τότε ο μηχανισμός χρησιμοποιείται ευρέως, εμφανίζοντας σημαντική βελτίωση αποτελεσμάτων σε προβλήματα της επεξεργασίας της φυσικής γλώσσας, όπως αυτό της αυτόματης παραγωγής περίληψης.

Η πρόσθεση του μηχανισμού προσοχής Bahdanau στην αρχιτεκτονική του μοντέλου, αυξάνει κάπως την πολυπλοκότητά του αλλά βελτιώνει σημαντικά την απόδοσή του. Η κύρια διαφορά που προκύπτει είναι ότι για τον υπολογισμό του context vector χρησιμοποιούνται όλες οι κρυφές καταστάσεις του κωδικοποιητή (και προς τις δύο κατευθύνσεις) και του αποκωδικοποιητή. Αντιθέτως, στην κλασσική αρχιτεκτονική χωρίς τον μηχανισμό προσοχής, μόνο η τελευταία κρυφή κατάσταση του κωδικοποιητή λαμβάνεται υπόψιν. Η λειτουργία του μηχανισμού προσοχής Bahdanau περιγράφεται αναλυτικά παρακάτω.



***Εικόνα 2.12**: Απεικόνιση αρχιτεκτονικής seq2seq με μηχανισμό προσοχής [22].*

Αρχικά, ο μηχανισμός προσοχής Bahdanau προσπαθεί να ευθυγραμμίσει ή αλλιώς να "ταιριάξει", την ακολουθία εξόδου με την ακολουθία εισόδου, χρησιμοποιώντας ένα μέτρο ευθυγράμμισης. Συγκεκριμένα, το μέτρο ευθυγράμμισης προσδιορίζει πόσο καλά "ταιριάζουν" η έξοδος στη θέση i της ακολουθίας εξόδου, με τα στοιχεία κοντά στη θέση j της ακολουθίας εισόδου. Το μέτρο υπολογίζεται από τη προηγούμενη κρυφή κατάσταση του αποκωδικοποιητή s_{i-1} και την αμφίδρομη κρυφή κατάσταση h_j του κωδικοποιητή, όπως φαίνεται στην **2.14**.

$$e_{ij} = \alpha(s_{i-1}, h_j) \tag{2.14}$$

Η συνάρτηση α μοντελοποιείται από ένα εκπαιδευόμενο νευρωνικό δίκτυο, και καθορίζει πόσο σχετική είναι η κάθε κωδικοποιημένη κρυφή κατάσταση h_j της ακολουθίας εισόδου με την προς παραγωγή έξοδο y_i του αποκωδικοποιητή, δεδομένης της προηγούμενης κρυφής κατάστασης s_{i-1}.

Στη συνέχεια, με βάση το μέτρο ευθυγράμμισης αντιστοιχίζονται σε κάθε στοιχείο της ακολουθίας εισόδου κάποιες συγκεκριμένες τιμές, που ονομάζονται βάρη προσοχής (attention weights). Αυτά τα βάρη, πληροφορούν τον αποκωδικοποιητή πόση "προσοχή" πρέπει να

αποδώσει σε κάθε λέξη του αρχικού κειμένου, κάθε φορά που παράγει μια λέξη της περίληψης. Ο μαθηματικός υπολογισμός των βαρών πραγματοποιείται ως εξής:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})} \quad (2.15)$$

Ουσιαστικά εφαρμόζεται μια softmax συνάρτηση που θέτει τιμές των βαρών από 0 έως 1. Όσο μεγαλύτερη η τιμή του βάρους τόσο μεγαλύτερη η σημασία της εισόδου που αντιστοιχεί.

Έπειτα, η πληροφορία της προσοχής ενσωματώνεται στο διάνυσμα συμφραζομένων. Για κάθε παραγόμενη έξοδο y_i του αποκωδικοποιητή, το διάνυσμα συμφραζομένων c_i υπολογίζεται ως το σταθμισμένο άθροισμα των γινομένων των βαρών προσοχής με τις αντίστοιχες κωδικοποιημένες καταστάσεις, όπως φαίνεται παρακάτω:

$$c_i = \sum_j a_{ij} h_j \quad (2.16)$$

Τελικά, για την παραγωγή της εκάστοτε λέξης ο αποκωδικοποιητής χρησιμοποιεί το διάνυσμα συμφραζομένων c_i, τη προηγούμενη έξοδό του y_{i-1} και τη προηγούμενη κρυφή του κατάσταση s_{i-1}, σύμφωνα με τη παρακάτω εξίσωση :

$$y_i = f(s_{i-1}, c_i, y_{i-1}) \quad (2.17)$$

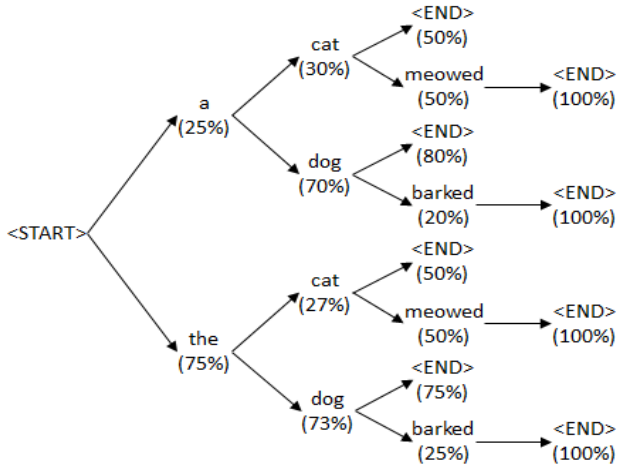
Ο μηχανισμός προσοχής Bahdanau αποτελεί πλέον βασικό κομμάτι της αρχιτεκτονικής seq2seq και χρησιμοποιείται στη πλειοψηφία των συστημάτων αυτόματης παραγωγής περίληψης. Για αυτό το λόγο υιοθετείται και στη παρούσα εργασίας.

2.3.4 Μηχανισμός Ακτινικής Αναζήτηση

Ένα ακόμη σημαντικό στάδιο του συστήματος αυτόματης παραγωγής περίληψης, αποτελεί η τελική επιλογή της παραγόμενης λέξης από τον αποκωδικοποιητή σε κάθε βήμα, για την σύνταξη της περίληψης. Σε όλα τα νευρωνικά δίκτυα, τα αποτελέσματα και οι προβλέψεις τους αναπαρίστανται αριθμητικά με συγκεκριμένες τιμές που υποδηλώνουν τη πιθανότητα της εκάστοτε πρόβλεψης να είναι η ορθή. Στη περίπτωση της αυτόματης παραγωγής περίληψης, καθώς και στα υπόλοιπα προβλήματα επεξεργασίας φυσικής γλώσσας, η έξοδος σε κάθε βήμα αποκωδικοποίησης αποτελεί μια πιθανοτική κατανομή λέξεων.

Η κάθε λέξη της κατανομής αντιπροσωπεύεται από μια τιμή στο διάστημα [0,1], που εκφράζει τη πιθανότητά της να είναι αυτή η καταλληλότερη λέξη προς επιλογή. Για ευκολία, η κατανομή αυτή μπορεί να αναπαρασταθεί σαν ένα δίκτυο με ακμές τις πιθανές λέξεις και τις αντίστοιχες πιθανότητές τους σε κάθε βήμα αποκωδικοποίησης, όπως φαίνεται στο σχήμα **2.13**. Οι πιθανές λέξεις ανήκουν σε ένα καθορισμένο λεξιλόγιο, και το πλήθος τους είναι ίσο με το μέγεθος αυτού του λεξιλογίου. Το λεξιλόγιο συνήθως προκύπτει από τα κείμενα που χρησιμοποιούνται για την εκπαίδευση του μοντέλου. Σε αυτό το σημείο, φαίνεται πόσο δύσκολο είναι η δημιουργία ενός

καθολικού συστήματος αυτόματης περίληψης, καθώς πρέπει να στηρίζεται σε ένα τεράστιο λεξιλόγιο και συνεπώς να εκπαιδευτεί με ένα τεράστιο όγκο κειμένων.



Εικόνα 2.13: *Απεικόνιση ακτινικής αναζήτησης.*

Ο κλασσικός και απλούστερος αλγόριθμος που μπορεί να χρησιμοποιηθεί για την επιλογή της καταλληλότερης εξόδου, είναι ένας άπληστος αλγόριθμος που ονομάζεται άπληστη αναζήτηση (greedy search). Ο άπληστος αλγόριθμος δεν πραγματοποιεί κάποια διεργασία, αλλά απλά επιλέγει κάθε φορά τη λέξη με τη μεγαλύτερη τιμή πιθανότητας. Ο συγκεκριμένος αλγόριθμος βασίζεται στην ιδέα ότι επιλέγοντας σε κάθε βήμα τη λέξη που φαίνεται να είναι η καταλληλότερη, η συνολική ακολουθία λέξεων που θα προκύψει στο τέλος θα είναι η βέλτιστη. Η προσέγγιση αυτή όμως αποδεικνύεται πως δεν είναι πάντα σωστή, καθώς η επιλογή της λέξης με τη μεγαλύτερη τιμή σε κάθε βήμα δεν εξασφαλίζει τη μέγιστη συνολική τιμή στο τέλος της ακολουθίας. Αυτό το πρόβλημα καλείται να λύσει ένας άλλος πιο εκλεπτυσμένος αλγόριθμος, αυτός της ακτινικής αναζήτησης (beam search).

Η ακτινική αναζήτηση έχει παρόμοια λειτουργία με τον άπληστο αλγόριθμο, με την κύρια διαφορά να είναι ότι σε κάθε βήμα ερευνά πολλαπλές επιλογές και όχι απλά αυτή που φαίνεται καλύτερη. Ο αριθμός των επιλογών ή αλλιώς το βάθος της αναζήτησης που μελετά κάθε φορά ο αλγόριθμος εξαρτάται από μια παράμετρο που ονομάζεται πλάτος ακτίνας (beam width). Έστω ότι ο αλγόριθμος έχει πλάτος ακτίνας k, τότε σε κάθε έξοδο του αποκωδικοποιητή επιλέγει τις k λέξεις με τις μεγαλύτερες τιμές και στη συνέχεια για κάθε μια από αυτές επιλέγει στην επόμενη έξοδο πάλι τις k λέξεις με τις μεγαλύτερες τιμές, υπολογίζοντας σε κάθε βήμα το συνδυαστικό σκορ για κάθε περίπτωση. Η διαδικασία αυτή συνεχίζεται μέχρι το τέλος της αποκωδικοποίησης, όπου ο αλγόριθμος παράγει την ακολουθία με το μεγαλύτερο συνολικό σκορ.

Από τη διαδικασία αυτή, εύκολα μπορεί να παρατηρήσει κάποιος ότι η ακτινική αναζήτηση απαιτεί αρκετούς υπολογισμούς, οι οποίοι αυξάνονται εκθετικά με την αύξηση του πλάτους της

ακτίνας. Παρά το γεγονός όμως ότι ο συγκεκριμένος αλγόριθμος αυξάνει τις απαιτήσεις του μοντέλου όσον αφορά την υπολογιστική ισχύ, δίνει πολύ καλύτερα αποτελέσματα και για αυτό προτιμάται.

2.3.5 Μηχανισμός αντιγραφής άγνωστων λέξεων

Όπως αναφέρθηκε και στη προηγούμενη ενότητα, οι λέξεις που παράγονται από το μοντέλο σε κάθε βήμα και συντάσσουν στο τέλος τη περίληψη, ανήκουν σε ένα καθορισμένο λεξιλόγιο. Το λεξιλόγιο αυτό προκύπτει κατά κύριο λόγο από το σύνολο των κειμένων εκπαίδευσης, και ουσιαστικά αποτελείται από κάθε διαφορετική λέξη που υπάρχει μέσα στα κείμενα αυτά. Το μοντέλο μέσω της εκπαίδευσης έχει κατανοήσει τις λέξεις του συγκεκριμένου λεξιλογίου και μπορεί να τις αναγνωρίσει, αλλά αδυνατεί να επεξεργαστεί άγνωστες λέξεις που δεν ανήκουν σε αυτό. Κατά συνέπεια, όσο λιγότερα είναι τα δεδομένα εκπαίδευσης, τόσο περιορίζεται το λεξιλόγιο του μοντέλου και αντίστοιχα η ικανότητα του να παράγει τη περίληψη ενός νέου κειμένου.

Ο προφανής τρόπος αντιμετώπισης αυτού του προβλήματος, είναι η σημαντική αύξηση του όγκου των δεδομένων εκπαίδευσης, για να δημιουργηθεί ένα λεξιλόγιο που θα περιέχει όσο γίνεται περισσότερες λέξεις και θα δώσει τη δυνατότητα στο μοντέλο να ανταπεξέλθει σε μια μεγάλη γκάμα κειμένων. Παρ' όλα αυτά, το λεξιλόγιο μιας γλώσσας μπορεί να είναι τόσο μεγάλο, που χρειάζεται ένας τεράστιος αριθμός κειμένων εκπαίδευσης για να “μάθει” το μοντέλο όλες τις λέξεις που υπάρχουν. Η ελληνική γλώσσα αποτελεί ένα χαρακτηριστικό παράδειγμα, καθώς θεωρείται μια από τις πλουσιότερες γλώσσες στο κόσμο , όσον αφορά τον αριθμό των λέξεων που την απαρτίζουν. Συνεπώς, είναι σχεδόν αδύνατο να μην υπάρχουν κάθε φορά κάποιες άγνωστες λέξεις για το μοντέλο. Επιπλέον, η συλλογή ενός τεράστιου όγκου διαφορετικών θεματικά κειμένων δεν είναι πάντα εύκολη και η εκπαίδευση του μοντέλου με αυτά απαιτεί μεγάλη υπολογιστική ισχύ και αρκετό χρόνο.

Ο συνδυασμός όλων των παραπάνω, δημιουργεί την ανάγκη για μια δεύτερη λύση. Η λύση αυτή αποτελεί ένας μηχανισμός αντιγραφής άγνωστων λέξεων γνωστός ως pointer generator. Ο μηχανισμός αυτός παρουσιάστηκε πρώτη φορά από την έρευνα [27], και η λειτουργία του είναι ουσιαστικά να αντιγράφει κάποια λέξη από την είσοδο στην έξοδο, δηλαδή από το αρχικό κείμενο στη περίληψη του, όταν η λέξη αυτή θεωρείται σημαντική για την περίληψη αλλά δεν περιέχεται στο λεξιλόγιο του μοντέλου. Ο μηχανισμός αυτός σε κάθε επανάληψη παραγωγής εξόδου, χρησιμοποιεί το διάνυσμα συμφραζομένων c_t , τη κρυφή κατάσταση του αποκωδικοποιητή h_t και τη είσοδο του x_t για να υπολογίσει τη πιθανότητα παραγωγής (generation probability), σύμφωνα με τη παρακάτω εξίσωση:

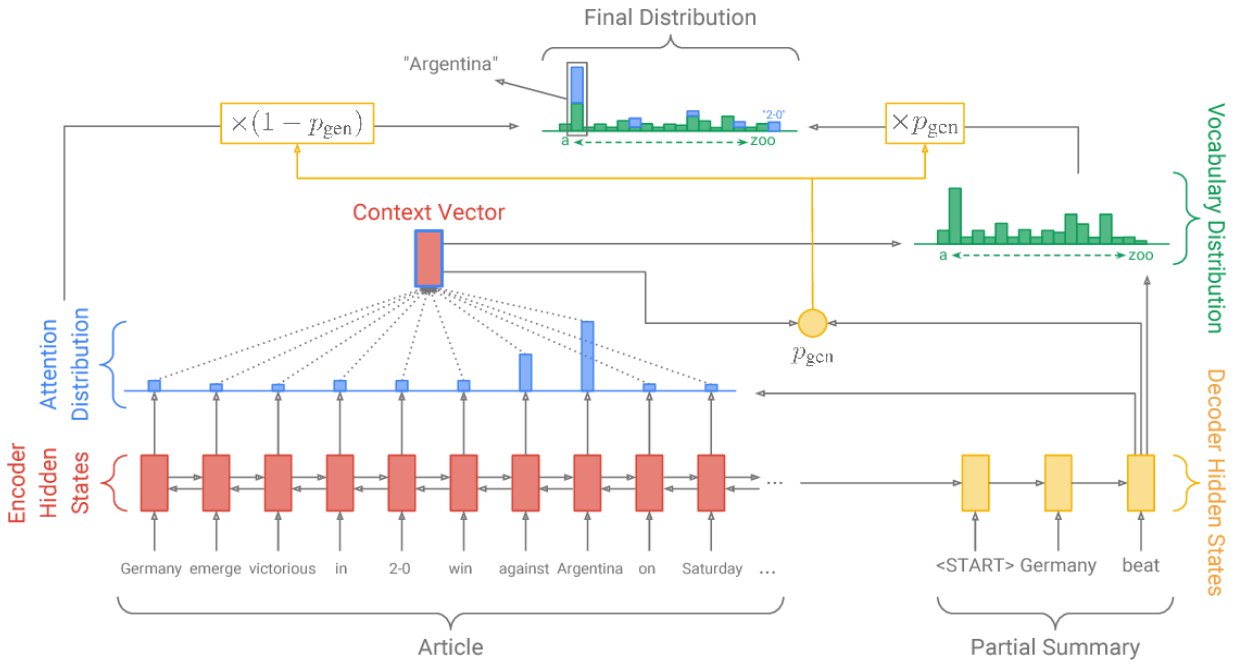
$$p_{gen} = \sigma(W_c * c_t + W_s * s_t + W_x * x_t) \quad (2.18)$$

όπου W_c, W_t, W_x είναι πίνακες βαρών και σ η σιγμοειδή συνάρτηση.

Η πιθανότητα αυτή χρησιμοποιείται σαν ένας διακόπτης που επιλέγει αν η παραγόμενη έξοδος θα προκύψει από τη φυσιολογική διαδικασία που περιεγράφηκε στη προηγούμενη ενότητα και θα ανήκει στο γνωστό λεξιλόγιο ή θα αντιγραφεί από το κείμενο εισόδου, σ. Η επιλογή αυτή εκφράζεται μαθηματικά ως εξής :

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen})\sum_{i:w} a_i \qquad (2.19)$$

όπου P_{vocab} είναι η πιθανοτική κατανομή του λεξιλογίου και $\sum_{i:w} a_i$ είναι η πιθανοτική κατανομή που προκύπτει από το μηχανισμό προσοχής και εκφράζει την είσοδο. Όπως είναι, προφανές αν η πιθανότητα p_{gen} είναι κοντά στο 1 τότε η λέξη προκύπτει από το λεξιλόγιο, ενώ αν είναι 0 αντιγράφεται από την είσοδο. Στο παρακάτω γράφημα παρουσιάζεται το συνολικό σύστημα που υλοποιήθηκε προσθέτοντας το μηχανισμό pointer generator.



Εικόνα 2.14: *Οπτική αναπαράσταση σου συνολικού συστήματος αυτόματης παραγωγής περίληψης που υλοποιήθηκε (αρχιτεκτονική κωδικοποιητή – αποκωδικοποιητή, μηχανισμός προσοχής, μηχανισμός αντιγραφής άγνωστων λέξεων) [12].*

Σημειώνεται πως ο μηχανισμός αντιγραφής λέξεων αποτελεί στοιχείο βελτίωσης του προς υλοποίηση συστήματος και για αυτό το λόγο υπάρχει η δυνατότητα της χρήσης ή μη χρήσης του μέσω του λογισμικού. Επίσης, πραγματοποιήθηκαν πειράματα για τη σύγκριση των αποτελεσμάτων με και χωρίς τη χρήση του μηχανισμού, τα οποία παρουσιάζονται στο κεφάλαιο 5.

2.3.6 Μηχανισμός επικάλυψης

Ένα άλλο φαινόμενο που παρατηρείται συχνά στα συστήματα αυτόματα παραγωγής περίληψης και γενικά στα μοντέλα με αρχιτεκτονική ακολουθία σε ακολουθία, είναι η επανάληψη φράσεων και λέξεων. Οι επαναλήψεις αυτές εμφανίζονται συχνότερα σε κείμενα μεγάλης έκτασης, και έχει ως αποτέλεσμα τη σύνταξη δυσνόητων περιλήψεων και κατά επέκταση τη μείωση της αποτελεσματικότητας του συστήματος. Για την αντιμετώπιση αυτού του προβλήματος, παρουσιάστηκε για πρώτη φορά από την έρευνα [28] ο μηχανισμός επικάλυψης (coverage mechanism).

Ο μηχανισμός αυτός ενημερώνει το μοντέλο για το αν οι πιθανές έξοδοι σε κάθε βήμα, έχουν ήδη παραχθεί σε προηγούμενα βήματα. Αυτό επιτυγχάνεται υπολογίζοντας ένα διάνυσμα επικάλυψης (coverage vector), που αποτελεί το άθροισμα όλων των βαρών προσοχής που έχουν υπολογιστεί σε όλα τα προηγούμενα βήματα αποκωδικοποίησης:

$$c_t = \sum_{t'}^{t-1} \alpha_{t'} \tag{2.20}$$

Ουσιαστικά το διάνυσμα αυτό εκφρα΄ζει μια μη κανονικοποιημένη κατανομή του ποσοστού της επικάλυψης που έχουν δεχθεί από τον μηχανισμό προσοχής οι λέξεις της ακολουθίας εισόδου. Το διάνυσμα επικκάλυξης προστίθεται σαν επιπλέον είσοδο στον υπολογισμό του μέτρου προσοχής σε κάθε βήμα, μεταβάλλοντας την εξίσωση **2.14** ως εξής :

$$e_{ij} = \alpha(s_{i-1}, h_j, c_j) \tag{2.21}$$

όπου η συνάρτηση α μοντελοποιείται από ένα εκπαιδευόμενο νευρωνικό δίκτυο.

Με αυτό το τρόπο ο μηχανισμός προσοχής πληροφορείται μέσω του διανύσματος επικάλυψης για την κατανομή προσοχής που έχει διαμορφώσει στα προηγούμενα βήματα, δηλαδή γνωρίζει τις λέξεις στις οποίες έχει εστιάσει περισσότερο. Ως επακόλουθο, ο μηχανισμός προσοχής δεν θα εστιάζει τη προσοχή του στις ίδιες λέξεις επανελημένα, αποφεύγοντας τις επαναλήψεις.

Ο μηχανισμός επικάλυψης, όπως και ο μηχανισμός αντιγραφής λέξεων, αποτελεί στοιχείο βελτίωσης του προς υλοποίηση συστήματος, και για αυτό το λόγο υπάρχει η δυνατότητα της χρήσης ή μη χρήσης του μέσω του λογισμικού. Επίσης, ο τρόπος με τον οποίο επηρέαζει την επίοδοση του συστήματος ερευνάται στο κεφάλαιο 5.

3. Ελληνικά δεδομένα και προ-επεξεργασία τους

3.1 Σύνολο δεδομένων ελληνικών άρθρων-περιλήψεων

Όπως έχει ήδη αναφερθεί στα προηγούμενα κεφάλαια, η επιστήμη της μηχανικής μάθησης είναι μια επιστήμη που βασίζεται κατά κύριο λόγο σε δεδομένα, και για αυτό πολλές φορές αναφέρεται ως επιστήμη μελέτης των δεδομένων. Είναι γεγονός ότι χωρίς δεδομένα δε μπορεί να υπάρξει ένα λειτουργικό και χρήσιμο σύστημα μηχανικής μάθησης, καθώς δε μπορεί να μοντελοποιηθεί η λειτουργία του. Μέσω της μηχανικής μάθησης αναδεικνύεται κατά κάποιο τρόπο η αξία ενός μεγάλου πλήθους δεδομένων αλλά και πόση πληροφορία κρύβεται μέσα σε αυτά , που μόνος του ο άνθρωπος δε θα μπορούσε να κατανοήσει.

Η ραγδαία ανάπτυξη της βαθιάς μάθησης τα τελευταία χρόνια καθιστά την ανάγκη για δεδομένα ακόμα πιο έντονη, καθώς τα προβλήματα που καλείται να λύσει απαιτούν ένα τεράστιο όγκο δεδομένων. Όσο πιο σύνθετα και γενικευμένα είναι αυτά τα προβλήματα, τόσο πιο σύνθετες γίνονται και οι αρχιτεκτονικές των μοντέλων που καλούνται να τα αντιμετωπίσουν, αυξάνοντας σημαντικά τις απαιτήσεις τους σε δεδομένα. Από την άλλη, η ανάπτυξη της τεχνολογίας και η εξάπλωση του διαδικτύου, προκαλούν μία ραγδαία αύξηση των διαθέσιμων δεδομένων και πληροφορίων που κατακλύζουν καθημερινά τις ζωές μας. Το γεγονός αυτό κάνει εφικτή την υλοποίηση τέτοιων σύνθετων μοντέλων μηχανικής μάθησης και δίνει ώθηση για την υλοποίηση νέων ακόμα πιο σύνθετων.

Στη παρούσα εργασία υλοποιείται ένα μοντέλο αυτόματης παραγωγής περίληψης ελληνικών κειμένων βασισμένο σε μηχανισμούς βαθιάς μάθησης και είναι προφανές ότι χρειάζονται δεδομένα, τα οποία αποτελούνται από ελληνικά κείμενα. Συγκεκριμένα, τα δεδομένα συλλέχθηκαν από ελληνικές ιστοσελίδες ειδήσεων και αποτελούνται από άρθρα μαζί με τους τίτλους και την σύντομη περιγραφή (περίληψη) τους. Το συνολικό πλήθος των άρθρων είναι 215385 και αντίστοιχο είναι το πλήθος των τίτλων και των περιλήψεων. Τα δεδομένα είναι όλα μαζί μορφοποιημένα σε ένα csv αρχείο με 215385 γραμμές και χωρισμένα σε τρείς στήλες (title_alt->τίτλος, description->άρθρο, short description->περίληψη). Ο μέσος όρος των μηκών των δεδομένων ανά κατηγορία είναι: άρθρο -> 266.15 λέξεις, περίληψη -> 25.72 λέξεις και τίτλος -> 11.66 λέξεις.

Για την εκπαίδευση και τις δοκιμές του μοντέλου χρησιμοποιήθηκαν τα ζεύγη description(άρθρα) – short_description(περίληψη). Οι τίτλοι χρησιμοποιήθηκαν μόνο στη δημιουργία του λεξιλογίου και χωρίς ιδιαίτερη συμβολή, καθώς συνήθως συντάσσονται από αυτούσιες λέξεις του άρθρου. Όπως είναι προφανές, αυτό το πλήθος δεδομένων δεν είναι αρκετό για την μοντελοποίηση ενός αποτελεσματικού γενικού σκοπού συστήματος αυτόματης παραγωγής περίληψης της ελληνικής γλώσσας, καθώς σε αυτή τη περίπτωση θα χρειαζόταν ένας πραγματικά τεράστιος όγκος δεδομένων. Παρόλα αυτά σημειώνεται ότι το υλοποιήσιμο σύστημα είναι αρκετά αποδοτικό, ιδιαίτερα σε κείμενα με σχετικό αντικείμενο – θέμα με τα

άρθρα που υπάρχουν στα δεδομένα και χρησιμοποιήθηκαν στην εκπαίδευσή του. Τα άρθρα συλλέχθηκαν από ιστοσελίδες ελληνικών ειδήσεων και συγκεκριμένα πολιτικών, οικονομικών και αθλητικών, οπότε και το περιεχόμενό τους είναι αντίστοιχο.

Ένα σύνολο δεδομένων όμως στην αρχική, "ακατέργαστη" μορφή του (raw data) δεν είναι κανονικοποιημένο και περιέχει αρκετά σφάλματα, για αυτό τις περισσότερες φορές δεν είναι διαχειρίσιμο από το σύστημα μηχανικής μάθησης και σε περίπτωση που είναι δεν προσφέρει τα βέλτιστα αποτελέσματα. Για να ενσωματωθούν τα δεδομένα στο σύστημα και να είναι επεξεργάσιμα από τον εκάστοτε αλγόριθμο μηχανικής μάθησης, πρέπει να έχει προηγηθεί η κατάλληλη επεξεργασία και κανονικοποιησή τους. Η προ-επεξεργασία των δεδομένων αποτελεί ένα απαραίτητο και πολύ σημαντικό στάδιο για το σχεδιασμό και την υλοποίηση ενός εύρωστου συστήματος μηχανικής μάθησης, καθώς επηρεάζει σημαντικά την απόδοσή του. Η ανάγκη προ-επεξεργασίας των δεδομένων είναι ακόμα πιο εμφανής στα συστήματα βαθιάς μάθησης, καθώς μεγαλώνει η πολυπλοκότητα και ο όγκος των δεδομένων. Στην επόμενη ενότητα παρουσιάζονται αναλυτικά τα βήματα προ-επεξεργασίας των δεδομένων που ακολουθήθηκαν στη παρούσα εργασία.

3.2 Προ-επεξεργασία ελληνικών δεδομένων – κειμένων

Η προ-επεξεργασία των δεδομένων είναι ιδιαίτερα σημαντική στο πρόβλημα της αυτόματης παραγωγής περίληψης, καθώς προέρχονται από γραπτό λόγο (κείμενα), όπου η εμφάνιση σφαλμάτων είναι αρκετά συχνή. Επίσης, αναγκαία είναι και η κατάλληλη μορφοποίηση των δεδομένων για την εισαγωγή τους στο μοντέλο. Στα στάδια προ-επεξεργασίας που ακολουθήθηκαν παρουσιάζονται με τη σειρά παρακάτω.

1. **Μορφοποίηση και εισαγωγή δεδομένων**. Αρχικό στάδιο της προ-επεξεργασίας αποτελεί η εισαγωγή των δεδομένων στη κατάλληλη μορφή για την περαιτέρω επεξεργασίας τους. Αυτό πραγματοποιήθηκε με τη βοήθεια της βιβλιοθήκης pandas της γλώσσας προγραμματισμού python και αναλύεται σε παρακάτω κεφάλαιο. Συγκεκριμένα έγινε η μετατροπή και η αποθήκευση των δεδομένων από το csv αρχείο, σε τρία data structures (title, summary, text).
2. **Αφαίρεση όλων των κενών και δυσνόητων καταχωρήσεων.** Στο σύνολο δεδομένων υπάρχουν πολλές καταχωρήσεις που είναι κενές. Δηλαδή εμφανίζονται περιπτώσεις που υπάρχει το άρθρο και λείπει η περίληψη και αντιστρόφως. Επίσης, υπάρχουν αντίστοιχες περιπτώσεις όπου εμφανίζονται αυτοματοποιημένα και δυσνόητα μηνύματα. Πραγματοποιήθηκε λοιπόν, η εύρεση και αφαίρεση όλων αυτών των καταχωρήσεων για την καλύτερη εκπαίδευση του μοντέλου.
3. **Αφαίρεση πολλαπλών καταχωρήσεων.** Αντιστοίχως με το προηγούμενο βήμα, υπάρχουν καταχωρήσεις που δεν είναι μοναδικές αλλά εμφανίζονται πολλαπλές φορές. Τα αντίγραφα αυτών των καταχωρήσεων αφαιρέθηκαν.

4. **Αφαίρεση HTML/Javascript tags.** Επειδή τα δεδομένα προκύπτουν από ιστοσελίδες, περιέχουν αρκετούς χαρακτήρες που αντιπροσωπεύουν κώδικα για την ενσωμάτωσή τους σε αυτές, τα λεγόμενα HTML tags. Η αφαίρεση αυτών αλλά και των υπόλοιπων ειδικών αχρείαστων χαρακτήρων (rubbish data) που προκύπτουν από τη διαδικασία προγραμματισμού των ιστοσελίδων κρίνεται απαραίτητη.
5. **Αφαίρεση ειδικών χαρακτήρων και μεγάλων κενών (tabs).** Όλοι οι ειδικοί χαρακτήρες που δεν αποτελούν κάποια λέξη αφαιρούνται, εκτός από τα βασικά σημεία στίξης. Συγκεκριμένα, το σύνολο των ειδικών χαρακτήρων που αφαιρούνται είναι το εξής : [\^\{}]%__@#$&*-_<>–?:"<>+=,«»…] . Επίσης, αφαιρούνται και τα μεγάλα κενά (tabs), καθώς δυσκολεύουν την επεξεργασία των δεδομένων.
6. **Αφαίρεση μη ελληνικών χαρακτήρων.** Αφαιρούνται όλες οι ξενόγλωσσες λέξεις, όπως και τα Greeklish, καθώς ο σκοπός είναι η υλοποίησης ενός συστήματος αυτόματης περίληψης πάνω σε ελληνικά κείμενα.
7. **Μετατροπή όλων των γραμμάτων σε πεζά.** Πραγματοποιείται η μετατροπή όλων των γραμμάτων σε πεζά για λόγους κανονικοποίησης, και για λόγους αποφυγής ύπαρξης πολλαπλών μορφών ίδιων λέξεων, όπου διαφέρει απλά το αρχικό τους γράμμα, επειδή κάποια βρίσκεται στην αρχή της πρότασης και είναι κεφαλαίο. Μετά τη περάτωση και αυτού του βήματα επεξεργασίας παραμένουν 168858 αξιοποιήσιμα ζεύγη άρθρων-περιλήψεων.
8. **Αντικατάσταση συντμήσεων με την ολική τους μορφή.** Ένα σημαντικό βήμα αποτελεί η αντιμετώπιση της συχνής χρήσης της αποστρόφου στην ελληνική γλώσσα, η οποία είναι πολλές φορές λανθασμένη και δυσχεραίνει τη λειτουργία του μοντέλου, καθώς ουσιαστικά “κόβονται” λέξεις. Έτσι το μοντέλο μαθαίνει πολλαπλές και λανθασμένες αναπαραστάσεις της ίδιας λέξης. Για την αντιμετώπιση αυτού του προβλήματος κάθε λέξη με απόστροφο αντικαθίσταται από την ολόκληρη μορφή της.
9. **Αφαίρεση περιλήψεων μεγαλύτερου μεγέθους από αυτό των άρθρων**. Ένα φαινόμενο που παρουσιάστηκε στο σύνολο δεδομένων είναι η ύπαρξη περιλήψεων μεγαλύτερου μήκους από το αυτό του άρθρου, γεγονός που δεν συνάδει με την κύρια προυπόθεση της περίληψης να αποτελεί μικρότερο κείμενο από το αρχικό και για αυτό αυτά τα ζεύγη αφαιρέθηκαν. Τα ζεύγη στα οποία παρατηρήθηκε το παραπάνω φαινόμενο είναι 22. Σημειώνεται, ότι εξετάστηκε η περίπτωση αντιστροφής της κατηγορίας αυτών των ζευγών, δηλαδή να θεωρηθούν οι περιλήψεις άρθρα και τα άρθρα περιλήψεις, αλλά τα άρθρα ήταν κομμένες προτάσεις και όχι ολοκληρωμένες.
10. **Αφαίρεση περιλήψεων με αυτούσια κομμάτια από τα αντίστοιχα άρθρα**. Ένα άλλο πολύ συχνό φαινόμενο που παρατηρήθηκε, είναι ύπαρξη περιλήψεων που αποτελούν αυτούσιο κομμάτι του αρχικού άρθρου, ο αριθμός αυτών των καταχωρήσεων ήταν αρκετά μεγάλος 89609. Η αφαίρεση αυτών των ζευγών προέκυψε απαραίτητα μετά από δοκιμές του μοντέλου, καθώς παρήγαγε περιλήψεις με μεγάλα αυτούσια κομμάτια από το αρχικό κείμενο.

Μετά από την προ-επεξεργασία που πραγματοποιήθηκε στα δεδομένα, το τελικό πλήθος των διαχειρίσιμων και αξιόπιστων ζευγών άρθρων – περιλήψεων είναι **75229** από τα αρχικά **215385**. Όπως είναι προφανές, το ποσοστό των μη αξιόπιστων δεδομένων είναι αρκετά υψηλό. Παράγοντες εμφάνισης αυτού του γεγονότος αποτελούν η πηγή των δεδομένων, που είναι ιστοσελίδες, όπου πολλές φορές η περίληψη προέρχεται από αυτούσιο κομμάτι του άρθρου και ο τρόπος συλλογής δεδομένων. Ένα μέρος του λογισμικού που πραγματοποιεί την προ-επεξεργασία των δεδομένων παρουσιάζεται στο **Παράρτημα Β**.

Βήμα προ-επεξεργασίας	**Παράδειγμα αφαίρεσης – μετατροπής**
Μορφοποίηση και εισαγωγή δεδομένων	CSV file (title_alt | short_description | description) **->** pandas data structure (title, summary, text)
Αφαίρεση όλων των κενών και δυσνόητων καταχωρήσεων	" ", NaN, uuuuiiuiuuuiuiuiui, 880808909089
Αφαίρεση πολλαπλών καταχωρήσεων	«Η ολοκλήρωση της τετραετίας..» , «Η ολοκλήρωση της τετραετίας..» **->** «Η ολοκλήρωση της τετραετίας..»
Αφαίρεση HTML tags	<html></html> , <head> </head>, , <body></body>, <img> …
Αφαίρεση ειδικών χαρακτήρων και μεγάλων κενών (tabs)	[\^\{}]%__@#$&*-_<>–?:"<>+=,«»…] , " "
Αφαίρεση μη ελληνικών χαρακτήρων	Abcdefghijklmnopqrstuvwxyz
Μετατροπή όλων των γραμμάτων σε πεζά	«Κατά της αναθεώρησης του κανονισμού του Δουβλίνου…» **->** «κατά της αναθεώρησης του κανονισμού του δουβλίνου…»
Αντικατάσταση συντμήσεων με την ολική τους μορφή	γι' αυτό **->** για αυτό, κατ' αναλογία **->** κατά αναλογία, παρ 'όλα **->** παρά όλα, υπ' αριθμόν **->** υπό αριθμόν
Αφαίρεση περιλήψεων μεγαλύτερου μεγέθους από αυτό των άρθρων.	**Άρθρο:** «Η αμετροέπεια εκφράζει την έλλειψη του» **Περίληψη:** «Λόγια χωρίς μέτρο, αδιάκοπη φλυαρία, μεγαλοστομία και υπερβολή είναι η αμετροέπεια, ἀ στερητικό μέτρον και ἔπος τα συνθετικά της»
Αφαίρεση περιλήψεων με αυτούσια κομμάτια από τα αντίστοιχα άρθρα.	**Άρθρο:** «Η συμφωνία κινείται σε θετική κατεύθυνση και στην εθνικός γραμμή όπως χαράχτηκε μετά» **Περίληψη:** «Η συμφωνία κινείται σε θετική κατεύθυνση»

Πίνακας 3.1: *Αναφορά παραδειγμάτων της χρήσεις των βημάτων προ-επεξεργασίας, μέσα από το σύνολο δεδομένων.*

3.3 Η ελληνική γλώσσα

Η ελληνική γλώσσα αποτελεί την αρχαιότερη ευρωπαική γλώσσα και μια από τις αρχαιότερες παγκοσμίως, καθώς έχουν βρεθεί ελληνικές επιγραφές που χρονολογούνται από τη δεύτερη χιλιετία π.Χ. και ελληνικά λογοτεχνικά κείμενα που είναι 2500 ετών [29]. Επίσης, υποστηρίζεται ότι η ελληνική γλώσσα είναι η μοναδική που ομιλείται και γράφεται συνεχώς επί 4000 έτη τουλάχιστον [30]. Βέβαια, με το πέρας τόσων χιλιάδων χρόνων η ελληνική γλώσσα υπέστη πολλές αλλαγές, μέχρι να φτάσει να έχει τη μορφή που έχει σήμερα. Μπορεί η σύγχρονη με την αρχαία ελληνική να μην είναι πανομοιότυπες, αλλά είναι ολοφάνερο ότι η σύγχρονή ελληνική πηγάζει από την αρχαία, καθώς οι περισσότερες ρίζες των λέξεων προέρχονται από αυτήν.

Η επιρροή της ελληνικής γλώσσας στις υπόλοιπες γλώσσες είναι πραγματικά αξιοσημείωτη, με πιο χαρακτηριστικό παράδειγμα αυτό της αγγλικής όπου 50747 από τις περίπου 500000 λέξεις της αγγλικής γλώσσας έχουν ελληνική προέλευση. Το ποσοστό αυτό μεγαλώνει αν εστιάσουμε στις ιατρικές ορολογίες όπου 24862 από τις 46251 αγγλικές λέξεις είναι ελληνικές και αντίστοιχα στις επιστημονικές και τεχνολογικές ορολογίες με 11366 ελληνικές λέξεις στις 25487 [26]. Παρ' όλα αυτά ο συνολικός αριθμός των ανθρώπων που χρησιμοποιεί την ελληνική ως πρώτη ή δεύτερη γλώσσα είναι μόνο 25 εκατομμύρια. Αυτό οφείλεται στο γεγονός ότι τα χαρακτηριστικά που κάνουν την ελληνική γλώσσα τόσο πλούσια και ξεχωριστή, τη καθιστούν ταυτόχρονα δυσκολότερη στη μάθηση συγκριτικά με άλλες γλώσσες όπως στα αγγλικά.

Η υλοποίηση ενός συστήματος αυτόματης παραγωγής περίληψης αποτελεί ένα πρόβλημα που έχει απασχολήσει πολλές φορές την ερευνητική και τεχνολογική κοινότητα. Σχεδόν σε όλες τις περιπτώσεις όμως αφορούσαν την αγγλική γλώσσα και συγκεκριμένα αγγλικά κείμενα, εκτός από κάποιες λιγοστές περιπτώσεις, οι οποίες αναφέρθηκαν στο κεφάλαιο 1. Είναι αξιοσημείωτο, ότι δεν έχει πραγματοποιηθεί ακόμα κάποια υλοποίηση ενός τέτοιου συστήματος εστιασμένο σε ελληνικά κείμενα, για αυτό και ένα αντικείμενο έρευνας της παρούσας εργασίας αποτέλεσε η διερεύνηση των ιδιομορφιών της ελληνικής γλώσσας και των κύριων διαφορών της με την αγγλική. Έπειτα, λαμβάνοντας υπόψιν τις δυσκολίες που προσθέτουν αυτές οι ιδιομορφίες στο μοντέλο, διερευνήθηκαν και προτείνονται τρόποι αντιμετώπισης τους, προσαρμόζοντας έτσι το μοντέλο στην ελληνική γλώσσα.

3.4 Ιδιομορφίες της ελληνικής γλώσσα και αντιμετώπισή τους

Οι προφανείς διαφορές ανάμεσα στην ελληνική και αγγλική γλώσσα αποτελούν το αλφάβητό, και ο τονισμός. Όσον αφορά το αλφάβητο, το ελληνικό αλφάβητο αποτελείται από 24 γράμματα ενώ το αγγλικό από 26, ενώ αρκετά γράμματα του ελληνικού αλφαβήτου δεν υφίστανται στο αγγλικό ή διαφέρει η γραφή τους από τα αντίστοιχα γράμματα του αγγλικό. Με αντίστοιχο τρόπο διαφέρουν τα φωνήεντα, τα σύμφωνα και οι δίφθογγοι της ελληνικής γλώσσας με αυτά της αγγλικής. Όσον αφορά τον τονισμό, στον ελληνικό γραπτό λόγο ο τονισμός της κάθε λέξης εκφράζεται με μια κάθετη παύλα πάνω από την τονιζόμενη συλλαβή, ενώ αυτό δεν υφίστανται στην αγγλική γλώσσα.

Αυτές οι διαφορές μπορεί να είναι βασικές και να ξεχωρίζουν την μια γλώσσα από την άλλη, δεν επηρεάζουν όμως την υλοποίηση και τη λειτουργία του μοντέλου αυτόματης παραγωγής περίληψης. Αυτό συμβαίνει γιατί η κάθε λέξη για να είναι επεξεργάσιμη, μετατρέπεται πριν εισαχθεί στο μοντέλο σε ένα διάνυσμα αριθμητικών τιμών, που ονομάζεται word embedding και θα αναλυθεί περισσότερο σε παρακάτω κεφάλαιο. Έτσι το προς υλοποίηση σύστημα αυτόματης παραγωγής περίληψης, δεν βλέπει λέξεις αλλά αριθμούς, οπότε δεν επηρεάζεται από τη μορφή της κάθε λέξεις αλλά από την εκάστοτε σημασία και χρήση της σε μια πρόταση.

Υπάρχουν όμως αρκετές ιδιαιτερότητες – ιδιομορφίες της ελληνικής γλώσσας, που τη διαφοροποιούν σημαντικά από την αγγλική γλώσσα και επηρεάζουν σημαντικά τη λειτουργία και την αποδοτικότητα του συστήματος. Για αυτό το λόγο, πρέπει να ληφθούν υπόψιν και να αντιμετωπιστούν. Οι ιδιομορφίες αυτές και οι τρόποι αντιμετώπισης που προτείνονται παρουσιάζονται παρακάτω και αφορούν 3 βασικά μέρη της γλώσσας, το λεξιλόγιο, τη γραμματική και το συντακτικό.

3.4.1 Λεξιλόγιο

Ιδιομορφίες λεξιλογίου :

- **Η δυνατότητα παραγωγής νέων σύνθετων λέξεων, πολλαπλασιάζοντας το λεξιλόγιο της.** Στην ελληνική γλώσσα είναι πολύ συχνό φαινόμενο η παραγωγή νεών σύνθετων λέξεων από δυο διαφορετικές λέξεις (συνθετικά). Η συγκεκριμένη διαδικασία είναι αρκετά απλή, καθώς αφαιρείται η κατάληξη του πρώτου συνθετικού, διατηρώντας το θέμα του, και έπειτα συνδέεται με το δεύτερο συνθετικό. Η αντίστοιχη διαδικασία μπορεί να γίνει με παραπάνω από δύο λέξεις. Με τον ίδιο τρόπο προκύπτουν νέες λέξεις προσθέτοντας ένα “α” (στερητικό) και λέξεις όπως “κατά”, “ανά” και “υπό”, μπροστά από μια λέξη.
  π.χ. αλέξω (=απωθώ) + θυμός -> αλεξιθυμία, ανά + λαμβάνω -> αναλαμβάνω
  Με αυτό το τρόπο, το λεξιλόγιο της ελληνική γλώσσας είναι ικανό να αυξάνεται συνεχώς, μειώνοντας την αποδοτικότητα του μοντέλου, καθώς εμφανίζονται όλο και

περισσότερες άγνωστες λέξεις. Το φαινόμενο αυτό δε συναντάται πολύ συχνά στην αγγλική γλώσσα.

- **Συχνή χρήση συγκεκριμενών λέξεων.** Χαρακτηριστικά παραδείγματα συχνά χρησιμοποιημένων λέξεων στην ελληνική γλώσσα αποτελούν το "που", το οποίο αντικαθιστά πολλές φορές και την αναφορική αντωνυμία "ο οποίος" και οι σύνδεσμοι "να" και "και". Επίσης, αρκετές φορές παρατηρείται κατάχρηση των άρθρων, τα οποία βρίσκονται και σε μεγάλη ποικιλία.
  π.χ. "Την επομένη του γάμου συνόδευσαν τη σύζυγο του πατέρα στο σπίτι."
  Η συχνή εμφάνιση συγκεκριμένων λέξεων οδηγεί στη συχνή παραγωγή τους από το μοντέλο αυτόματης παραγωγής περίληψης, καθώς αυτό εκπαιδεύεται στο γεγονός ότι αυτές η λέξεις είναι σημαντικής σημασίας και για αυτό χρησιμοποιούνται συχνά, πράγμα που δεν ισχύει γενικά. Βέβαια αντίστοιχες λέξεις υπάρχουν σε κάθε γλώσσα, όπως και στην αγγλική, αλλά είναι γενικά λιγότερες. Για παράδειγμα, στην αγγλική υπάρχουν δυο βασικά άρθρα "the" και "a/an", ενώ στην ελληνική πολύ περισσότερα.

Αντιμετώπιση:

Όπως ήδη αναφέρθηκε στο προηγούμενο κεφάλαιο, το γνωστό λεξιλόγιο του μοντέλου είναι ένα υποσύνολο του συνολικού λεξιλογίου της ελληνικής γλώσσας, και προέρχεται από το σύνολο των δεδομένων. Η υλοποίηση ενός μοντέλου που να καλύπτει όλο το λεξιλόγιο της ελληνικής γλώσσας, απαιτεί ένα απεριόριστο όγκο δεδομένων λόγω της παραγωγής νέων σύνθετων λέξεων. Παρ΄ όλα αυτά, στη παρούσα εργασία οι άγνωστες λέξεις αντιμετωπίζονται μέσω του μηχανισμού αντιγραφής άγνωστων λέξεων, ο οποίος περιγράφεται αναλυτικά στην υποενότητα 2.3.6.

Για την αντιμετώπιση των συχνά χρησιμοποιημένων λέξεων ενσωματώνεται στο μοντέλο ο αλγόριθμος ποσοτικοποίησης λέξεων TF-IDF (Term Frequency – Inverse Document Frequency). Ο μηχανισμός αυτός χρησιμοποιείται ευρέως σε προβλήματα επεξεργασίας της φυσικής γλώσσας και αυτό που κάνει είναι να υπολογίζει μια τιμή-βάρος για κάθε λέξη, η οποία υποδηλώνει τη σημασία αυτής της λέξης μέσα σε ένα κείμενο. Η τιμή αυτή προκύπτει για κάθε λέξη από τον πολλαπλασιασμό δύο μετρικών :

- **Term frequency:** Η τιμή αυτή υποδηλώνει την συχνότητα εμφάνισης μίας λέξης μέσα σε ένα κείμενο (άρθρο ή περίληψη). Υπολογίζεται πολύ απλά ως το πηλίκο των συνολικών αριθμών εμφάνισης της εκάστοτε λέξης, με τον συνολικό αριθμό λέξεων του κειμένου στο οποίο ανήκει.

$$tf(w,d) = \frac{count\ of\ w\ in\ d}{number\ of\ words\ in\ d} \quad (3.1)$$

  Όπου w: η λέξη που εξετάζεται κάθε φορά και d: το κείμενο.

- **Inverse document frequency:** Η τιμή αυτή εκφράζει πόσο κοινή ή σπάνια είναι η εκάστοτε λέξη μέσα σε όλα τα διαθέσιμα κείμενα, δηλαδή σε όλο το σύνολο δεδομένων. Υπολογίζεται διαιρώντας τον συνολικό αριθμό των κειμένων, με τον αριθμό των

κειμένων που περιέχουν τη λέξη και παίρνοντας τον λογάριθμο αυτού του πηλίκου. Συνεπώς, αν μια λέξη είναι κοινή και εμφανίζεται σε πολλά κείμενα, η τιμή αυτή πλησιάζει στο 0, ενώ αν είναι πιο σπάνια στο 1.

$$idf(w, D, d) = \log\left(\frac{D}{count\ of\ d\ that\ has\ w}\right) \quad (3.2)$$

Όπου w: η λέξη που εξετάζεται κάθε φορά, D: το σύνολο των κειμένων και d: το κείμενο.

Ο τελικός υπολογισμός του TF-IDF προκύπτει ως εξής:

$$tfidf(w, D, d) = tf(w, d) * idf(w, D, d) \quad (3.3)$$

Οι τιμές που προκύπτουν είναι στο διάστημα [0,1] και ενσωματώνονται στο τέλος των word embeddings, τα οποία είναι οι αριθμητικές αναπαραστάσεις των λέξεων, ως διάνυσμα μεγέθους 10.

3.4.2 Γραμματική

Ιδιομορφίες γραμματικής :

- **Η ελληνική γλώσσα είναι κλιτή.** Τα έξι από τα δέκα μέρη του λόγου της ελληνικής γλώσσας (άρθρο, ουσιαστικό, επίθετο, αντωνυμία, ρήμα) κλίνονται, δηλαδή συνθέτουν μορφήματα αλλάζοντας τη κατάληξή τους και διατηρώντας πάντα το θέμα και συνεπώς το βασικό νόημά τους. Αρκεί να αναφέρει κανείς ότι σχεδόν κάθε ουσιαστικό, άρθρο, επίθετο και αντωνυμία έχει τουλάχιστον 6 μορφήματα (γενική, αιτιατική, κλητική ενικού και πληθυντικού αριθμού) που προκύπτουν από τη κλίση τους.
  π.χ. ο άνθρωπος – του ανθρώπου – τον άνθρωπο - άνθρωπε
  Ως αποτέλεσμα, γίνεται δυσκολότερο το έργο του μοντέλου που καλείται μέσω της εκπαίδευσης να αναγνωρίσει τις νοηματικές διαφορές μεταξύ σχεδόν πανομοιότυπων λέξεων. Η αγγλική γλώσσα από την άλλη έχει πιο απλή γραμματική, καθώς κλίνονται μόνο τα ρήματά της. π.χ. η λέξη "human" δεν κλίνεται.
- **Ύπαρξη ελάχιστων λέξεων με πολλαπλή γραμματική χρήση.** Ένα χαρακτηριστικό της ελληνικής γλώσσας, το οποίο μπορεί να θεωρηθεί ως πλεονέκτημά της, είναι η ύπαρξη ελάχιστων λέξεων, οι οποίες ανά περίπτωση λειτουργούν ως διαφορετικά μέρη του λόγου.
  π.χ. που (ο οποίος) -> αναφορική αντωνυμία, που (όπου) -> τοπικό επίρρημα
  πως -> τροπικό επίρρημα, πως (ότι) -> ειδικός σύνδεσμος
  Στην αγγλική γλώσσα οι λέξεις που εμφανίζουν αυτή την ιδιότητα είναι πολύ περισσότερες, καθώς συναντώνται συχνά λέξεις που λειτουργούν χωρίς καμία αλλαγή, σαν ουσιαστικά και σαν ρήματα.
  π.χ. guess (=μαντεύω) -> ρήμα, guess (=μαντεψιά) -> ουσιαστικό
  must (=πρέπει) -> ρήμα, must (=υποχρέωση) -> ουσιαστικό
  see through(=βλέπω μέσα) -> ρήμα, see through(=διάφανος) -> επίθετο

Από τις παραπάνω ιδιομορφίες, όπως και από άλλες πιθανές γραμματικές διαφορές των δύο γλωσσών, μπορεί κανείς να συμπεράνει ότι είναι κρίσιμης σημασίας για το μοντέλο αυτόματης παραγωγής περίληψης να γνωρίζει την γραμματική ταυτότητα κάθε λέξης που επεξεργάζεται.

Αντιμετώπιση:

Για την αντιμετώπιση των παραπάνω ιδιομορφιών, ενσωματώνεται στο μοντέλο ένας μηχανισμός που ονομάζεται POS (Part of Speech) tagging. Συγκεκριμένα, μέσω του μηχανισμού αυτού τίθεται σε κάθε λέξη μια ετικέτα (tag) που εκφράζει την γραμματική της υπόσταση και λειτουργία, δηλαδή τι μέρος του λόγου είναι. Αρχικά, η κάθε πρόταση, χωρίζεται στις λέξεις που την αποτελούν (tokenization), και έπειτα σε κάθε μια λέξη ορίζεται η ετικέτα που της αντιστοιχεί. Η διαδικασία της κατηγοριοποίησης πραγματοποιείται χρησιμοποιώντας τη βιβλιοθήκη Spacy της Python, η οποία έχει εκπαιδευτεί με λέξεις από πολλές γλώσσες και μια από αυτές είναι η ελληνική. Περαιτέρω ανάλυση της συγκεκριμένης βιβλιοθήκης ακολουθεί στο κεφάλαιο 4.

Αφού έχει κατηγοριοποιηθεί η κάθε λέξη με βάση τη γραμματική της χρήση, αυτή η πληροφορία πρέπει να περάσει με κάποιο τρόπο στο σύστημα αυτόματης παραγωγής περίληψης. Αυτό πραγματοποιείται αναπαριστάνοντας τη κάθε ετικέτα με ένα διάνυσμα συγκεκριμένων ίδιων τιμών, το οποίο ενσωματώνεται στο τέλος της αριθμητικής αναπαράστασης (word embedding) της κάθε λέξης. Στον παρακάτω πίνακα παρουσιάζονται όλες οι πιθανές ετικέτες.

Ετικέτα (Tag)	**Γραμματικός τύπος**
ADJ	adjective = επίθετο
ADP	αdposition = πρόθεμα
ADV	adverb = επίρρημα
AUX	auxiliary = βοηθητικό ρήμα
CCONJ	coordinating conjunction = συντονιστικός σύνδεσμος
DET	Determiner = άρθρο
NOUN	noun = ουσιαστικό
NUM	numeral = αριθμός
PRON	pronoun = αντωνυμία
PROPN	proper noun = κύριο ουσιαστικό
PUNCT	punctuation = σημείο στίξης
SCONJ	subordinating conjunction = δευτερεύων σύνδεσμος
SYM	symbol = σύμβολο
VERB	verb = ρήμα

Πίνακας 3.2: *Ετικέτες και γραμματικοί τύποι.*

3.4.4 Συντακτικό

Ιδιομορφίες συντακτικού:

- **Μεγάλη συντακτική ευκαμψία-ελαστικότητα.** Στην ελληνική γλώσσα δεν υπάρχει κάποια υποχρεωτική συντακτική σειρά των λέξεων μέσα σε μια πρόταση. Επομένως, το ίδιο ακριβώς μήνυμα μπορεί να εκφραστεί με πολλούς διαφορετικούς τρόπους, αλλάζοντας τις θέσεις των λέξεων στη πρόταση.
  π.χ. "Ο Γιώργος αγόρασε καινούριο αυτοκίνητο." -> "Αγόρασε καινούριο αυτοκίνητο ο Γιώργος." -> "Καινούριο αυτοκίνητο αγόρασε ο Γιώργος."
  Στην αγγλική γλώσσα αυτές οι αλλαγές δε μπορούν να επιτελεστούν, καθώς μεταβάλλεται το νοηματικό περιεχόμενο της πρότασης. Έτσι, η συντακτική σειρά υποκείμενο-ρήμα-αντικείμενο (Y-P-A) είναι σχεδόν υποχρεωτική σε μία πρόταση.
  π.χ. "George bought a new car" -> "Bought George a new car" -> "A new car bought George".
  Όπως φαίνεται, το νόημα της πρότασης αλλάζει εντελώς από το "Ο Γιώργος αγόρασε καινούριο αυτοκίνητο." στο " Ένα καινούριο αυτοκίνητο αγόρασε τον Γιώργο."
- **Μη χρήση προσωπικής αντωνυμίας.** Αρκετές φορές στην ελληνική γλώσσα παραλείπεται η προσωπική αντωνυμία, καθώς εννοείται από τα συμφραζόμενα.
  π.χ. "Εγώ πηγαίνω στη παραλία σήμερα." -> "Πηγαίνω στη παραλία σήμερα."
  Στην αγγλική γλώσσα αυτό δεν είναι εφικτό στις περισσότερες περιπτώσεις, καθώς πάλι μεταβάλλεται το νόημα της πρότασης.
  π.χ. "I go to the beach today" -> "Go to the beach today."
  Όπως φαίνεται, το νόημα της πρότασης αλλάζει από το "Εγώ πηγαίνω στη παραλία σήμερα." στο "Πήγαινε στη παραλία σήμερα."

Από τις παραπάνω ιδιομορφίες, όπως και από άλλες πιθανές συντακτικές διαφορές των δύο γλωσσών, μπορεί κανείς να συμπεράνει ότι είναι κρίσιμης σημασίας για το μοντέλο αυτόματης παραγωγής περίληψης να γνωρίζει το συντακτικό ρόλο της κάθε λέξης σε μία πρόταση που επεξεργάζεται.

Αντιμετώπιση:

Για την αντιμετώπιση των παραπάνω ιδιομορφιών, ενσωματώνεται στο μοντέλο ένας μηχανισμός που ονομάζεται DEP (Dependency) tagging. Συγκεκριμένα, μέσω του μηχανισμού αυτού τίθεται σε κάθε λέξη μια ετικέτα (tag) που εκφράζει το συντακτικό της τύπο της υπόσταση και λειτουργία, δηλαδή το ρόλο της μέσα στη πρόταση. Η διαδικασία που ακολουθείται είναι παρόμοια με αυτή του POS tagging, όπου αρχικά διαχωρίζονται οι λέξεις της κάθε πρότασης και στη συνέχεια κατηγοριοποιούνται με τη χρήση της βιβλιοθήκης Spacy.

Αντιστοίχως, στο τέλος οι ετικέτες ενσωματώνονται στα word embeddings για να μπορεί το σύστημα να προσπελάσει και αυτή τη πληροφορία. Στον παρακάτω πίνακα παρουσιάζονται όλες οι πιθανές ετικέτες.

Ετικέτα (Tag)	**Συντακτικός Τύπος**
acl	clausal modifier = αναφορικός πρόσδιορισμός
advmod	adverbial modifier = επιρρηματικός προσδιορισμός
amod	adjectival modifier = επιθετικός προσδιορισμός
appos	appositional modifier = παράθεση
aux	auxiliary = βοηθητικό ρήμα
cc, cccomp	coordinating conjunction = συντονιστικός σύνδεσμος
conj	conjunct = συνδεσμος
cop	copula = συνδετικό ρημα
csubj	clausal subject = υποκείμενο πρότασης
dep	unclassified dependent = μη κατηγοριοποιημένος τύπος
det	determiner = προσδιορισμός
iobj	indirect object = έμμεσο αντικείμενο
nmod	modifier of nominal = ονοματικός προσδιορισμός
nsubj	nominal subject= υποκείμενο κύριας πρότασης
nummod	numeric modifier = αριθμητικό επίθετο
obj	object = αντικείμενο
parataxis	parataxis = παράταξη
punct	punctuation = σημεία στίξης
xcomp	clausal complement = κατηγορούμενο

Πίνακα 3.3: *Ετικέτες και συντακτικοί τύποι.*

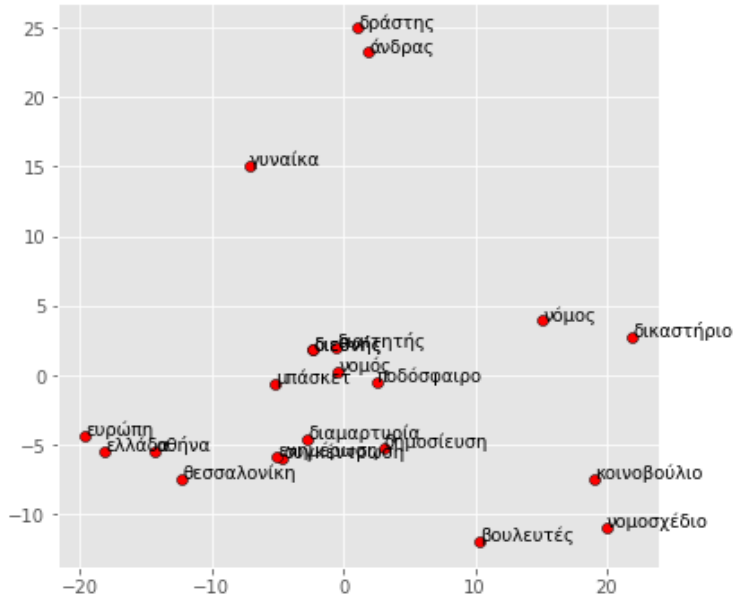
3.5 Αριθμητικές αναπαραστάσεις λέξεων (Word Embeddings)

Τα μοντέλα μηχανικής και βαθιάς μάθησης χρησιμοποιούν αλγορίθμους και μηχανισμούς για να προσεγγίσουν μια μαθηματική συνάρτηση, ικανή να παράγει τις επιθυμητές εξόδους από την εκάστοτε είσοδο. Αυτή η διαδικασία υλοποιείται πραγματοποιώντας πολλαπλούς αριθμητικούς υπολογισμούς κατά τη διάρκεια της εκπαίδευσης του μοντέλου. Συνεπώς, είναι απαραίτητο οι λέξεις να μετατραπούν σε αριθμητικές αναπαραστάσεις, οι οποίες ονομάζονται word embeddings, για να μπορούν να είναι προσπελάσιμες από το μοντέλο. Η τελική μορφή ενός word embedding είναι ένα διάνυσμα με ορισμένο μέγεθος, όπου κάθε του στοιχείο είναι μια τιμή - βάρος που εκφράζει το σημασιολογικό και νοηματικό περιεχόμενο της λέξης.

Τα word embeddings βρίσκουν εφαρμογή στα περισσότερα προβλήματα της επιστήμης της φυσικής επεξεργασίας της γλώσσας, έχοντας σημαντικό αντίκτυπο. Η ικανότητά τους να αναπαριστούν τις σημασιολογικές και νοηματικές σχέσεις μεταξύ των λέξεων, σε συνδυασμό με τη χρήση συνελικτικών (CNN) και LSTM νευρωνικών δικτύων έχουν οδηγήσει σε αξιοσημείωτα αποτελέσματα σε προβλήματα όπως αυτό της κατηγοριοποίησης κειμένου [31], αυτόματης παραγωγής μετάφρασης και περίληψης [32] . Η διαδικασία μετατροπής των λέξεων σε αριθμητικές αναπαραστάσεις είναι αρκετά περίπλοκη, καθώς πρέπει τα πολλαπλά νοήματα και οι σημασιολογικές σχέσεις της κάθε λέξης να ληφθούν υπόψιν, για να εκφράζεται στο τέλος ορθά από την αναπαράστασή της. Για αυτό το λόγο, στις περισσότερες περιπτώσεις τα word embeddings προκύπτουν από εκπαιδευόμενα νευρωνικά δίκτυα.

Με το ίδιο τρόπο πραγματοποιείται η παραγωγή των word embeddings στη παρούσα εργασία, όπου με τη χρήση της βιβλιοθήκης Gensim παράγονται τα word embeddings κάθε λέξης από όλα τα διαθέσιμα κείμενα (άρθρα-περιλήψεις) στο διαθέσιμο σύνολο δεδομένων. Συγκεκριμένα, η μέθοδος που χρησιμοποιεί η βιβλιοθήκη Genism για την παραγωγή των word embeddings αποτελείται από Skip-gram και Continuous Bag of Words (CBOW) μοντέλα. Τα μοντέλα αυτά εκπαιδεύονται για περίπου 2 ώρες με τα διαθέσιμα κείμενα ως δεδομένα εκπαίδευσης, για να παράγουν word embeddings μεγέθους 120 στοιχείων. Στη συνέχεια, στο τέλος αυτών των διανυσμάτων προστίθενται τα 3 διανύσματα (TF-IDF, POS tags, DEP tags) μεγέθους 10 στοιχείων, που αναλύθηκαν στο προηγούμενο κεφάλαιο. Στο τέλος, κάθε λέξη αντιστοιχίζεται στο μοντέλο με τη διανυσματική αναπαράστασή της μεγέθους 150.

Στο παρακάτω γράφημα απεικονίζονται κάποια παραδείγματα κοντινότερων συνόλων λέξεων με βάση τα word embeddings τους. Όσο πιο κοντά βρίσκονται οι λέξεις τόσο πιο πολύ συνδέονται νοηματικά.



Εικόνα 3.1: *Αναπαράσταση κοντινών συνόλων λέξεων, βάσει των word embeddings.*

Από το γράφημα παρατηρείται ότι οι λέξεις [ευρώπη, ελλάδα, αθήνα, θεσσαλονίκη] είναι κοντά μεταξύ τους, άρα τα word embeddings τους μοιάζουν, πράγμα που υποδηλώνει ότι συνδέονται νοηματικά. Το αντίστοιχο μπορεί να σημειωθεί για τα σύνολα [διαμαρτυρία, συγκέντρωση, ενημέρωση] και [κοινοβούλιο, βουλευτές, νομοσχέδιο]. Ένα μέρος του λογισμικού που πραγματοποιεί την προ-επεξεργασία των δεδομένων παρουσιάζεται στο **Παράρτημα Β.**

4. Δοκιμές και Αξιολόγηση

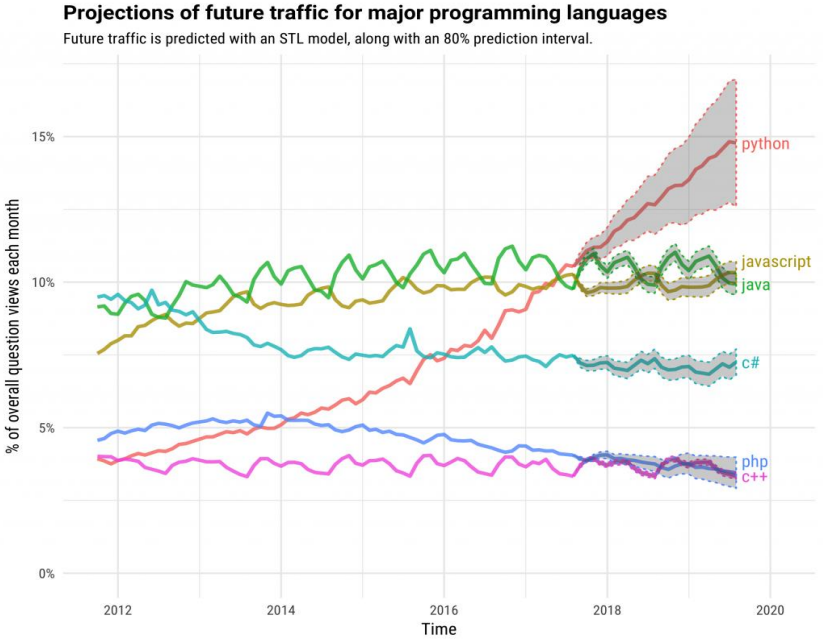
4.1 Εργαλεία Προγραμματισμού

Σε αυτό το κεφάλαιο θα γίνει αναφορά στα εργαλεία που χρησιμοποιήθηκαν για τον προγραμματισμό και την υλοποίηση του λογισμικού του μοντέλου, που αναπτύχθηκε στα πλαίσια της εργασίας. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την ανάπτυξη του μοντέλου είναι η python και οι λόγοι για αυτή την επιλογή αναφέρονται παρακάτω.

Python

Η python είναι μια υψηλού επιπέδου, αντικειμενοστραφής γλώσσα προγραμματισμού με δυναμική σημασιολογία [33]. Επίσης, έχει την ικανότητα να χρησιμοποιεί συναρτήσεις-μεθόδους (functions) και διαδικασίες (procedures), γεγονός που τι κάνει μια πολύ ευέλικτη γλώσσα και κατάλληλη για γρήγορη ανάπτυξη λογισμικού. Η απλή και εύκολη στη χρήση σύνταξη της python την καθιστά μια από τις ευκολότερες στην εκμάθηση γλώσσες προγραμματισμού και της προσδίδει υψηλή αναγνωσιμότητα, μειώνοντας το κόστος συντήρησης του εκάστοτε προγράμματος.

Η γλώσσα προγραμματισμού python χρησιμοποιείται σε ένα τεράστιο φάσμα εφαρμογών που περιέχουν κάποια ανάπτυξη λογισμικού, από ανάπτυξη ιστοσελίδων μέχρι προγραμματισμό μικρο-ελεγκτών. Ένας μεγάλος τομέας στον οποίο εφαρμόζεται η python είναι η μηχανική μάθηση και οι εφαρμογές της. Συγκεκριμένα, αποτελεί τη δημοφιλέστερη γλώσσα προγραμματισμού στην ανάπτυξη εφαρμογών μηχανικής και βαθιάς μάθησης, αλλά και γενικότερα.



Εικόνα 4.1: *Η ραγδαία αύξηση της χρήσης της python [34].*

Ένας βασικό λόγος που καθιστά τη python τη δημοφιλέστερη και καταλληλότερη γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών μηχανική και βαθιάς μάθησης είναι η ύπαρξη πολλών βιβλιοθηκών που περιέχουν έτοιμα προσαρμοζόμενα μοντέλα και μηχανισμούς, τα οποία μπορούν να ενσωματωθούν δυναμικά σε μία εφαρμογή. Στη παρούσα εργασία χρησιμοποιήθηκε μια πληθώρα βιβλιοθηκών, οι οποίες περιγράφονται πολύ συνοπτικά παρακάτω.

Οι κύριες βιβλιοθήκες της python που χρησιμοποιήθηκαν για την ανάπτυξη του μοντέλου είναι:

- **TensorFlow:** Είναι μια opensource βιβλιοθήκη που χρησιμοποιείται για την υλοποίηση διάφορων εφαρμογών της μηχανικής μάθησης. Επικεντρώνεται όμως κυρίως στην σχεδίαση, εκπαίδευση και στη δοκιμή βαθιών νευρωνικών δικτυών. Στη παρούσα εργασία η βιβλιοθήκη αυτή αποτέλεσε το βασικό εργαλείο υλοποίησης του μοντέλου, καθώς όλα τα επιμέρους στοιχεία του κορμού του (κωδικοποιητής, αποκωδικοποιητής, μηχανισμός προσοχής κ.τ.λ.), προγραμματίστηκαν με βάση αυτή. Το ίδιο έγινε και με την εκπαίδευση και δοκιμή του μοντέλου.
- **Pandas:** Αποτελεί ένα γρήγορο, ευέλικτο και opensource εργαλείο που χρησιμοποιείται για την ανάλυση, διαχείριση και επεξεργασίας δεδομένων. Οι δυνατότητες που προσφέρει είναι πάρα πολλές καθώς υποστηρίζει πολλών ειδών δεδομένων, είτε είναι αριθμητικά είτε κείμενα, με διαφορετικό format όπως txt,csv, HDF5 αρχεία και SQL βάσεις δεδομένων. Η pandas είναι επίσης μια βιβλιοθήκη που χρησιμοποιήθηκε αρκετά στη παρούσα εργασία, τόσο στο αρχικό στάδιο προ-επεξεργασίας των δεδομένων, όσο και μετέπειτα όπου χρειαζόταν.
- **NumPy (Numerical Python):** Είναι μια βιβλιοθήκη ιδανική για διαχείριση και επεξεργασίας μαθηματικών πινάκων. Υποστηρίζει πίνακες πολλαπλών διαστάσεων και πράξεις γραμμικής άλγεβρας όπως η ένωση, ο πολλαπλασιασμός, η πρόσθεση και η συνέλιξη πινάκων, αλλά και στατιστικές πράξεις (μέση τιμή, διακύμανση, συσχέτιση κ.α.). Χαρακτηριστικά παραδείγματα χρήσης της βιβλιοθήκης στη παρούσα εργασία είναι, η προσάρτηση των επιπλέον χαρακτηριστικών, που αναλύθηκαν στην προηγούμενη ενότητα, στα word embeddings και το τελικό στάδιο επιλογής των αποκωδικοποιημένων αποτελεσμάτων (beam search).
- **SpaCy:** Αποτελεί μια βιβλιοθήκη ειδικά σχεδιασμένη για την παραγωγή εφαρμογών που επεξεργάζονται κείμενα. Περιέχει συναρτήσεις και μοντέλα που εφαρμόζονται είτε για την προ-επεξεργασία, είτε για την εξαγωγή χαρακτηριστικών μέσα από κείμενα. Στη παρούσα εργασία η βιβλιοθήκη spacy χρησιμοποιείται για την αντιμετώπιση των ιδιομορφιών της ελληνικής γλώσσας και συγκεκριμένα την εξαγωγή των ετικετών POS και DEP tagging από τα ελληνικά κείμενα.
- **NLTK (Natural Language Toolkit):** Είναι ένα εργαλείο που στοχεύει στην ανάπτυξη προγραμμάτων που επεξεργάζονται ανθρώπινα κείμενα. Περιέχει βιβλιοθήκες που εφαρμόζονται στις περισσότερες εφαρμογές της επεξεργασίας της φυσικής γλώσσας (NLP), όπως η προ-επεξεργασία, η ταξινόμηση και η κατάτμηση κειμένων και λέξεων.

Αποτελεί τη πλουσιότερη πλατφόρμα λογισμικού στο τομέα της φυσικής επεξεργασίας γλώσσας, καθώς περιέχει πάνω από 50 εκπαιδευμένα μοντέλα. Στη παρούσα εργασία χρησιμοποιείται στην προ-επεξεργασία των δεδομένων, στο χωρισμό τους σε υποσύνολα training και testing, στη παραγωγή του λεξιλογίου και τον υπολογισμό των TF-IDF τιμών.

- **GenSim:** Αποτελεί επίσης μια opensource βιβλιοθήκη που περιέχει μοντέλα για την επεξεργασία της φυσικής γλώσσας και συγκεκριμένα μη εποπτευόμενα μοντέλα, όπως είναι και αυτά που παράγουν τις αριθμητικές αναπαραστάσεις των λέξεων. Για αυτό το λόγο, η βιβλιοθήκη αυτή χρησιμοποιείται για την παραγωγή των word embeddings από τα ελληνικά κείμενα.
- **Rouge:** Χρησιμοποιείται για τον υπολογισμό των μετρικών Rouge που χρησιμοποιούνται για την αξιολόγηση των παραγόμενων περιλήψεων από το μοντέλο. Οι μετρικές περιγράφονται αναλυτικότερα σε παρακάτω κεφάλαιο.

4.2 Μετρικές αξιολόγησης ROUGE

Για τη μέτρηση της επίδοσης των μοντέλων που σχεδιάζονται στη συνέχεια, έγινε χρήση των μετρικών ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Οι συγκεκριμένες μετρικές είναι αρκετά διαδεδομένες στα προβλήματα επεξεργασίας φυσικής γλώσσας, και χρησιμοποιούνται συχνά ως μέτρο σύγκρισης συστημάτων αυτόματης παραγωγής περίληψης. Οι μετρικές ROUGE χρησιμοποιούν διάφορους αλγορίθμους για να εντοπίσουν ομοιότητες μεταξύ προτάσεων. Όσο περισσότερες ομοιότητες έχουν οι προτάσεις που συγκρίνονται, τόσο μεγαλύτερο είναι το σκορ που συλλέγουν οι μετρικές.

Στα πλαίσια αυτή της εργασίας, συγκρίνονται οι παραγόμενες περιλήψεις με τις “ιδανικές” περιλήψεις αναφοράς του διαθέσιμου συνόλου δεδομένων. Στόχος είναι η βαθμολόγηση κάθε παραγόμενης περίληψης ως προς την αντίστοιχη περίληψη του συνόλου δεδομένων, η οποία ορίζεται ως η ορθή. Ο όρος ορθή περίληψη είναι λίγο αμφιλεγόμενος και δύσκολα μπορεί να οριστεί μαθηματικά σωστά, ώστε η μετρική επίδοσης να αποτελεί μια αντικειμενική συνάρτηση επίδοσης για τις παραγόμενες περιλήψεις. Καθώς όμως τα μοντέλα έχουν εκπαιδευτεί σε παρόμοια δεδομένα με αυτά των συνόλων μέτρησης της επίδοσης, θεωρείται ότι προσπαθούν να προσεγγίσουν περιλήψεις που μοιάζουν στο ύφος με τις περιλήψεις σύγκρισης, οπότε θεωρούνται αυτές ιδανικές.

Όλοι οι διαφορετικοί αλγόριθμοι Rouge που θα παρουσιαστούν σε αυτήν την υποενότητα, έχουν ως στόχο να μοντελοποιήσουν την σύγκριση των περιλήψεων που απαιτείται, με μαθηματικό τρόπο. Στόχος όλων των αλγορίθμων ROUGE που παρουσιάζονται σε αυτή την υποενότητα είναι να μοντελοποιήσουν με κάποιο μαθηματικό τρόπο την απαιτούμενη σύγκριση των περιλήψεων. Για αυτό εξετάζουν πόσες διαφορετικές/όμοιες υποακολουθίες των περιλήψεων εμφανίζονται και στις δύο περιλήψεις. Στη συγκεκριμένη εργασία,

χρησιμοποιούνται 3 αλγόριθμοι ROUGE για σύγκριση των περιλήψεων, και παρουσιάζονται παρακάτω.

- **ROUGE-1:** Ο αλγόριθμος ROUGE-1 προσπαθεί να μοντελοποιήσει μαθηματικά την ομοιότητα δύο περιλήψεων, μετρώντας πόσες μεμονωμένες λέξεις (unigrams) εμφανίζονται και στις δύο περιλήψεις. Με αυτό το τρόπο υπολογίζεται έμμεσα το πόσο κοντά είναι νοηματικά οι δύο εξεταζόμενες περιλήψεις
- **Αλγόριθμος ROUGE-2:** Ο αλγόριθμος ROUGE-2 κινείται στα ίδια πλαίσια, αλλά μετράει τον αριθμό των ζευγών λέξεων (bigrams) που είναι κοινές στις εξεταζόμενες περιλήψεις. Μέσω αυτής της μετρικής, ελέγχεται σε κάποιο βαθμό η συντακτική ορθότητα της εξαγόμενης περίληψης του εκάστοτε μοντέλου, δίνοντας παραπάνω έμφαση στους συντακτικούς κανόνες.
- **Αλγόριθμος ROUGE-L:** Ο αλγόριθμος ROUGE-L ακολουθεί παρόμοια φιλοσοφία και προσπαθεί να εντοπίσει την πιο μεγάλη κοινή υποακολουθία λέξεων (longest common subsequence).

Όλοι αυτοί οι αλγόριθμοι υπολογίζουν τις ομοιότητες των ακολουθιών σύμφωνα με τους κανόνες που περιγράφηκαν παραπάνω, και πραγματοποιούν μετρήσεις στις εξής μετρικές:

- **Ανάκληση (Recall):** Η μετρική της ανάκλησης, υπολογίζεται από το πλήθος των κοινών ακολουθιών που βρέθηκαν σύμφωνα με τον κάθε αλγόριθμο ROUGE, δια το συνολικό μήκος της ”ορθής” περίληψης. Ουσιαστικά αυτή η μετρική μετράει πόσο μέρος της ”ορθής” περίληψης που βρίσκεται στο σύνολο δεδομένων, εμφανίζεται και στην παραγόμενη από το μοντέλο περίληψη.

$$Recall = \frac{number\ of\ overlaping\ sequence\ tokens}{total\ tokens\ in\ reference\ summary} \quad (4.1)$$

- **Ακρίβεια (Precision):** Η μετρική της ακρίβειας υπολογίζεται με αντίστοιχο τρόπο αλλά αντί για το συνολικό μήκος της περίληψης αναφοράς χρησιμοποιεί το μήκος της παραγόμενης περίληψης. Με αυτό το τρόπο υπολογίζει το πλήθος των κοινών ακολουθιών των περιλήψεων, δια το συνολικό μήκος της προβλεπόμενης περίληψης.

$$Precision = \frac{number\ of\ overlaping\ sequence\ tokens}{total\ tokens\ in\ generated\ summary} \quad (4.2)$$

- **Μετρική F1 (F1 score):** Η μετρική F1 προσπαθεί να μοντελοποιήσει τις δύο προηγούμενες μετρικές σε μια κοινή, ώστε να μεγιστοποιείται όταν μεγιστοποιούνται και οι άλλες. Αποτελεί πιο πολύ ένας μαθηματικός συνδυασμός τους χωρίς εμφανές πρακτικό νόημα. Χρησιμοποιείται παρόλα αυτά πολύ συχνά καθώς καταφέρνει να αναπαριστά και τις δύο προηγούμενες μετρικές σε έναν τελικό αριθμό.

$$F1score = \frac{Precision*Recall}{Precision+Recall} \quad (4.3)$$

Σημειώνεται ότι όλες οι παραπάνω μετρικές βαθμολογούν σε κλίμακα 0-1 (ή αντίστοιχα σε ποσοστιαία κλίμακα 0-100%), όπου σκορ ίσο με 1 υποδηλώνει πως οι περιλήψεις που

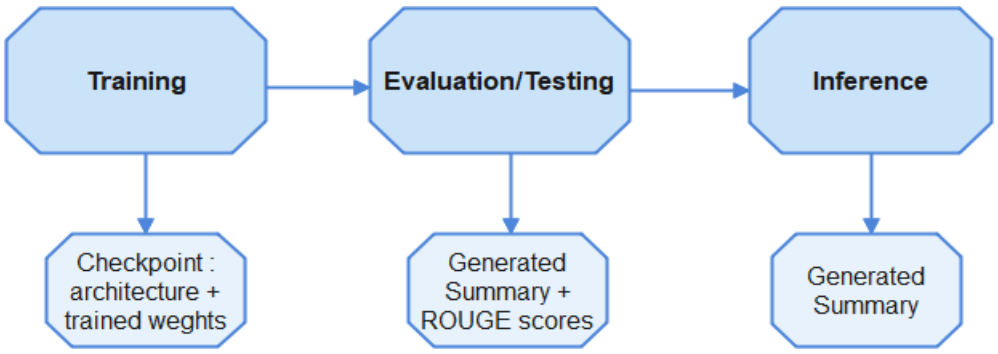
συγκρίθηκαν είναι πανομοιότυπες ως προς την εκάστοτε μετρική, ενώ όταν είναι 0 δηλώνει ότι είναι διαφορετικές. Όταν η μετρική F1 είναι 1, αυτό σημαίνει πως και η ακρίβεια και η ανάκληση έχουν σκορ 1, και υποδηλώνει πως οι περιλήψεις είναι ολόιδιες μεταξύ τους.

4.3 Περιγραφή διαδικασιών υλοποίησης του μοντέλου

Σε αυτό το κεφάλαιο πραγματοποιείται μια περιγραφή του λογισμικού που χρησιμοποιήθηκε για την υλοποίηση του συστήματος αυτόματης παραγωγής περίληψης. Συγκεκριμένα, περιγράφονται σειριακά τα στάδια που ακολουθήθηκαν για την υλοποίηση και δοκιμή του συστήματος.

Αναφέρεται ότι το υλοποιήσιμο σύστημα αποτελεί κατά βάση ένα μοντέλο μηχανικής μάθησης και συγκεκριμένα βαθιάς μάθησης, οπότε τα κυρίως στάδια υλοποίησης του είναι αυτά που ακολουθούνται στις περισσότερες περιπτώσεις συστημάτων μηχανικής μάθησης :

1) **Training** (Εκπαίδευση συστήματος)
2) **Evaluation/Testing** (Αξιολόγηση/Δοκιμή συστήματος)
3) **Inference** (Δοκιμή συστήματος σε άγνωστα δεδομένα)



Εικόνα 4.2: *Απεικόνιση των τριών βασικών σταδίων υλοποίησης και των εξόδων τους.*

Τα τρία αυτά στάδια περιγράφονται στα επόμενα υπο-κεφάλαια. Πρέπει πρώτα όμως να γίνει αναφορά σε ένα στάδιο που είναι απαραίτητο για την υλοποίησης ενός μοντέλου αυτόματης παραγωγής περίληψης και είναι αυτό της δημιουργίας του λεξιλογίου.

4.3.1 Χωρισμός δεδομένων – Δημιουργία Λεξιλογίου

Μετά την ολοκλήρωση της προ-επεξεργασίας και καθαρισμού των δεδομένων που αναλύθηκε στο κεφάλαιο 4.2, ακολουθεί ο χωρισμός των διαθέσιμων δεδομένων σε δεδομένα εκπαίδευσης και αξιολόγησης και η δημιουργία του λεξιλογίου. Το σύνολο των διαθέσιμων δεδομένων, που αποτελείται συνολικά από **75468** ελληνικά ζεύγη κειμένων (άρθρα-περιλήψεις), χωρίζεται ως εξής:

- **Training data :** 73468 (97.3% των συνολικών δεδομένων)
- **Evaluation\Testing data:** 2000 (2.7% των συνολικών δεδομένων)

Ακολουθώντας την πολιτική που εφαρμόζεται συνήθως σε μοντέλα μηχανικής μάθησης.

Έπειτα ακολουθεί η δημιουργία του λεξιλογίου του μοντέλου. Το λεξιλόγιο αποτελεί ένα σημαντικό κομμάτι ενός μοντέλου αυτόματης παραγωγής περίληψης και πρέπει να είναι αυστηρά ορισμένο, καθώς μέσα από αυτό επιλέγονται οι λέξεις που θα αποτελούν τη παραγόμενη περίληψη κάθε φορά. Είναι πολύ σημαντικό, το παραγόμενο λεξιλόγιο να αποτελείται από λέξεις που προκύπτουν από τα δεδομένα εκπαίδευσης και όχι από τα κείμενα αξιολόγησης. Αυτό γιατί, ο σκοπό της αξιολόγησης είναι να τεστάρει την επίδοση του μοντέλου σε πραγματικές συνθήκες , οπότε τα δεδομένα αξιολόγησης πρέπει να είναι άγνωστα.

Το μήκος του λεξιλογίου αποτελεί μία σημαντική υπερπαράμετρος του συστήματος, η οποία επηρεάζει τόσο την επίδοση του συστήματος, όσο και την εκπαίδευσή του. Συγκεκριμένα, όσο μεγαλύτερο είναι το λεξιλόγιο του συστήματος, τόσο πιο γενικευμένο γίνεται, καθώς γνωρίζει περισσότερες λέξεις τις οποίες μπορεί να χρησιμοποιήσει για την παραγωγή περιλήψεων διάφορων θεμάτων. Από την άλλη όμως, για να δημιουργηθεί ένα τεράστιο λεξιλόγιο χρειάζεται ένα τεράστιο όγκο δεδομένων-κειμένων και παράλληλα όσο μεγαλώνει αυτό το λεξιλόγιο, αυξάνεται ο απαιτούμενος χρόνος εκπαίδευσης του συστήματος. Το μέγεθος του λεξιλογίου που προκύπτει από τα διαθέσιμα κείμενα στη παρούσα εργασία είναι **100000** διαφορετικές λέξεις. Μέσω του λογισμικού που υλοποιήθηκε για την ανάπτυξη του μοντέλου, δίνεται η δυνατότητα στο χρήστη να καθορίσει το μέγεθος του λεξιλογίου (vocab_size), ορίζοντας τιμη0-100000. Αν η ορισμένη τιμή είναι 0 ή μεγαλύτερη του 100000, τότε το μέγεθος ορίζεται ως το μέγιστο (100000). Πραγματοποιήθηκαν δοκιμές του μοντέλου για λεξιλόγιο μεγέθους 50000 και 100000 οι οποίες παρουσιάζονται στο παρακάτω κεφάλαιο.

4.3.2 Διαδικασία εκπαίδευσης (Training)

Η εκπαίδευση ενός συστήματος βαθιάς μάθησης αποτελεί η διαδικασία κατά την οποία οι τιμές-βάρη του βαθύ νευρωνικού δικτύου προσαρμόζονται με σκοπό να παράγουν στην έξοδο ένα αποτέλεσμα που προσεγγίζει το ζητούμενο, το οποίο καθορίζεται από τα δεδομένα εκπαίδευσης. Στη περίπτωση μας, το μοντέλο εκπαιδεύεται να παράγει περιλήψεις που να προσεγγίζουν τις περιλήψεις αναφοράς των κειμένων εκπαίδευσης. Έτσι αφού μάθει να αντιμετωπίζει αποτελεσματικά τα κείμενα εκπαίδευσης, μπορεί να αντιμετωπίσει σε ένα ικανοποιητικό βαθμό και άγνωστα κείμενα.

Κατά τη διάρκεια της εκπαίδευσης, οι νέες τιμές των βαρών των νευρωνικών δικτύων (κωδικοποιητή και αποκωδικοποιητή) υπολογίζονται σε κάθε βήμα με βάση τις αποκλίσεις της παραγόμενης περίληψης από την επιθυμητή. Ο υπολογισμός αυτών των αποκλίσεων πραγματοποιείται μέσω μιας συνάρτησης κόστους (loss function). Στη παρούσα εργασία η συνάρτηση κόστους που επιλέχθηκε είναι η διαδεδομένη συνάρτηση απώλειας εντροπίας (crossentropy loss function), η οποία εκφράζεται ως εξής:

$$J = -\frac{1}{N}(\sum yt * \log(yp) \tag{4.4}$$

όπου yt είναι το διάνυσμα με τις επιθυμητές τιμές εξόδου και yp το διάνυσμα με τις προβλέψεις-παραγόμενες τιμές εξόδου από το μοντέλο σε κάθε βήμα.

Η συνάρτηση αυτή εφαρμόζεται ευρέως σε προβλήματα κατάταξης με νευρωνικά δίκτυα και επιλέχθηκε στη προκειμένη περίπτωση, γιατί το τελικό στάδιο επιλογής των παραγόμενων λέξεων από το μοντέλο αυτόματης παραγωγής περίληψης ανάγεται σε πρόβλημα κατάταξης. Συγκεκριμένα, σε κάθε βήμα παραγωγής μιας λέξης στην έξοδο του μοντέλου, η έξοδος αυτή κατατάσσεται σε τόσες κατηγορίες όσες είναι οι λέξεις του λεξιλογίου, για να ακολουθήσει η τελική επιλογή της λέξης-κατάταξης μέσω της ακτινικής αναζήτησης.

Μετά τον υπολογισμό του κόστους απόκλισης σε κάθε βήμα της εκπαίδευσης, ακολουθεί η ενημέρωση των βαρών και μεταβλητών του μοντέλου. Αυτό πραγματοποιείται μέσω ενός βελτιστοποιητή (optimizer), ο οποίος επηρεάζει την ορθότητα και την ταχύτητα της εκπαίδευσης. Στη παρούσα εργασία επιλέχθηκε ως βελτιστοποιητής, ο προσαρμοστικός αλγόριθμος παραγώγου (adaptive gradient algorithm - Adagard), ο οποίος χρησιμοποιείται στη πλειοψηφία των υλοποιημένων μοντέλων αυτόματης παραγωγής περίληψης και μετάφρασης και αποτελεί έναν εύρωστο μηχανισμό με γρήγορους χρόνους σύγκλισης. Σημαντική παράμετρος του βελτιστοποιητή αποτελεί ο ρυθμός μάθησης, ο οποίος είναι ένας πολλαπλασιαστικός παράγοντας που επηρεάζει το πόσο γρήγορα συγκλίνει το μοντέλο. Στη παρούσα εργασία ο ρυθμός μάθησης ορίστηκε ίσως με 0.15, καθώς με αυτόν τον ρυθμό μάθησης το δίκτυο κατάφερε να συγκλίνει στις επιθυμητές εξόδους σχετικά γρήγορα.

Στην εκπαίδευση μοντέλων βαθιάς μάθησης εφαρμόζεται ακόμη μια μέθοδος με την οποία χωρίζονται τα δεδομένα εκπαίδευσης σε μικρότερα υποσύνολα-δέσμες, που ονομάζονται

batches. Αφού ολοκληρωθεί η δημιουργία των batches, πραγματοποιείται η εκπαίδευση του μοντέλου σε κάθε ένα από αυτά ξεχωριστά, και όχι σε ολόκληρο το σύνολο των δεδομένων μονομιάς. Η χρήση αυτής της μεθόδου γίνεται για δυο λόγους. Ο πρώτος είναι η ελαχιστοποίηση της μνήμης του επεξεργαστή (CPU) που απαιτείται κατά τη διάρκεια της εκπαίδευσης. Αποθηκεύοντας και επεξεργάζοντας ένα μεγάλο όγκο δεδομένων τμηματικά και όχι εξολοκλήρου μονομιάς, μειώνει σημαντικά τους πόρους και τη μνήμη που δεσμεύονται κατά την εκπαίδευση. Ο δεύτερος λόγος είναι ότι χρησιμοποιώντας ολόκληρο το σύνολο δεδομένων κατά την εκπαίδευση, θα πρέπει σε κάθε κύκλο εκπαίδευσης να αποθηκεύονται τα σφάλματα-αποκλίσεις για όλα τα δεδομένα, Το γεγονός αυτό επιβραδύνει σημαντικά τη διαδικασία. Από την άλλη, εκπαιδεύοντας το μοντέλο με δέσμες, θα αποθηκεύονται τα σφάλματα μόνο των δεδομένων που ανήκουν στο δέσμη κάθε φορά, μειώνοντας τον απαιτούμενο χρόνο εκπαίδευσης. Η μέθοδος διαχωρισμού των δεδομένων εκπαίδευσης χρησιμοποιήθηκε και στη παρούσα εργασία, όπου δημιουργήθηκαν **4717** batches με **16** δεδομένα εκπαίδευσης το καθένα.

4.3.3 Διαδικασία Αξιολόγησης και Δοκιμής (Evaluation and Inference)

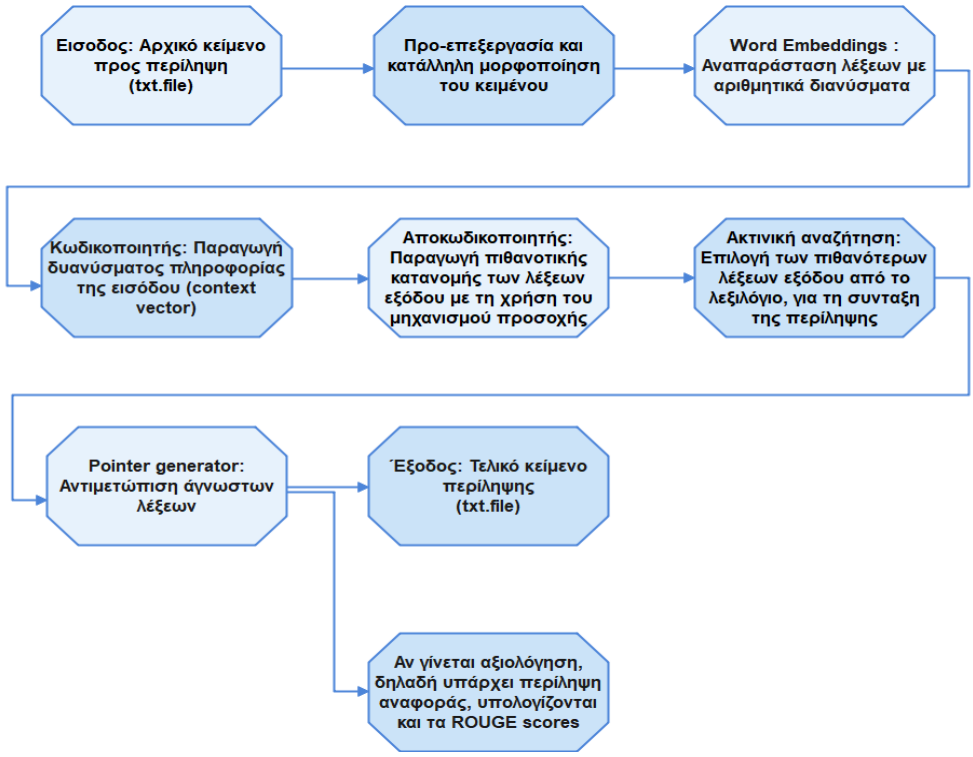
Οι διαδικασίες που ακολουθούνται κατά την αξιολόγηση (evaluation/testing) και κατά τη δοκιμή του σε άγνωστα κείμενα (inference) του συστήματος είναι παρόμοιες, για αυτό περιγράφονται ως μια. Η κύρια διαφορά είναι ότι στη περίπτωση της αξιολόγησης το μοντέλο δοκιμάζεται σε κείμενα για τα οποία υπάρχει περίληψη αναφοράς, ενώ στη δεύτερη περίπτωση το μοντέλο δοκιμάζεται σε κείμενα για τα οποία δεν υπάρχει κάποια έτοιμη περίληψη.

Τα κείμενα στη περίπτωση του evaluation προκύπτουν από το σύνολο των διαθέσιμων δεδομένων (ζευγών άρθρων-περιλήψεων), μετά το χωρισμό αυτού του συνόλου σε δεδομένα εκπαίδευσης και αξιολόγησης, όπως αναφέρθηκε παραπάνω. Συγκρίνοντας τις ήδη υπάρχουσες περιλήψεις αναφοράς αυτών των άρθρων, με αυτές που παράγονται από το μοντέλο, μπορεί να πραγματοποιηθεί η αξιολόγηση του με βάση των μετρικών Rouge.

Στη περίπτωση του inference , εισάγονται στο σύστημα άγνωστα κείμενα/άρθρα, τα οποία δεν εμπεριέχονται στο διαθέσιμο σύνολο κειμένων και συνεπώς δεν υπάρχει κάποια περίληψη αναφοράς. Αυτή η χρήση περιγράφει και το γενικότερο σκοπό του συστήματος, που είναι παραγωγή περίληψης ενός τυχαίου άγνωστου κειμένου του εκάστοτε χρήστη. Σε αυτή τη περίπτωση η αξιολόγηση της παραγόμενης περίληψης, βρίσκεται στη κρίση του εκάστοτε χρήστη.

Στη συνέχεια, περιγράφεται η ροή της πληροφορίας από το κείμενο εισόδου εως το κείμενο εξόδου, το οποίο αποτελεί τη περίληψη του αρχικού. Η περιγραφή γίνεται σειριακά, ακολουθώντας τη ροή της πληροφορίας για την καλύτερη κατανόηση της λειτουργίας του συστήματος από τον αναγνώστη. Στο παρακάτω διάγραμμα απεικονίζεται αυτή η ροή της

πληροφορίας του συστήματος αυτόματης παραγωγής περίληψης που υλοποιήθηκε στα πλαίσια αυτής της εργασίας.



Εικόνα 4.3: *Απεικόνιση της ροής της λειτουργίας του συστήματος αυτόματης παραγωγής περίληψης.*

Ως είσοδος στο σύστημα εισάγεται το κείμενο του οποίου την περίληψη καλείται να συντάξει το μοντέλο αυτόματης παραγωγής περίληψης που υλοποιήθηκε. Το κείμενο αυτό μπορεί να έχει οποιαδήποτε μορφή (π.χ. άρθρο, ανακοίνωση, διαφήμιση), με τη βασική προυπόθεση να είναι γραμμένο στην ελληνική γλώσσα. Στη περίπτωση εισαγωγής ενός νέου άγνωστου κειμένου, το κείμενο αυτό εισάγεται στο μοντέλο ως ένα txt αρχείο. Έπειτα το κείμενο “διαβάζεται” από το μοντέλο για την περαιτέρω επεξεργασία του, που προκύπτει από τα παρακάτω στάδια. Αντίστοιχη διαδικασία ακολουθείται και κατά την αξιολόγηση του μοντέλου, με τη κύρια διαφορά να είναι ότι αντί για ένα κείμενο, έχουμε πολλαπλά ζεύγη κειμένων, τα οποία αποτελούνται από άρθρα και την περίληψη αναφοράς τους. Επειδή τα κείμενα είναι πολλαπλά και σε μορφή ζευγών, η εισαγωγή τους γίνεται τμηματικά με .bin αρχεία.

Μετά την είσοδο του εκάστοτε κειμένου στο σύστημα ακολουθεί η κατάλληλη προ-επεξεργασία και μορφοποίηση για την αφαίρεση του λεγόμενου “θορύβου”, που στη περίπτωσή των κειμένων μπορεί να είναι κάποιος ξενόγλωσσος ή δυσνόητος χαρακτήρας ή δυσνόητο μήνυμα, και την κανονικοποίηση του κειμένου (μετατροπή όλων των χαρακτήρων σε πεζά γράμματα). Τα βήματα προ-επεξεργασίας – μορφοποίησης έχουν ήδη περιγράφει αναλυτικότερα στο

κεφάλαιο 3.3. Στη συνέχεια, ακολουθεί η αριθμητική αναπαράσταση της κάθε λέξης του κειμένου, τα λεγόμενα word embeddings. Αυτό είναι αναγκαίο γιατί το μοντέλο, όπως κάθε αλγόριθμος ή μηχανισμός μηχανικής μάθησης, αναγνωρίζει και επεξεργάζεται αριθμούς και όχι γράμματα.. Έτσι γίνεται δυνατή η "ανάγνωση" του κειμένου από το μοντέλο.

Στη συνέχεια, ακολουθεί ο αμφίδρομος LSTM κωδικοποιητής, ο οποίος τροφοδοτείται σε κάθε χρονικό βήμα με μια λέξη του κειμένου εισόδου και η έξοδος που παράγεται κάθε φορά ανατροφοδοτείται διατηρώντας τη χρήσιμη πληροφορία. Όταν ολόκληρο το κείμενο εισόδου περάσει από τον κωδικοποιητή, προκύπτει στην έξοδό του μια διανυσματική αναπαράσταση της συμπυκνωμένης πληροφορίας του κειμένου, που ονομάζεται context vector. Το διάνυσμα αυτό τροφοδοτείται στον LSTM αποκωδικοποιητή, οποίος με τη σειρά του σε κάθε χρονικό βήμα, παίρνει σαν είσοδο τη προηγούμενη, έξοδό του και συμβουλεύεται μέσω του μηχανισμού προσοχής το διάνυσμα context vector που παρήγαγε ο κωδικοποιητής. Έτσι, αποκωδικοποιητής γνωρίζει σε ποια τμήματα του αρχικού κειμένου να εστιάσει και αναλόγως παράγει σε κάθε εξοδό μια πιθανοτική κατανομή λέξεων. Συγκεκριμένα, η κατανομή αυτή είναι ένα διάνυσμα μεγέθους όσο και το μέγεθος του συνολικού λεξιλογίου του μοντέλου, στο οποίο κάθε αριθμητική τιμή αντιστοιχεί στη πιθανότητα μιας λέξης του λεξιλογίου να είναι η ορθότερη για να συμπεριληφθεί στη περίληψη.

Η τελική επιλογή των λέξεων της περίληψης πραγματοποιείται μέσω του μηχανισμού της ακτινικής αναζήτησης. Έστω ότι τίθεται ακτινική αναζήτηση μεγέθους Ν, σε αυτή τη περίπτωση αντί σε κάθε βήμα να επιλέγεται ως έξοδος η λέξη από τη πιθανοτική κατανομή με τη μεγαλύτερη τιμή-πιθανότητα, επιλέγονται οι Ν πιθανότερες. Για κάθε μια από αυτές τις λέξεις συνεχίζεται να παράγεται η ακολουθία εξόδου παίρνοντας στο επόμενο βήμα πάλι τις 5 πιθανότερες. Η διαδικασία αυτή συνεχίζεται μέχρι το τέλος, όπου τελικά επιλέγεται η ακολουθία με το συνολικό μεγαλύτερο σκορ. Στο τελικό στάδιο πραγματοποιείται η αντιμετώπιση των άγνωστων λέξεων μέσω του pointer generator. Συγκεκριμένα, όλες οι άγνωστες λέξεις που υπάρχουν στην εξαγόμενη ακολουθία - περίληψη, δηλαδή αυτές που συμβολίζονται με [unk], αντιγράφονται από το αρχικό κείμενο. Ένα χαρακτηριστικό παράδειγμα είναι, ότι αν σε κάποιο άρθρο αναφέρεται συχνά το όνομα ενός διάσημου προσώπου, το μοντέλο θα αναγνωρίσει ότι το όνομα αυτό είναι σημαντικό και θα το συμπεριλάβει στην εξαγόμενη περίληψη. Όμως το πιθανότερο είναι αυτό το όνομα να μην υπάρχει στο λεξιλόγιο του μοντέλου, οπότε ο μηχανισμός αντιγραφής έρχεται να αντιγράψει το όνομα αυτό για να είναι ορθή και κατανοητή η τελική περίληψη.

Η έξοδος του συστήματος είναι ένα .txt αρχείο. Στη περίπτωση της παραγωγής περίληψης ενός νέου άγνωστου κειμένου το txt αρχείο περιέχει τη περίληψη αυτού του κειμένου, ενώ σε περίπτωση αξιολόγησης το txt αρχείο περιέχει όλα τα κείμενα που δοκιμάστηκαν μαζί με την περίληψη αναφορά τους, τη παραγόμενη περίληψη τους από το σύστημα και τα αντίστοιχα ROUGE scores. Επίσης, στο πάνω μέρος του αρχείου καταγράφονται οι μέσοι όροι των μετρικών αξιολόγησης ROUGE από όλες τις παραγόμενες περιλήψεις. Στο **Παράρτημα Α** παρουσιάζονται παραδείγματα αξιολόγησης/δοκιμής του συστήματος

4.3.4 Υλοποίηση στο υπολογιστικό σύστημα ARIS

Σε αυτό το σημείο πρέπει να σημειωθεί πως η εκπαίδευση ενός μοντέλου βαθιάς μηχανικής μάθησης, όπως είναι το μοντέλο αυτόματης παραγωγής περίληψης που υλοποιήθηκε, απαιτεί αρκετούς υπολογιστικούς πόρους και μνήμη για να πραγματοποιηθεί. Συνεπώς, η εκπαίδευση ενός τέτοιου συστήματος σε ένα μεγάλο όγκο δεδομένων είναι τρομερά δύσκολο να πραγματοποιηθεί σε έναν τοπικό υπολογιστή. Το πρόβλημα αυτό επιλύθηκε μέσω του Εθνικού Δικτύου Υποδομών Τεχνολογίας και Έρευνας (ΕΔΥΤΕ) και του υπολογιστικού του συστήματος υψηλών επιδόσεων ARIS (Advanced Research Information System) [35]. Το ARIS αποτελεί μια υποδομή υπολογιστικών πόρων, την οποία παρέχει το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας στην επιστημονική και τεχνολογική κοινότητα για την διεργασία ερευνών και έργων.

Η υποδομή ARIS αποτελείται από τέσσερις κόμβους υπολογιστικών συστημάτων αρχιτεκτονικής Intel x86 οι οποίοι συνδέονται μεταξύ τους, δημιουργώντας ένα ενιαίο δίκτυο υπολογιστικών πόρων.

GPU node: 44 εξυπηρετητές Dell PowerEddge R730 με το κάθε εξυπηρετητή να περιέχει 2 επεξεργαστές Intel Xeon E5-2660v3 64 GB μνήμης ο καθένας και 2 κάρτες GPU NVidia K40.

Όλες οι διαδικασίες εκπαίδευσης, αξιολόγησης και δοκιμής των μοντέλων που υλοποιήθηκαν στα πλαίσια αυτής της εργασίας, πραγματοποιήθηκαν στο συγκεκριμένο υπολογιστικό κόμβο. Η εκτέλεση όλων των πειραμάτων έγινε σε περιβάλλον Command Line Interface (CLI), απομακρυσμένα μέσω σύνδεσης Secure Shell (SSH), και χρησιμοποιώντας shell scripts. Μέσω των shell scripts υπήρχε η δυνατότητα καθορισμού των επιθυμητών υπολογιστικών πόρων (πυρήνων επεξεργασίας, μνήμη) και του επιθυμητού χρόνου εκτέλεσης της εργασίας που εκτελούσε το κάθε shell script. Με αυτό το τρόπο υπήρχε έλεγχος όχι μόνο της εκάστοτε διαδικασίας υλοποίησης, αλλά και των διαθέσιμων υπολογιστικών πόρων. Η συμβολή του ΕΔΥΤΕ αποδείχθηκε πολύτιμη στην υλοποίηση της εργασίας και για αυτό το ευχαριστούμε θερμά.

4.4 Διερεύνηση παραμέτρων

Σε αυτό το κεφάλαιο παρουσιάζονται οι δοκιμές που πραγματοποιήθηκαν για την αξιολόγηση του υλοποιημένου συστήματος αυτόματης παραγωγής περίληψης και τη διερεύνηση των παραμέτρων του. Συγκεκριμένα, παρουσιάζονται τα αποτελέσματα των διάφορων μοντέλων που σχεδιάστηκαν με την επιλογή διαφορετικών τιμών των προς διερεύνηση παραμέτρων. Μέσω των αποτελεσμάτων αυτών διερευνάται ο τρόπος με τον οποίο επηρεάζει η κάθε παράμετρος την τελική επίδοση του μοντέλου, λαμβάνοντας υπόψιν και τον απαιτούμενο χρόνο εκπαίδευσης του σε κάθε περίπτωση. Χρησιμοποιώντας ως μετρικές αξιολόγησης τις μετρικές Rouge, συγκρίνονται τα επιμέρους μοντέλα που σχεδιάστηκαν για να καταλήξουμε στο βέλτιστο μοντέλο. Το βέλτιστο μοντέλο θα συγκριθεί στο επόμενο κεφάλαιο με παλαιότερες υλοποιήσεις μοντέλων αυτόματης παραγωγής περίληψης.

Στο παρακάτω πίνακα παρουσιάζονται όλοι οι παράμετροι τους οποίους ο χρήστης έχει τη δυνατότητα να μεταβάλλει μέσω του λογισμικού, επηρεάζοντας σημαντικά την επίδοση του μοντέλου. Καθώς επίσης και οι αριθμητικές τιμές για τις οποίες οι βασικότεροι από αυτούς διερευνήθηκαν.

Παράμετρος	**Τιμές Παραμέτρου**
Μέγεθος κρυφών καταστάσεων κωδικοποιητή και αποκωδικοποιητή -> **hidden_dim**	128, 256, 320
Μέγεθος word embeddings -> **emb_dim**	150
Μέγεθος δέσμης εκπαίδευσης -> **batch_size**	16
Μέγιστο μέγεθος κειμένου εισόδου -> **max_enc_steps**	200, 400, 600 (λέξεις)
Μέγιστο μέγεθος περίληψης -> **max_dec_steps**	50, 100, 200 (λέξεις)
Μέγεθος λεξιλογίου -> **vocab_size**	50000, 100000
Μέγεθος ακτινικής αναζήτησης -> **beam_size**	8, 12
Ρυθμός εκπαίδευσης -> **learning_rate**	0.15

Πίνακας 4.1: *Παρουσίαση μεταβαλλόμενων παραμέτρων μοντέλου.*

Όπως προκύπτει από τον παραπάνω πίνακα οι παράμετροι που διερευνήθηκαν είναι οι : hidden_dim, mac_enc_steps, max_dec_steps, vocab_size και beam_size. Επίσης συγκρίθηκε η επίδοση του μοντέλου ως προς τη χρήση προ-επεξεργασμένων ή μη δεδομένων.

Για την οπτικοποίηση και παρακολούθηση της διαδικασίας της εκπαίδευσης χρησιμοποιήθηκε το εργαλείο TensorBoard της TensorFlow [36]. Σε κάθε μοντέλο που δοκιμάστηκε παρουσιάζεται η γραφική παράσταση απόκλισης ανά βήμα εκπαίδευσης (loss per training steps) που παρέχεται για κάθε μοντέλο από το TesnorBoard.

Χωρίζοντας τα δεδομένα εκπαίδευσης σε δέσμες εκπαίδευσης (batches) μεγέθους 16 προκύπτει ότι για κάθε εποχή εκπαίδευσης, δηλαδή για κάθε ολοκλήρωση της εκπαίδευσης σε όλα τα

δεδομένα, απαιτούνται 73468 / 16 = 4592 βήματα εκπαίδευσης (training steps) για την ολοκλήρωση μιας εποχής (epoch).

Στη συνέχεια παρουσιάζονται τρία μοντέλα που σχεδιάστηκαν προς μελέτη της επίδρασης του μεγέθους των κρυφών καταστάσεων στην επίδοση του συστήματος αυτόματης παραγωγής περίληψης. Η παράμετρος αυτή αποτελεί μια από τις βασικότερες παραμέτρους του συστήματος και για αυτό η επιλογή της πρέπει να γίνει με προσοχή. Πολύ μικρές τιμές του μεγέθους κρυφών καταστάσεων μπορεί να οδηγήσουν σε ανικανότητα του συστήματος να συγκλίνει σε ορθά και γενικευμένα αποτελέσματα. Ενώ πολύ μεγάλες τιμές τείνουν να αυξάνουν δραματικά τον απαιτούμενο χρόνο και την απαιτούμενη μνήμη κατά την εκπαίδευση.

4.4.1 Διερεύνηση μεγέθους κρυφών καταστάσεων

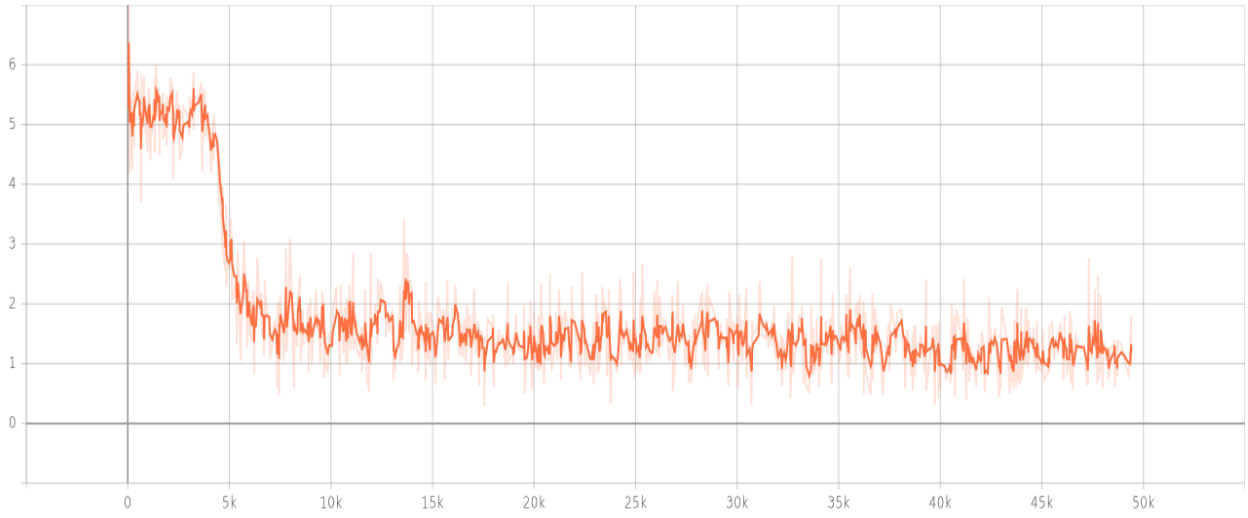
Μοντέλο με μέγεθος κρυφών καταστάσεων 128

Το συγκεκριμένο μοντέλο σχεδιάστηκε και δοκιμάστηκε για την διερεύνηση της επίδρασης των κρυφών καταστάσεων του κωδικοποιητή και αποκωδικοποιητή στην επίδοση του μοντέλου. Η επιλογή της παραμέτρου ορίστηκε σε **128 κρυφές καταστάσεις (hidden layers).**

Οι υπόλοιποι παράμετροι ορίστηκαν στα πλαίσια του συγκεκριμένου μοντέλου ως εξής : emb_dim = 150, batch_size = 16, max_enc_steps = 400, max_dec_steps = 100, vocab_size = 50000, beam_size = 8 και learning_rate = 0.15.

Υπολογίστηκε ότι ο χρόνος που απαιτείται για την εκπαίδευση του μοντέλου για μία εποχή είναι **1 ώρα και 14 λεπτά.** Για **10 εποχές(=45920 training steps),** ο συνολικός χρόνος εκπαίδευσης είναι **12 ώρες και 50 λεπτά.**

Στο παρακάτω σχήμα παρουσιάζεται το γράφημα loss per training step.



Εικόνα 4.4: *Γραφική παράσταση συνάρτησης κόστους (loss per training steps) για 50000 training steps του μοντέλου με 128 hidden layers.*

Το σημείο στο οποίο η συνάρτηση κόστους ελαχιστοποιείται και αρχίζει να ομαλοποιείται γύρω από αυτές τις τιμές είναι κοντά στα 10000 training steps δηλαδή στα 2 epochs. Αυτό είναι και το σημείο στο οποίο αρχίζει να συγκλίνει το μοντέλο. Η πολύ παραπάνω εκπαίδευση του μοντέλου πέρα από αυτό το σημείο μπορεί να προκαλέσει overfitting, το οποίο επηρεάζει την επίδοσή του σε άγνωστα δεδομένα. Ο χρόνος λοιπόν που απαιτείται για την σύγκλιση του μοντέλου είναι περίπου **2 ώρες και 30 λεπτά.**

Στο παρακάτω πίνακα παρουσιάζονται τα Rouge scores που προκύπτουν σε κάθε εποχή εκπαίδευσης του προς μελέτη μοντέλου:

	Rouge-1 F1 score	**Rouge-2 F1 score**	**Rouge-L F1 score**
1 Epoch (4592 steps)	0.1008	0.0015	0.0490
2 Epochs (9184 steps)	0.3735	0.2411	0.3112
3 Epochs (13776 steps)	**0.3791**	**0.2507**	**0.3161**
4 Epochs (18368 steps)	0.3704	0.2391	0.3072
5 Epochs (22960 steps)	0.3638	0.2299	0.3017
6 Epochs (27552 steps)	0.3444	0.2107	0.2809
7 Epochs (32144 steps)	0.3565	0.2250	0.2961
8 Epochs (36736 steps)	0.3766	0.2449	0.3123
9 Epochs (41328 steps)	0.3721	0.2406	0.3094
10 Epochs (45920 steps)	0.3420	0.2088	0.2821

Πίνακας 4.2: *Πίνακας αποτελεσμάτων (Rouge scores) για το μοντέλο με 128 hidden layers για 10 epochs.*

Παρατηρείται ότι το βέλτιστο μοντέλο με 128 hidden layers προκύπτει από **3 epochs -> 13776 training steps** και επιτυγχάνεται μετά από **3 ώρες και 42 λεπτά.**

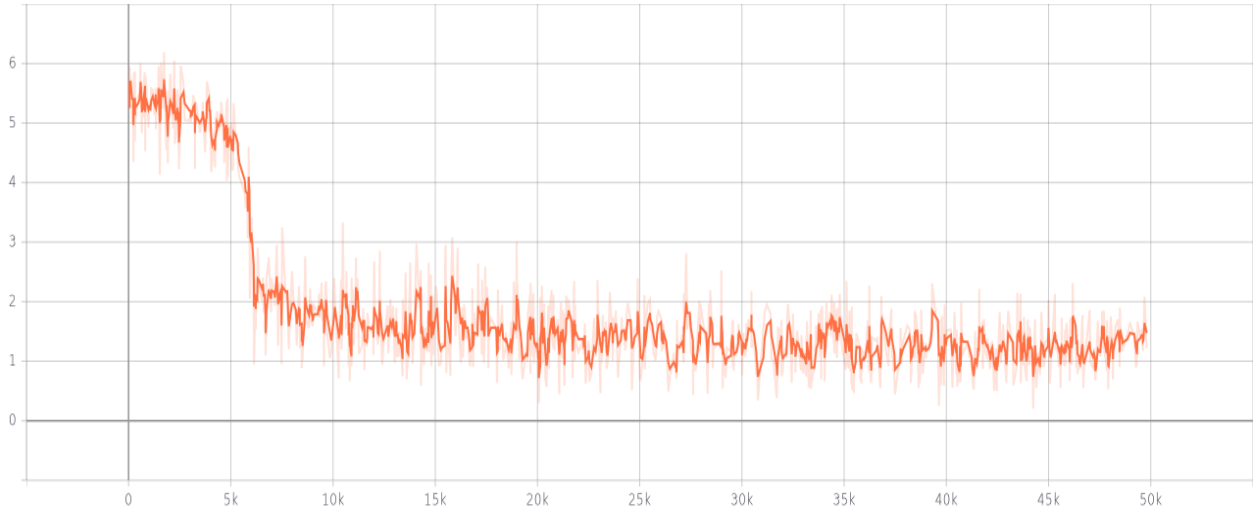
Μοντέλο με μέγεθος κρυφών καταστάσεων 256

Όπως και το προηγούμενο, το συγκεκριμένο μοντέλο σχεδιάστηκε και δοκιμάστηκε για την διερεύνηση της επίδρασης του μεγέθους των κρυφών καταστάσεων του κωδικοποιητή και αποκωδικοποιητή στην επίδοση του μοντέλου. Η επιλογή της παραμέτρου ορίστηκε τώρα σε **256 κρυφά στρώματα (hidden layers).**

Οι άλλες παράμετροι ορίστηκαν στα πλαίσια του συγκεκριμένου μοντέλου ως εξής : emb_dim = 150, batch_size = 16, max_enc_steps = 400, max_dec_steps = 100, vocab_size = 50000, beam_size = 8 και learning_rate = 0.15.

Υπολογίστηκε ότι ο χρόνος που απαιτείται για την εκπαίδευση του μοντέλου για μία εποχή είναι **1 ώρα και 40 λεπτά.** Παρατηρείται ότι απαιτείται περισσότερος χρόνος για την ολοκλήρωση μιας εποχής σε σχέση με το μοντέλο των 128 hidden layers. Αυτό συμβαίνει γιατί αυξάνοντας τον αριθμό των κρυφών στρωμάτων, αυξάνεται η πολυπλοκότητα του νευρωνικού δικτύου. Συνεπώς, αυξάνονται οι υπολογισμοί και το χρονικό κόστος.

Στο παρακάτω σχήμα παρουσιάζεται το γράφημα loss per training step.



Εικόνα 4.5: *Γραφική παράσταση συνάρτησης κόστους (loss per training steps) για 50000 training steps του μοντέλου με 256 hidden layers.*

Το σημείο στο οποίο η συνάρτηση κόστους ελαχιστοποιείται και αρχίζει να ομαλοποιείται γύρω από αυτές τις τιμές είναι κοντά στα 10000 training steps δηλαδή στα 2 epochs. Τώρα όμως το χρονικό κόστος αυξήθηκε στις **3 ώρες και 25 λεπτά,** λόγω των περισσότερων κρυφών στρωμάτων.

Στο παρακάτω πίνακα παρουσιάζονται τα Rouge scores που προκύπτουν σε κάθε εποχή εκπαίδευσης του προς μελέτη μοντέλου:

	Rouge-1 F1 score	**Rouge-2 F1 score**	**Rouge-L F1 score**
1 Epoch (4592 steps)	0.1106	0.0017	0.0533
2 Epochs (9184 steps)	0.3498	0.1960	0.2742
3 Epochs (13776 steps)	0.3626	0.2304	0.3006
4 Epochs (18368 steps)	0.3756	0.2439	0.3134
5 Epochs (22960 steps)	0.3523	0.2174	0.2901
6 Epochs (27552 steps)	**0.3832**	**0.2582**	**0.3177**
7 Epochs (32144 steps)	0.3656	0.2321	0.3033
8 Epochs (36736 steps)	0.3720	0.2406	0.3118
9 Epochs (41328 steps)	0.3586	0.2238	0.2958
10 Epochs (45920 steps)	0.3709	0.2386	0.3096

Πίνακας 4.3: *Πίνακας αποτελεσμάτων (Rouge scores) για το μοντέλο με 256 hidden layers για 10 epochs.*

Παρατηρείται ότι το βέλτιστο μοντέλο με 256 hidden layers προκύπτει από **6 epochs -> 27552 training steps** και επιτυγχάνεται μετά από **10 ώρες.** Η αύξηση των κρυφών καταστάσεων σε 256 από 128 οδήγησε στην ανάπτυξή ενός πιο αποδοτικού μοντέλου, όπως προκύπτει από τις μετρικές ROUGE. Αυτό

εξηγείται από το γεγονός ότι, ένα νευρωνικό δίκτυο με περισσότερα layers, επεξεργάζεται πιο διεξοδικά τη πληροφορία και αναγνωρίζει περισσότερες συσχετίσεις μεταξύ των λέξεων του αρχικού κειμένου, με αποτέλεσμα η παραγόμενη περίληψη να προσεγγίζει καλύτερα την περίληψη αναφοράς. Από την άλλη αυξάνεται το χρονικό κόστος, καθώς όπως προαναφέρθηκε περισσότερα κρυφά στρώματα απαιτούν περισσότερους υπολογισμούς.

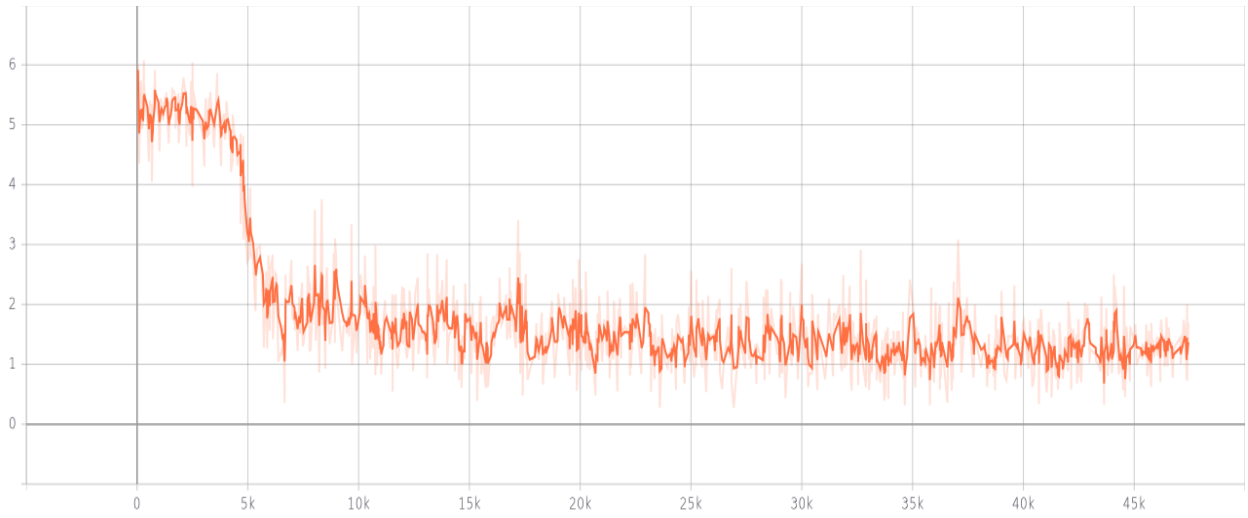
Μοντέλο με μέγεθος κρυφών καταστάσεων 320

Αντιστοίχως με τα προηγούμενα μοντέλα, το συγκεκριμένο μοντέλο σχεδιάστηκε και δοκιμάστηκε για την διερεύνηση της επίδρασης του μεγέθους των κρυφών καταστάσεων του κωδικοποιητή και αποκωδικοποιητή στην επίδοση του μοντέλου. Η επιλογή της παραμέτρου αυτή τη φορά ορίστηκε σε **320 κρυφές καταστάσεις (hidden layers).**

Οι άλλες παράμετροι του μοντέλου ορίστηκαν ως εξής : emb_dim = 150, batch_size = 16, max_enc_steps = 400, max_dec_steps = 100, vocab_size = 50000, beam_size = 8 και learning_rate = 0.15.

Υπολογίστηκε ότι ο χρόνος που απαιτείται για την εκπαίδευση του μοντέλου για μία εποχή είναι **2 ώρες και 6 λεπτά.** Ο απαιτούμενος χρόνος εκπαίδευσης αυξήθηκε και άλλο, λόγω της αύξησης του αριθμού των κρυφών καταστάσεων σε 320.

Στο παρακάτω σχήμα παρουσιάζεται το γράφημα loss per training step.



Εικόνα 4.6: *Γραφική παράσταση συνάρτησης κόστους (loss per training steps) για 50000 training steps του μοντέλου με 320 hidden layers.*

Το μοντέλο αυτή τη φορά συγκλίνει στα 10000 training steps, δηλαδή στα 2-3 epochs, μετά από **5 ώρες και 15 λεπτά**. Παρατηρείται αρκετά μεγάλη διαφορά από το αντίστοιχο χρονικό κόστος του μοντέλου με τα 128 hidden layers (3 ώρες και 20 λεπτά), γεγονός που δικαιολογείται πλήρως, καθώς υπάρχει σχεδόν τριπλασιασμός των κρυφών καταστάσεων.

Στο παρακάτω πίνακα παρουσιάζονται τα Rouge scores που προκύπτουν σε κάθε εποχή εκπαίδευσης του προς μελέτη μοντέλου:

	Rouge-1 F1 score	**Rouge-2 F1 score**	**Rouge-L F1 score**
1 Epoch (4592 steps)	0.1758	0.0191	0.0984
2 Epochs (9184 steps)	0.3773	0.2443	0.3119
3 Epochs (13776 steps)	0.3815	0.2512	0.3185
4 Epochs (18368 steps)	0.3775	0.2469	0.3155
5 Epochs (22960 steps)	0.3720	0.2418	0.3096
6 Epochs (27552 steps)	0.3837	0.2499	0.3201
7 Epochs (32144 steps)	**0.4008**	**0.2706**	**0.3407**
8 Epochs (36736 steps)	0.3711	0.2386	0.3081
9 Epochs (41328 steps)	0.3611	0.2271	0.2983
10 Epochs (45920 steps)	0.3727	0.2391	0.3086

Πίνακας 4.4*: Πίνακας αποτελεσμάτων (Rouge scores) για το μοντέλο με 320 hidden layers για 10 epochs.*

Παρατηρείται ότι το βέλτιστο μοντέλο με 320 hidden layers προκύπτει από **7 epochs -> 32144 training steps**. Παρατηρείται επιπλέον αύξηση της αποδοτικότητας του μοντέλου σύμφωνα με τις μετρικές ROUGE, φτάνοντας την καλύτερη επίδοση των δοκιμασμένων μοντέλων μέχρι τώρα με αποτελέσματα:

[Rouge-1 F1 : 0.4008, Rouge-2 F1 : 0.2706, Rouge-L F1 : 0.3407]

Ως φυσικό επακόλουθο της αύξησης των κρυφών καταστάσεων, παρατηρείται αύξηση του απαιτούμενου χρόνου εκπαίδευσης για την απόκτηση του βέλτιστου μοντέλου, συγκεκριμένα ο απαιτούμενος χρόνος είναι **14 ώρες και 32 λεπτά.**

Το μοντέλο αυτό χρησιμοποιήθηκε ως μοντέλο αναφοράς και μέτρο σύγκρισης για την διερεύνηση του τρόπου με τον οποίο διάφοροι παράμετροι επηρεάζουν την επίδοση του μοντέλου. Τα αντίστοιχα πειράματα παρουσιάζονται στη συνέχεια.

4.4.2 Διερεύνηση προ-επεξεργασίας δεδομένων

Παίρνοντας ως μοντέλο αναφοράς αυτό που προέκυψε από τη προηγούμενη υποενότητα με παραμέτρους: hidden_layers: 320, emb_dim = 150, batch_size = 16, max_enc_steps = 400, max_dec_steps = 100, beam_size = 8, learning_rate = 0.15, διερευνάτε η επίδραση που έχει η προ-επεξεργασία των δεδομένων στην επίδοση και αποτελεσματικότητα του συστήματος. Οι περιπτώσεις που εξετάζονται είναι δυο:

- μοντέλο εκπαιδευμένο σε **προ-επεξεργασμένα δεδομένα**

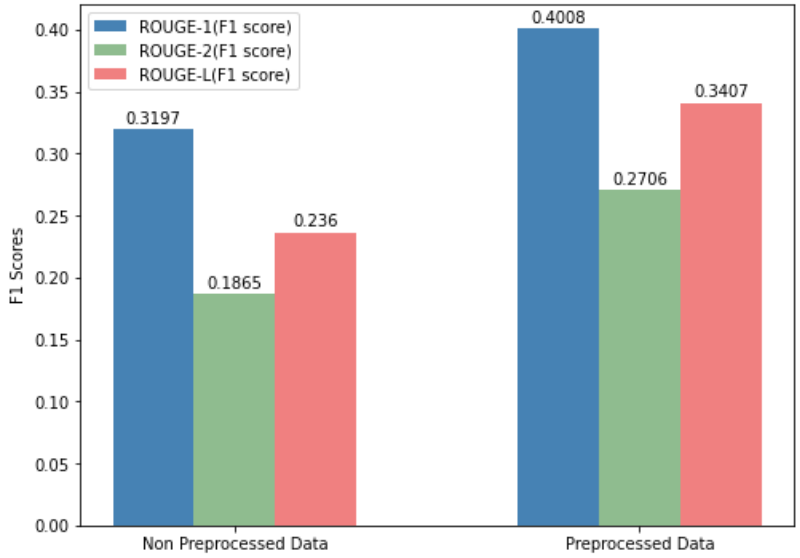
- μοντέλο εκπαιδευμένο στα πρωτότυπα, **μη προ-επεξεργασμένα δεδομένα**

Εφόσον χρησιμοποιείται το βέλτιστο μοντέλο που προέκυψε από τις προηγούμενες δοκιμές, ως μοντέλο αναφοράς και μέτρο σύγκρισης, το μοντέλο χωρίς τα προ-επεξεργασμένα δεδομένα εκπαιδεύτηκε για το ίδιο χρονικό διάστημα, δηλαδή **14 ώρες και 32 λεπτά.**

Τα αποτελέσματα παρουσιάζονται παρακάτω.

	Rouge-1 F1 score	**Rouge-2 F1 score**	**Rouge-L F1 score**
With Non preprocessed Data	0.3197	0.1865	0.2360
With Preprocessed Data	0.4008	0.2706	0.3407

Πίνακας 4.5*: Σύγκριση αποτελεσμάτων χρήσης και μη χρήσης προ-επεξεργασμένων δεδομένων.*



Εικόνα 4.7: *Αναπαράσταση σύγκρισης αποτελεσμάτων χρήσης και μη χρήσης προ-επεξεργασμένων δεδομένων.*

Όπως έχει τονιστεί αρκετές φορές στη παρούσα εργασία, η προ-επεξεργασία των δεδομένων είναι ένα πολύ σημαντικό στάδιο και είναι απαραίτητο για την υλοποίηση ενός αποδοτικού και εύρωστου μοντέλου. Αυτό είναι ιδιαίτερα εμφανές στο πρόβλημα της αυτόματης παραγωγής περίληψης, όπου ο “θόρυβος” στα κείμενα μπορεί να έχει πάρα πολλές μορφές. Η μεγάλη σημαντικότητα της προ-επεξεργασίας επαληθεύεται από τα παραπάνω αποτελέσματα και τις αξιοσημείωτες διαφορές, που δηλώνουν ξεκάθαρα ότι η μη προ-επεξεργασία των δεδομένων μειώνει σημαντικά την επίδοση του μοντέλου.

4.4.3 Διερεύνηση μεγέθους λεξιλογίου

Μία άλλη σημαντική παράμετρος είναι το μέγεθος του λεξιλογίου του μοντέλου. Το λεξιλόγιο, περιέχοντας όλες τις γνωστές λέξεις του μοντέλου, καθορίζει σε μεγάλο βαθμό τη γενικότητα του και το θέμα-ύφος των κειμένων στα οποία αποδίδει καλύτερα. Είναι προφανές πως όσο μεγαλύτερο είναι το λεξιλόγιο, τόσο πιο γενικευμένο είναι το μοντέλο. Από την άλλη όμως ένα μεγαλύτερο λεξιλόγιο αυξάνει τις απαιτήσεις σε χρόνο και μνήμη, τόσο κατά τη διάρκεια της εκπαίδευσης, όσο και κατά τη διάρκεια της δοκιμής του μοντέλου, καθώς αυξάνονται οι επιλογές λέξεων σε κάθε έξοδο. Με στόχο τη διερεύνηση της επίδοσης του μοντέλου σε σχέση με το μέγεθος του λεξιλογίου, πραγματοποιήθηκε σύγκριση δυο μοντέλων:

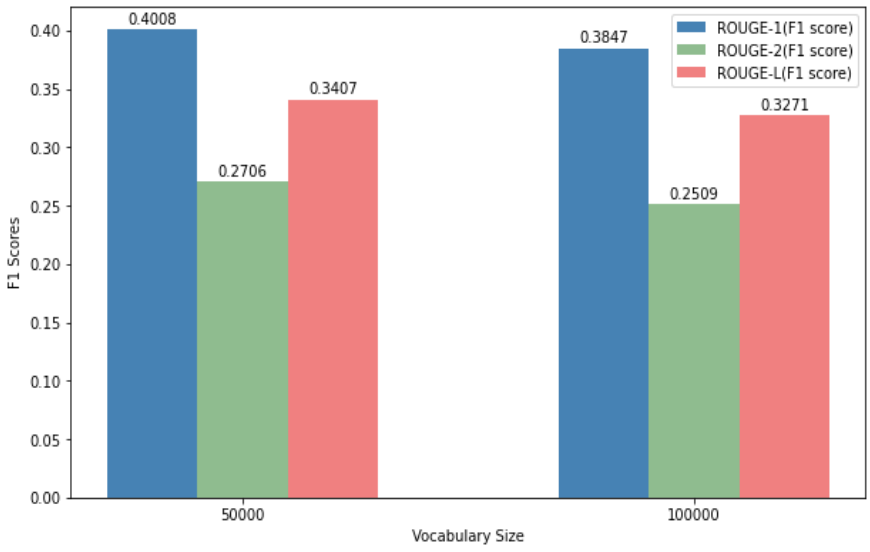
- μοντέλο με μέγεθος λεξιλογίου **50000 λέξεις**
- μοντέλο με ολόκληρο το λεξιλόγιο - **100000 λέξεις**

Ως μοντέλο αναφοράς τέθηκε το βέλτιστο μοντέλο που προέκυψε από το προηγούμενο κεφάλαιο, οπότε οι υπόλοιποι παράμετροι είναι οι εξής: hidden_layers: 320, emb_dim = 150, batch_size = 16, max_enc_steps = 400, max_dec_steps = 100, vocab_size = 50000, beam_size = 8, learning_rate = 0.15 και ο χρόνος εκπαίδευσης είναι **14 ώρες και 32 λεπτά.**

Τα αποτελέσματα παρουσιάζονται παρακάτω.

vocab_size	Rouge-1 F1 score	Rouge-2 F1 score	Rouge-L F1 score
50000	0.4008	0.2706	0.3407
100000	0.3847	0.2509	0.3271

Πίνακας 4.6: *Σύγκριση αποτελεσμάτων διερεύνησης μεγέθους λεξιλογίου.*



Εικόνα 4.8: *Αναπαράσταση σύγκρισης αποτελεσμάτων διερεύνησης λεξιλογίου.*

Παρατηρείτε ότι η αύξηση του λεξιλογίου δεν οδηγεί στη βελτίωση των αποτελεσμάτων του συστήματος, αντιθέτως στη συγκεκριμένη περίπτωση προκαλεί μείωση της επίδοσής του. Αυτό οφείλεται στο γεγονός ότι διπλασιάζοντας το μέγεθος του λεξιλογίου, το μοντέλο χρειάζεται πολύ περισσότερο χρόνο για να εκπαιδευτεί με βέλτιστο τρόπο, καθώς σε κάθε βήμα εξόδου υπάρχουν διπλάσιες πιθανές λέξεις από τις οποίες πρέπει να επιλεχθεί η κατάλληλη. Η παρατήρηση αυτή επαληθεύεται από τα παρακάτω δεδομένα, σύμφωνα με τα οποία η εκπαίδευση του μοντέλου με ολόκληρο το λεξιλόγιο των 100000 λέξεων είναι πολύ πιο αργή.

Συνολικά training steps σε 14 ώρες και 32 λεπτά:

- vocab_size = 50000 -> 32144 steps (7 epochs)
- vocab_size = 100000 -> 11594 steps (2.3 epochs)

4.4.4 Διερεύνηση μέγιστου μεγέθους κειμένου εισόδου και περίληψης

Οι παράμετροι max_enc_steps και max_dec_steps εκφράζουν το μέγιστο αριθμό των timesteps του κωδικοποιητή και του αποκωδικοποιητή και κατά επέκταση το μέγιστο μήκος κειμένου εισόδου και εξόδου αντίστοιχα σε λέξεις. Οι παράμετροι αυτοί παρουσιάζουν αρκετό ενδιαφέρον, καθώς το μέγεθος της παραγόμενης περίληψης είναι κάτι που απασχολεί αρκετά το χρήστη, που άλλες φορές αναζητά μεγαλύτερης και άλλες μικρότερης έκτασης περίληψη. Για αυτό αποφασίστηκε να διερευνηθεί η επίδραση αυτών των παραμέτρων στην αποτελεσματικότητα του συστήματος.

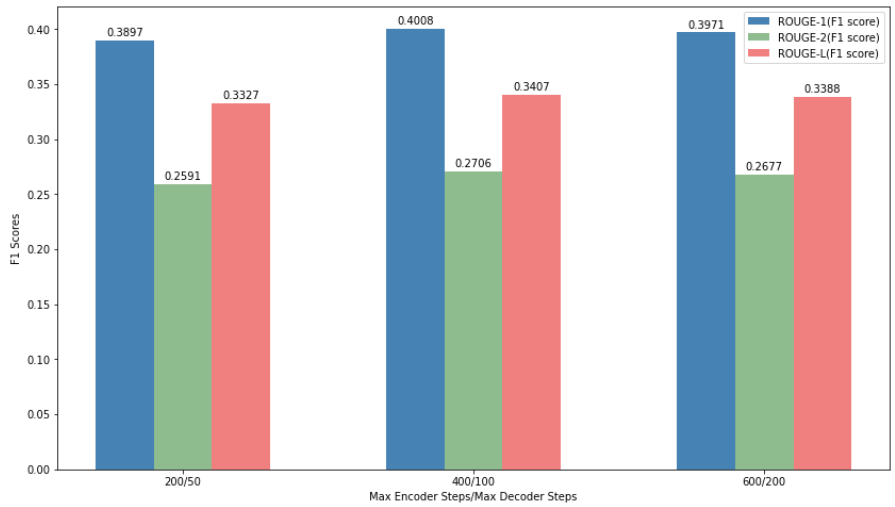
Η διερεύνηση αυτή πραγματοποιήθηκε συγκρίνοντας τα αποτελέσματα τριών μοντέλων, στα οποία όλοι οι παράμετροι τίθενται ίδιοι με αυτούς του βέλτιστου μοντέλου , εκτός από τους παραμέτρους max_enc_steps και max_dec_steps που μεταβάλλονται. Σε κάθε περίπτωση αποτελεί αναγκαία συνθήκη η σχέση max_enc_steps > max_dec_steps, αλλιώς διακινδυνεύεται η περίληψη να είναι μεγαλύτερη από το αρχικό κείμενο. Τα μοντέλα που συγκρίνονται είναι τα εξής:

- μοντέλο με max_enc_steps/max_dec_steps = **200/50**
- μοντέλο με max_enc_steps/max_dec_steps = **400/100**
- μοντέλο με max_enc_steps/max_dec_steps = **600/200**

Και τα 3 μοντέλα εκπαιδεύτηκαν για 14 ώρες και 32 λεπτά σύμφωνα με το βέλτιστο μοντέλο. Τα αποτελέσματα παρουσιάζονται παρακάτω.

max_enc_steps/max_dec_steps	Rouge-1 F1 score	Rouge-2 F1 score	Rouge-L F1 score
200/50 steps	0.3897	0.2591	0.3327
400/100 steps	0.4008	0.2706	0.3407
600/200 steps	0.3971	0.2677	0.3388

Πίνακας 4.7: *Σύγκριση αποτελεσμάτων διερεύνησης μέγιστου μεγέθους κειμένου εισόδου και παραγόμενης περίληψης.*



Εικόνα 4.9: *Αναπαράσταση σύγκρισης αποτελεσμάτων διερεύνησης μέγιστου μεγέθους κειμένου εισόδου και παραγόμενης περίληψης.*

Συνολικά training steps σε 14 ώρες και 32 λεπτά:

- max_enc_steps/max_dec_steps = **200/50** -> 62280 steps (13,4 epochs)
- max_enc_steps/max_dec_steps = **400/100** -> 32144 steps (7 epochs)
- max_enc_steps/max_dec_steps = **600/200** -> 21583 steps (4.7 epochs)

Παρατηρείται ότι τα αποτελέσματα είναι πολύ κοντά μεταξύ τους, με τη διαφορά ανάμεσα στο μοντέλο με 400/100 steps και αυτό με τα 600/200 steps να είναι μόνο 0.003 σε όλες τις μετρικές. Ενώ και η διαφορά με το απλούστερο μοντέλο των 200/50 steps είναι της τάξης του 0.01. Από την άλλη όμως παρατηρούνται μεγάλες διαφορές στους χρόνους εκπαίδευσης, καθώς για το ίδιο χρονικό διάστημα εμφανίζεται εως και τριπλάσια επιτάχυνση της διαδικασίας της εκπαίδευση του μοντέλου με 200/50 steps σε σχέση με αυτό των 600/200 steps. Το αποτέλεσμα αυτό είναι λογικό, καθώς για μεγαλύτερο κείμενο εισόδου απαιτείται περισσότερος χρόνος για τη σύνταξη της περίληψης του.

4.4.5 Διερεύνηση μεγέθους ακτινικής αναζήτησης

Η τελευταία παράμετρος που διερευνήθηκε είναι το μέγεθος της ακτινικής αναζήτησης((beam_size). Η παράμετρος αυτή καθορίζει το πλήθος των πιθανότερων λέξεων που επιλέγονται σε κάθε παραγωγή μιας εξόδου- λέξης από τον αποκωδικοποιητή. Αφού λοιπόν επηρεάζει ένα σημαντικό στάδιο της διαδικασίας που αποτελεί η τελική επιλογή των παραγόμενων λέξεων, αξίζει να διερευνηθεί η επίδρασή της στην επίδοση του μοντέλου.

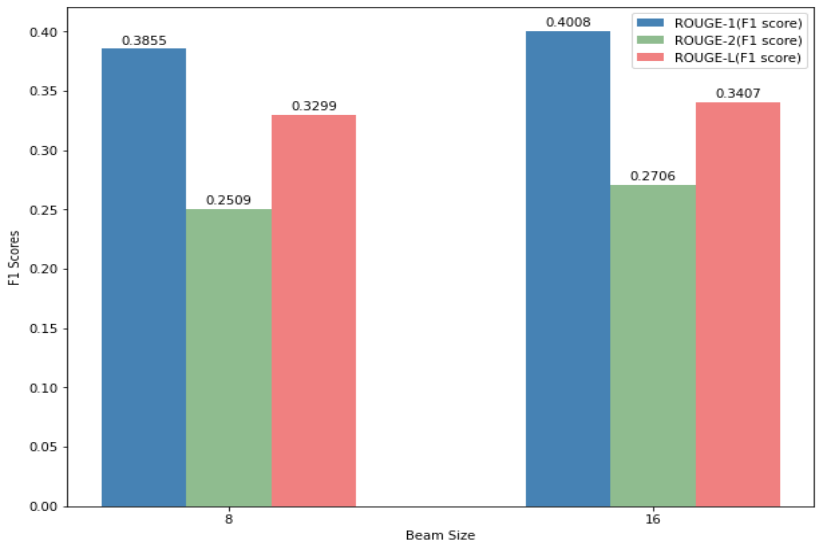
Για την μελέτη αυτή συγκρίθηκαν δυο μοντέλα με διαφορετικά μεγέθη ακτινικής αναζήτησης το κάθε ένα. Οι υπόλοιποι παράμετροι τίθενται σύμφωνα με το βέλτιστο μοντέλο που περιγράφηκε στο προηγούμενο κεφάλαιο. Τα δύο μοντέλα που προκύπτουν είναι:

- μοντέλο με beam_size = **8**
- μοντέλο με beam_size = **16**

Η εκπαίδευση και των δυο μοντέλων πραγματοποιήθηκε για 14 ώρες και 32 λεπτά σύμφωνα με το βέλτιστο μοντέλο. Τα αποτελέσματα παρουσιάζονται παρακάτω.

Beam size	Rouge-1 F1 score	Rouge-2 F1 score	Rouge-L F1 score
8	0.3855	0.2509	0.3299
16	0.4008	0.2706	0.3407

Πίνακας 4.8: *Σύγκριση αποτελεσμάτων διερεύνησης μεγέθους ακτινικής αναζήτησης.*



Εικόνα 4.10: *Αναπαράσταση σύγκρισης αποτελεσμάτων διερεύνησης μεγέθους ακτινικής αναζήτησης.*

Συνολικά training steps σε 14 ώρες και 32 λεπτά:

- beam_size = **8 ->** 40860 steps (8.3 epochs)
- beam_size = **16 ->** 32144 steps (7 epochs)

Από τα αποτελέσματα παρατηρείται μια βελτίωση της επίδοσης του συστήματος αυξάνοντας το μέγεθος της ακτινικής αναζήτησης. Το γεγονός αυτό δικαιολογείται, καθώς με μεγαλύτερο μέγεθος ακτινικής αναζήτησης, αυξάνονται οι επιλογές των λέξεων εξόδου που εξετάζονται και μαζί η πιθανότητα να επιλεχθεί η σωστή αλληλουχία λέξεων. Βέβαια, υπάρχει και μια αύξηση του υπολογιστικού κόστους, καθώς αυξάνονται οι συνολικοί υπολογισμοί και αυτό οδηγεί σε αύξηση του απαιτούμενου χρόνου.

4.5 Συγκρίσεις με παρόμοια συστήματα

Στο παρών κεφάλαιο συγκρίνεται το σύστημα που υλοποιήθηκε κατά τη διάρκεια εκπόνησης της παρούσας πτυχιακής εργασίας, με παρόμοιες υλοποιήσεις συστημάτων αυτόματης παραγωγής περίληψης. Το γεγονός ότι δεν υπάρχει κάποια ερευνητική εργασία που να έχει μελετήσει την αυτόματη παραγωγή περίληψης εστιασμένη στην ελληνική γλώσσα, στερεί τη δυνατότητα μιας πιο αξιόπιστης σύγκρισης. Παρ' όλα αυτά, οι ερευνητικές εργασίες που θα αναφερθούν στη συνέχεια αποτέλεσαν στοιχεία έμπνευσης και στήριξης της παρούσας εργασίας, καθώς οι πολλοί μηχανισμοί που χρησιμοποιήθηκαν κατά την υλοποίηση του συστήματος έχουν ήδη χρησιμοποιηθεί σε αυτές. Για αυτό το λόγο, η χρήση τους ως μέτρο σύγκρισης μπορεί να αποδώσει μια εικόνα της λειτουργικότητας και της αποδοτικότητας του μοντέλου και να αναδείξει αν οι κατευθύνσεις που ακολουθήθηκαν και οι επιλογές που έγιναν, είναι ορθές για την αντίστοιχη έρευνα στην ελληνική γλώσσα. του Συγκεκριμένα, οι ερευνητικές εργασίες που χρησιμοποιούνται ως μέτρο σύγκρισης είναι οι [11], [13], [16], [17].

Κάνοντας μια σύντομη ανασκόπηση των παραπάνω ερευνητικών εργασιών, στη πρώτη κατά σειρά έρευνα [11], εφαρμόστηκε η αρχιτεκτονική ακολουθία σε ακολουθία με κωδικοποιητή ένα συνελεκτικό νευρωνικό δίκτυο και αποκωδικοποιητή ένα κλασσικό feed-forward δίκτυο. Η έρευνα [13] ακολούθησε την ίδια αρχιτεκτονική αλλά χρησιμοποίησε ως αποκωδικοποιητή ένα αναδρομικό δίκτυο, βελτιώνοντας την αποτελεσματικότητα του συστήματος. Η επόμενη προς σύγκριση έρευνα [16], εφάρμοσε πλήρως αναδρομικά δίκτυα σε κωδικοποιητή και αποκωδικοποιητή και εισήγαμε τον μηχανισμό αντιγραφής pointer generator, παρουσιάζοντας πρωτοφανή βελτιωμένα αποτελέσματα σε σχέση με τις προηγούμενες έρευνες. Τέλος, η έρευνα [17] επέλεξε μηχανισμούς ενισχυτικής μάθησης για την εκπαίδευση του μοντέλου και εισήγαγε έναν νέο μηχανισμό προσοχής, βελτιώνοντας τα αποτελέσματα. Όλες οι παραπάνω εργασίες υλοποιήθηκαν με σκοπό την παραγωγή περιλήψεων αγγλικών κειμένων, και συνεπώς εκπαιδεύτηκαν και αξιολογήθηκαν πάνω σε αγγλικά κείμενα.

Στη συνέχεια, παρουσιάζεται ένας πίνακας που περιέχει τα βέλτιστα αποτελέσματα των τεσσάρων προαναφερθέντων εργασιών και της παρούσας εργασίας, με στόχο την ορθότερη και αντικειμενικότερη σύγκρισή τους. Ως μετρικές επίδοσης χρησιμοποιούνται και σε αυτή τη περίπτωση οι μετρικές ROUGE.

	Rouge-1 F1 score	**Rouge-2 F1 score**	**Rouge-L F1 score**
Rush et al. (2015) [11]	0.3100	0.1265	0.2834
Chopra et al. (2016) [13]	0.3378	0.1265	0.3113
See et al. (2017) [16]	0.3953	0.1728	0.3638
Paulus et al (2018) [17]	**0.4116**	0.1575	**0.3908**
Our model (2021)	0.4008	**0.2706**	0.3407

Πίνακας 4.9 *:Σύγκριση βέλτιστων αποτελεσμάτων της παρούσας εργασίας με παρόμοιες υλοποιήσεις.*

Όπως φαίνεται από το παραπάνω πίνακα, τα βέλτιστα αποτελέσματα της παρούσας εργασίας είναι συγκρίσιμα με τα αποτελέσματα των υπόλοιπων ερευνών και σε πολλές περιπτώσεις τα ξεπερνούν κατά πολύ. Συγκεκριμένα, το σύστημα που υλοποιήθηκε στα πλαίσια της παρούσας εργασίας παρουσιάζει τη δεύτερη καλύτερη επίδοση στη μετρική Rouge-1, η οποία εκφράζει το ποσοστό των ορθών μεμονωμένων λέξεων (κοινές λέξεις μεταξύ παραγόμενης περίληψης και προ-υπάρχουσας περίληψης), με σκορ 0.4008. Επίσης εμφανίζει αρκετά καλή επίδοση στη μετρική Rouge-L , η οποία υποδηλώνει το ποσοστό ορθών αλληλουχιών με πάνω από δύο λέξεις, ξεπερνώντας τα αντίστοιχα αποτελέσματα των ερευνών [11] και [13].

Το αξιοσημείωτο όμως αποτέλεσμα εμφανίζεται στη μετρική Rouge-2, η οποία εκφράζει το ποσοστό εμφάνισης φράσεων δύο λέξεων. Η συγκεκριμένη μετρική φτάνει τη τιμή 0.2706 ή αλλιώς το ποσοστό 27.06% και ξεπερνά τις αντίστοιχες τιμές των υπόλοιπων υλοποιήσεων κατά 10% το λιγότερο. Μια επεξήγηση της παραπάνω διαφοράς έγκειται στη φύση της ελληνικής και στις διαφορές της από την αγγλική. Συγκεκριμένα, στην ελληνική γλώσσα λόγω της κλίσης των ουσιαστικών, των επιθέτων και των άρθρων, προκύπτουν αμέτρητες φράσεις που αποτελούνται από το συνδυασμό άρθρο + ουσιαστικό ή άρθρο + επίθετο. Οι λέξεις τέτοιων φράσεων είναι νοηματικά και συντακτικά τόσο στενά συνδεδεμένες μεταξύ τους, που λειτουργούν σαν μια λέξη και συνεπώς εμφανίζονται πολύ συχνά σε ένα ελληνικό κείμενο. Στην αγγλική γλώσσα το πλήθος αντίστοιχων φράσεων είναι πολύ μικρότερος λόγω της ύπαρξης μόνο δυο άρθρων (the,a/an). Ένα χαρακτηριστικό παράδειγμα σύγκρισης αποτελεί το εξής: Έστω ότι παίρνουμε τη λέξη “σκύλος”, συνδυάζοντας αυτό το ουσιαστικό με ένα άρθρο προκύπτουν έξι συνδυασμοί από τις κλίσεις “ο σκύλος, του σκύλου, τον σκύλο, ένας σκύλος, ενός σκύλου, ένα σκύλο”, ενώ οι αντίστοιχοι συνδυασμοί στην αγγλική γλώσσα είναι μόνο δυο “the dog, a dog”. Μία δεύτερη επεξήγηση είναι το γεγονός ότι μέσω της γενικής κλίσης και του εμπρόθετου άρθρου (στο, στη κτλ.), η ελληνική γλώσσα μπορεί να εκφράσει πολλά πράγματα με δύο λέξεις, όταν η αγγλική απαιτεί το λιγότερο τρεις. Π.χ. “στο σπίτι” -> “at the house”, “στο κουτί” -> “in the box”, “της χώρας” -> “of the country”.

Τα αποτελέσματα των παραπάνω συγκρίσεων υποδηλώνουν ότι οι σχεδιαστικές επιλογές και οι αποφάσεις που πάρθηκαν κατά την υλοποίηση του συγκεκριμένου συστήματος αυτόματης παραγωγής περίληψης ελληνικών κειμένων, ήταν ορθές και οδήγησαν σε ένα αρκετά αξιόπιστο και αποδοτικό αποτέλεσμα.

5. Συμπεράσματα και προοπτικές εξέλιξης

5.1 Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας πτυχιακής εργασίας και από όλη την ανάλυση που παρουσιάστηκε στα προηγούμενα κεφάλαια.

Από τα αποτελέσματα που παρουσιάστηκαν, μπορεί κανείς να συμπεράνει ότι η ανάπτυξη ενός αποδοτικού συστήματος αυτόματης παραγωγής περίληψης για την ελληνική γλώσσα είναι ένα αρκετά περίπλοκο πρόβλημα, αλλά και πολύ ενδιαφέρον. Στη παρούσα εργασία, ο σχεδιασμός του συστήματος βασίστηκε στην εφαρμογή μηχανισμών βαθιάς μάθησης και στον συνδυασμό διάφορων μεθόδων που έχουν ήδη εφαρμοστεί, είτε σε παρόμοια συστήματα για την αγγλική γλώσσα, είτε σε άλλα προβλήματα επεξεργασίας φυσικής γλώσσας. Το σίγουρο είναι ότι η χρήση των αναδρομικών νευρωνικών δικτύων και της αρχιτεκτονικής ακολουθία σε ακολουθία με έναν κωδικοποιητή και έναν αποκωδικοποιητή, αποτελεί μια γερή βάση για τον σχεδιασμό ενός τέτοιου απαιτητικού συστήματος.

Ιδιαίτερη σημασία όμως πρέπει να δοθεί στις ιδιαιτερότητες που έχει η γλώσσα για την οποία αναπτύσσεται το σύστημα. Στη παρούσα εργασία, ερευνήθηκαν αρκετές πτυχές της ελληνικής γλώσσας και εντοπίστηκαν οι βασικότερες ιδιομορφίες και οι διαφορές της με την αγγλική. Για να μπορέσει το σύστημα να κατανοήσει καλύτερα την ελληνική γλώσσα, πραγματοποιήθηκαν προσαρμογές και προσθήκες μηχανισμών σε δύο στάδια. Το πρώτο στάδιο είναι αυτό της προ-επεξεργασίας, όπου εφαρμόστηκαν συγκεκριμένα βήματα, όπως η αφαίρεση μη ελληνικών χαρακτήρων και η σύντμηση των λέξεων με απόστροφο. Το δεύτερο και βασικότερο στάδιο αποτελεί αυτό των αριθμητικών αναπαραστάσεων των λέξεων (word embeddings), όπου προστέθηκαν σημαντικές πληροφορίες για την συντακτική και γραμματική υπόσταση της κάθε λέξης.

Από τη διερεύνηση της επίδρασης της προ-επεξεργασίας των δεδομένων στην επίδοση του συστήματος, επαληθεύεται η μεγάλη αξία αυτής της διαδικασίας. Συγκεκριμένα, η προ-επεξεργασία του συνόλου δεδομένων επέφερε πολύ βελτιωμένα αποτελέσματα σε σχέση με τη χρήση των δεδομένων στη πρωτότυπη μορφή τους. Κάθε στάδιο προ-επεξεργασία που εφαρμόστηκε στα πλαίσια της παρούσας εργασίας, είχε ως σκοπό την ελαχιστοποίηση κάθε μορφής θορύβου που μπορεί να συναντηθεί στα ελληνικά κείμενα. Είναι λοιπόν πολύ

σημαντικό, σε περίπτωση υλοποίησης παρόμοιου συστήματος να δίνεται προσοχή στη ποιότητα των δεδομένων και στην διαδικασία της προ-επεξεργασίας τους, καθώς λόγω της αναδρομικής φύσης του συστήματος, ένα στοιχείο θορύβου επηρεάζει συλλογικά όλη την ακολουθία εξόδου.

Παρατηρώντας τα αποτελέσματα από τα πειράματα που πραγματοποιήθηκαν για διάφορες σημαντικές παραμέτρους του συστήματος, συμπεραίνει κανείς ότι προκύπτουν αρκετά σχεδιαστικά διλήμματα. Ένα από αυτά είναι η επιλογή μικρού ή μεγάλου μεγέθους κρυφών στρωμάτων των νευρωνικών δικτύων του συστήματος. Η παράμετρος αυτή είναι βασική και επηρεάζει σημαντικά την αποτελεσματικότητα του συστήματος. Από τη διερεύνηση αυτής της παραμέτρου προέκυψε ότι η αύξηση των κρυφών καταστάσεων οδηγεί εν μέρη σε καλύτερα αποτελέσματα, αλλά αυξάνει ταυτοχρόνως κατά πολύ τις απαιτήσεις του συστήματος σε υπολογιστικούς πόρους και χρόνο. Για αυτό το λόγο η επιλογή της συγκεκριμένης παραμέτρου εξαρτάται από τις απαιτήσεις και τις προσδοκίες του χρήστη από το σύστημα.

Αντίθετα με την αρχική πρόβλεψη που μπορεί να κάνει κάποιος σχετικά με την επίδραση του μεγέθους του λεξιλογίου, ένα μεγαλύτερο λεξιλόγιο αποδεικνύεται ότι δεν οδηγεί υποχρεωτικά σε καλύτερα αποτελέσματα. Από την άλλη όμως, δημιουργεί σίγουρα ένα πιο γενικευμένο μοντέλο και αυξάνει τη δυνατότητα παραγωγής περιλήψεων κειμένων διαφορετικού ύφους και θέματος. Όσον αφορά τα μεγέθη του αρχικού κειμένου και της παραγόμενης περίληψης, παρατηρήθηκαν σχετικά μικρές διαφορές στα αποτελέσματα των μετρικών ROUGE, αλλά μεγάλες διαφορές στον απαιτούμενο χρόνο εκπαίδευσης του μοντέλου. Τα αποτελέσματα αυτά δείχνουν ότι για τη συγκεκριμένη παράμετρο, ίσως είναι καλύτερη σχεδιαστική απόφαση η επιλογή μικρών μεγεθών. Βέβαια η επιλογή της συγκεκριμένης παραμέτρου εξαρτάται σε μεγάλο βαθμό από την εκάστοτε χρήση του συστήματος και αν επιδιώκεται περίληψη μεγάλων ή μικρών κειμένων.

Η αντιστρόφως ανάλογη σχέση της επίδοσης του συστήματος και των υπολογιστικών - χρονικών απαιτήσεων του, παρατηρείται και στη διερεύνηση του μεγέθους της ακτινικής αναζήτησης. Συγκεκριμένα, μεγαλύτερο μέγεθος οδηγεί σε καλύτερα αποτελέσματα, αλλά και σε αυξημένες απαιτήσεις. Τα αποτελέσματα του συστήματος που υλοποιήθηκε στα πλαίσια της παρούσας εργασίας αποδείχθηκαν συγκρίσιμα και σε αρκετές περιπτώσεις καλύτερα από παρόμοιες υλοποιήσεις συστημάτων στην αγγλική γλώσσα.

5.2 Προοπτικές εξέλιξης

5.2.1 Ενσωμάτωση μοντέλου σε mobile application

Στη σημερινή εποχή της ραγδαίας τεχνολογικής ανάπτυξης, οι έξυπνες συσκευές έχουν μπει για τα καλά στη καθημερινότητα του ανθρώπου. Με το πιο χαρακτηριστικό παράδειγμα να αποτελεί αυτό των έξυπνων κινητών τηλεφώνων (smartphones), τα οποία χρησιμοποιούνται από το 48.37% (~3.5 δισεκατομμύρια ανθρώπων) του συνολικού πληθυσμού του πλανήτη. Ένα βασικό κομμάτι των έξυπνων κινητών συσκευών αποτελούν οι έξυπνες εφαρμογές που τρέχουν σε αυτά, καθώς ο μέσος χρήστης ενός smartphone καταναλώνει το 89% του συνολικού χρόνου που περνά πάνω από τη συσκευή σε έξυπνες εφαρμογές [37]. Για αυτό το λόγο προτείνεται ένας τρόπος ενσωμάτωσης του συστήματος που υλοποιήθηκε στα πλαίσια της συγκεκριμένης εργασίας, σε μια έξυπνη εφαρμογή. Επίσης, η διαχρονική αξία της περίληψης και η ανάγκη της χρήσης της, ενθαρρύνουν ακόμα περισσότερο την ανάπτυξη μιας τέτοιας εφαρμογής.

Ένα μοντέλο που στηρίζεται σε μηχανισμούς βαθιάς μηχανικής μάθησης και περιέχει βαθιά νευρωνικά δίκτυα, χρειάζεται πολλούς υπολογιστικούς πόρους, μεγάλη μνήμη και εμφανίζει υψηλή κατανάλωση. Τα έξυπνα κινητά και γενικότερα οι έξυπνες συσκευές αδυνατούν να παρέχουν αυτούς τους πόρους, καθώς βασίζονται σε πιο αδύναμους επεξεργαστές και χρησιμοποιούν μπαταρίες για την λειτουργία τους. Συνεπώς, οι πιο απαιτητικές διαδικασίες, που είναι η εκπαίδευση και η αξιολόγηση του μοντέλου είναι σχεδόν αδύνατο να εφαρμοστούν σε τέτοιες συσκευές με περιορισμένους υπολογιστικούς πόρους. Αυτό που μπορεί να εφαρμοστεί όμως είναι η διαδικασία του inference, δηλαδή της χρήσης του μοντέλου για την παραγωγή της περίληψης ενός νέου μεμονωμένου κειμένου. Για να πραγματοποιηθεί αυτό πρέπει να υπάρχει ένα προ-εκπαιδευμένο μοντέλο, το οποίο θα χρησιμοποιεί τις αποθηκευμένες μεταβλητές και τα βάρη των νευρωνικών δικτύων που υπολογίστηκαν κατά την εκπαίδευση, για να παράγει τη ζητούμενη περίληψη. Απαιτείται όμως και ένας τρόπος με τον οποίο θα γίνει η κατάλληλη μετατροπή του προ-εκπαιδευμένου μοντέλου σε μορφή ικανή να υποστηριχθεί από τα λειτουργικά συστήματα των smartphones, όπως Android ή IOS.

Αυτό το πρόβλημα έρχεται να λύσει το **TensorFlow Lite** [38]. Το TensorFlow Lite είναι ένα σύνολο εργαλείων προγραμματισμού που επιτρέπει την εκτέλεση μοντέλων μηχανικής μάθησης σε κινητές συσκευές και μικροεπεξεργαστές. Βασική προυπόθεση αποτελεί τα μοντέλα αυτά να έχουν αναπτυχθεί με τη χρήση του TensorFlow. Συγκεκριμένα, με τη χρήση του TensorFlow Lite δίνεται η δυνατότητα εφαρμογής του υλοποιημένου συστήματος αυτόματης παραγωγή περίληψης σε smartphones, για την εξαγωγή περιλήψεων νέων κειμένων με σχετικά μικρή καθυστέρηση. Το TensorFlow Lite αποτελείται από δύο βασικά εργαλεία:

- Τον μετατροπέα **TensorFlow Lite converter**, ο οποίος μετατρέπει ένα TensorFlow μοντέλο στη κατάλληλη μορφή για να μπορεί να χρησιμοποιηθεί από τον διερμηνέα και πραγματοποιεί βελτιστοποιήσεις για τη μείωση του μεγέθους του εκτελέσιμου και τη βελτίωση της επίδοσης.

- Τον διερμηνέα **TensorFlow Lite interpreter**, ο οποίος εκτελεί τα βελτιστοποιημένα μοντέλα που προκύπτουν από τον μετατροπέα σε διάφορα υπολογιστικά συστήματα, όπως κινητές συσκευές, ενσωματωμένα συστήματα και μικροεπεξεργαστές.

Για την ενθάρρυνση της ανάπτυξης μιας έξυπνης εφαρμογής, η οποία θα χρησιμοποιεί το σύστημα που υλοποιήθηκε στη συγκεκριμένη εργασία, περιγράφονται τα βήματα που πρέπει να ακολουθήσει κάποιος για να ενσωματώσει το σύστημα στην εφαρμογή χρησιμοποιώντας το TensorFlow Lite:

1. **Επιλογή του βέλτιστου προ-εκπαιδευμένου μοντέλου :** Συνίσταται η επιλογή του μοντέλου που παρουσίασε τα καλύτερα αποτελέσματα κατά την διερεύνηση που πραγματοποιήθηκε στο προηγούμενο κεφάλαιο. Τα αρχεία που προέκυψαν μετά από την εκπαίδευση και την αξιολόγηση είναι τρία. Ένα .meta αρχείο που περιέχει τον συνολικό TensorFlow γράφο του μοντέλου με όλες της μεταβλητές-παραμέτρους και τις λειτουργίες του. Ένα .ckp εκτελέσιμο που περιέχει όλα τα ενημερωμένα βάρη και τις μεταβλητές και σταθερές τιμές των νευρωνικών δικτύων του μοντέλου, όπως έχουν αυτά προκύψει μετά την εκπαίδευση. Και τέλος ένα checkpoint αρχείο που απλά καταγράφει τη τελευταία ανανέωση των παραπάνω αρχείων. Σε περίπτωση που κάποιος επιθυμεί να εκπαιδεύσει ξανά το σύστημα προσαρμόζοντας τις παραμέτρους του στις δικές του ανάγκες , μπορεί να το κάνει και να χρησιμοποιήσει τα αντίστοιχα παραγόμενα αρχεία.
2. **Μετατροπή του μοντέλου:** Επόμενο βήμα είναι η μετατροπή του μοντέλου στη κατάλληλη μορφή, δηλαδή από TensorFlow σε TensorFlow Lite, με τη χρήση του εργαλείου TensorFlow Lite converter και μερικών γραμμών κώδικα Python. Συγκεκριμένα, ο μετατροπέας επεξεργάζεται τα αρχεία που αναφέρθηκαν στο προηγούμενο βήμα και παράγει ένα .tflite εκτελέσιμο.
3. **Εκτέλεση του μοντέλου στη συσκευή:** Με τη χρήση του διερμηνέα TensorFlow Lite interpreter και της διεπαφής προγραμματισμού εφαρμογών (API) που παρέχει η TensorFlow Lite, γίνεται η ενσωμάτωση του μοντέλου στην έξυπνη εφαρμογή και κατ' επέκταση η εκτέλεση του στο smartphone. Η TensorFlow Lite παρέχει API και βιβλιοθήκες λογισμικού για τις πιο διαδεδομένες γλώσσες προγραμματισμού όπως Python, Java, C++, C# και Swift. Οπότε η έξυπνη εφαρμογή που θα "φιλοξενήσει" το μοντέλο μπορεί να υλοποιηθεί σε οποιαδήποτε από αυτές τις γλώσσες.
4. **Βελτιστοποίηση του μοντέλου :** Στο τέλος μπορεί να πραγματοποιηθεί κάποια βελτιστοποίηση στο μοντέλο με σκοπό τη μείωση της μνήμης που καταλαμβάνει, και της καθυστέρησης που παρουσιάζει η εκτέλεση του σε ένα smartphone, μειώνοντας όμως λίγο την απόδοσή του. Μία συχνή μέθοδος βελτιστοποίησης που χρησιμοποιεί το TensorFlow Lite είναι αυτή του κβαντισμού (quantization). Με το κβαντισμό μειώνεται η ακρίβεια των τιμών των παραμέτρων του μοντέλου, ο οποίες από 32 bit floating point αριθμοί μετατρέπονται σε 8 bit αριθμούς. Αυτό μειώνει το μέγεθος του μοντέλου και επιτυγχάνει γρηγορότερους υπολογισμούς.

Η προς υλοποίηση έξυπνη εφαρμογή, που μπορεί να αναπτυχθεί για λειτουργικά συστήματα Android ή IOS, θα δίνει στο χρήστη τη δυνατότητα να εισάγει το κείμενο που επιθυμεί να συντομεύσει και θα παρέχει ως έξοδο την επιθυμητή περίληψη. Ένα παράδειγμα λειτουργίας μιας τέτοιας εφαρμογής περιγράφεται από τα παρακάτω βήματα:

1. Ο χρήστης εισάγει το κείμενο από το οποίο επιθυμεί τη περίληψη. Το κείμενο μπορεί να εισαχθεί είτε απευθείας μέσω της πληκτρολόγησης του από τον χρήστη, είτε σε μορφή κάποιου αρχείου (π.χ. word, txt).
2. Αφού περάσουν κάποια δευτερόλεπτα, ο χρήστης λαμβάνει την επιθυμητή περίληψη είτε απευθείας στην οθόνη της συσκευής του, είτε σε μορφή κάποιου αρχείου. Το αρχείο αυτό θα αποθηκεύεται αυτόματα στο κινητό του ή/και θα στέλνεται αυτόματα στο email του.

Επισημαίνεται, ότι πολύ σημαντικό ρόλο σε μία έξυπνη εφαρμογή έχει το user experience δηλαδή το πόσο ευχάριστη και εύκολη είναι η χρήση της, οπότε αρκετή έμφαση πρέπει να δοθεί και σε αυτό το κομμάτι.

5.2.2 Μελλοντικές κατευθύνσεις έρευνας

Στη συγκεκριμένη εργασία σχεδιάστηκε και υλοποιήθηκε ένα πρωτοποριακό σύστημα αυτόματης παραγωγής περίληψης για την ελληνική γλώσσα. Επίσης πραγματοποιήθηκε ανάλυση των σχεδιαστικών επιλογών που πάρθηκαν, διερευνώντας εκτεταμένα συγκεκριμένες παραμέτρους του συστήματος που θεωρούνται βασικοί για τον καθορισμό της επίδοσης του. Αν και οι διερευνήσεις που έγιναν επέφεραν αρκετά χρήσιμα συμπεράσματα, η συνολική διερεύνηση του συστήματος δε μπορεί να χαρακτηριστεί ιδανικά ολοκληρωμένη, καθώς υπάρχουν ακόμη πολλές πτυχές του συγκεκριμένου προβλήματος που χρήζουν περεταίρω έρευνα. λόγω διάφορων περιορισμών που προέκυψαν. Ο περιορισμένος αριθμός δεδομένων και ο περιορισμένος χρόνος είναι κάποιοι παράγοντες που εμποδίζουν τη διεξαγωγή περεταίρω διερευνών, ιδαίτερα στη περίπτωση μοντέλων βαθιάς μάθησης. Αυτοί οι περιορισμοί έπαιξαν σημαντικό ρόλο και στην εκπόνηση της παρούσα εργασίας. Παρ’ όλα αυτά το συγκεκριμένο σύστημα, όντας το πρώτο σύστημα αυτόματης παραγωγής περίληψης για ελληνικά κείμενα, μπορεί να αποτελέσει ένα αρχικό στάδιο για την ανάπτυξη αποτελεσματικότερων παρόμοιων συστημάτων. Στη συνέχεια προτείνονται κάποιες κατευθύνσεις με σκοπό την προώθηση της έρευνας πάνω στη συγκεκριμένη εργασία και την επέκτασή της με στόχο την εξαγωγή καλύτερων ερευνητικών αποτελεσμάτων.

Μια αρκετά ενδιαφέρουσα κατεύθυνση αποτελεί η δημιουργία ενός πιο γενικευμένου συστήματος, επεκτείνοντας την ικανότητα του να κατανοεί ελληνικά κείμενα και να παράγει τη περίληψη τους, σε μεγαλύτερο φάσμα της ελληνικής γλώσσας. Για να επιτευχθεί αυτό απαιτείται ένα πολύ μεγάλο σύνολο δεδομένων αποτελούμενο από κείμενα διάφορων θεματικών ενοτήτων. Πηγαίνοντας ένα βήμα πιο πέρα, μπορεί να εξεταστεί η επέκταση της

συγκεκριμένης εργασίας με σκοπό τη δημιουργία ενός συστήματος για παραπάνω από μια γλώσσες. Η συγκεκριμένη υλοποίηση αποτελεί ένα αρκετά δύσκολο πρόβλημα, καθώς εκτός από το συνδυασμό συνόλων κειμένων διαφορετικών γλωσσών, απαιτεί τη διερεύνηση των ιδιομορφιών της κάθε γλώσσας και την ενσωμάτωση τους στο σύστημα.

Πολλές δυνατότητες έρευνας εμφανίζονται στη μελέτη των διαφορετικών παραμέτρων του συστήματος και στη διερεύνηση της επίδρασής τους σε αυτό. Επιπλέον, το υλοποιήσιμο σύστημα αποτελείται από πολλούς διαφορετικούς μηχανισμούς με τους οποίους μπορεί κάποιος να πειραματιστεί για την εύρεση καλύτερων αποτελεσμάτων.

- χρήση διαφορετικού τύπου αναδρομικών νευρωνικών δικτύων, όπως GRU
- χρήση συνδυασμού διαφόρων ειδών νευρωνικών δικτύων, όπως συνελεκτικά νευρωνικά δίκτυα με αναδρομικά νευρωνικά δίκτυα
- πειραματισμός με διαφορετικούς μηχανισμούς προσοχής
- εξέταση πρόσθεσης παραπάνω πληροφοριών στα word embeddings
- δοκιμή εναλλακτικών στρατηγικών επιλογής βέλτιστης εξόδου, αντί της ακτινικής αναζήτησης
- πειραματισμός με διαφορετικές μεθόδους εκπαίδευσης του συστήματος, όπως ενισχυτική μάθηση

Λαμβάνοντας υπόψιν ότι η έξοδος του υλοποιήσιμου συστήματος αποτελεί ένα κείμενο, δημιουργούνται νέες προοπτικές έρευνας. Συγκεκριμένα, το σύστημα μπορεί να ενσωματωθεί σε άλλα προβλήματα επεξεργασίας της φυσικής γλώσσας και να χρησιμοποιηθεί σαν ενδιάμεσο στάδιο επεξεργασία, όπου από ένα μεγάλης έκτασης κείμενο θα εξάγει το σημαντικότερο κομμάτι του (περίληψη), το οποίο θα χρησιμοποιηθεί στη συνέχεια για τους σκοπούς του εκάστοτε συστήματος. Κάποια παραδείγματα στα οποία θα μπορούσε να διερευνηθεί αυτή η χρήση του συστήματος είναι τα εξής:

- υλοποίηση συστήματος αυτόματης κατηγοριοποίησης ελληνικών κειμένων μεγάλης έκτασης (text categorization)
- υλοποίηση συστήματος εξαγωγής συναισθημάτων από ελληνικά κείμενα μεγάλης έκτασης (sentiment analysis)

Παράρτημα Α

Παράδειγμα δοκιμής του συστήματος και αξιολόγησης των αποτελεσμάτων:

Πηγή: https://www.liberal.gr/politics/em-makron-i-gallia-sto-pleuro-tis-elladas-otan-apeileitai-stin-anat-mesogeio/199976

ARTICLE : Μήνυμα συμπαράστασης προς την Ελλάδα έστειλε από το Στρασβούργο ο πρόεδρος της Γαλλίας, Εμμανουέλ Μακρόν, «φωτογραφίζοντας» ουσιαστικά τις απειλές της Τουρκίας απέναντι σε Ελλάδα και Κύπρο και διαμηνύοντας ότι η χώρα του θα στηρίξει τους δύο εταίρους της. Στο πλαίσιο της ομιλίας του στην ολομέλεια της ευρωβουλής για το μέλλον της Ευρώπης, Ο Ε Μακρόν τόνισε πως η Γαλλία πιστεύει στην ευρωπαϊκή αμυντική πολιτική και υπογράμμισε ότι η χώρα του θα είναι ανά πάσα στιγμή συμπαραστάτης κάθε κράτους-μέλους της Ευρωπαϊκής Ένωσης όταν απειλείται η κυριαρχία του ή υφίσταται επίθεση. Τη θέση αυτή όπως είπε, έχει μεταφέρει τόσο στη μεγάλη Βρετανία με αφορμή την υπόθεση Σκριπάλ, όσο και προς την Ελλάδα συγκεκριμένα, υπογράμμισε πως είναι πάγια θέση της Γαλλίας η υποστήριξη της Ελλάδας, όταν απειλείται στην Ανατολική Μεσόγειο. Κι αυτό, όπως ανέφερε, συζητήθηκε πρόσφατα στην τηλεφωνική επικοινωνία του με τον Έλληνα πρωθυπουργό, Αλέξη Τσίπρα, καθώς αυτή η πολιτική βρίσκεται ακριβώς στην καρδιά της έννοιας της αλληλεγγύης μεταξύ των ευρωπαϊκών κρατών.
REFERENCE SUMMARY: Ο πρόεδρος της Γαλλίας μιλώντας στην ολομέλεια της ευρωβουλής και απαντώντας σε ερωτήσεις βουλευτών, φωτογράφησε ουσιαστικά τις απειλές της Τουρκίας απέναντι σε Ελλάδα και Κύπρο, διαμηνύοντας ότι η χώρα του θα στηρίξει τους δύο εταίρους της.
GENERATED SUMMARY: σε απειλές της τουρκίας απέναντι σε ελλάδα και κύπρο και τη στήριξη της χώρας σε δύο εταίρους ανέφερε ο πρόεδρος της γαλλίας στο πλαίσιο της ομιλίας του ολομέλεια της ευρωβουλής για το μέλλον της ευρώπης
[{**'rouge-1'**: {'f': 0.6086956473513968, 'p': 0.75, 'r': 0.5121951219512195}, **'rouge-2':** {'f': 0.4999999951125001, 'p': 0.5882352941176471, 'r': 0.43478260869565216}, **'rouge-l':** {'f': 0.4882638407446772, 'p': 0.6428571428571429, 'r': 0.4390243902439024}}]

Παράδειγμα δοκιμής του συστήματος και αξιολόγησης των αποτελεσμάτων:

Πηγή: https://www.tovima.gr/2018/04/18/finance/me-tin-xrisi-kartas-tha-mporoyn-na-plirwnoyn-oi-forologoymenoi/

ARTICLE : Απευθείας με τη χρήση καρτών πληρωμών θα μπορούν οι φορολογούμενοι να εξοφλούν τις φορολογικές υποχρεώσεις τους εντός των προσεχών ημερών. Αναμένεται η δημοσίευση στην εφημερίδα της κυβερνήσεως της απόφασης που υπέγραψε ο διοικητής της ανεξάρτητης αρχής δημοσίων εσόδων Γιώργος Πιτσιλής, μέσω της οποίας η ΑΑΔΕ θα παρέχει στους φορολογουμένους τη δυνατότητα να πληρώνουν βεβαιωμένες αρρύθμιστες οφειλές φυσικών και νομικών προσώπων με τη χρήση καρτών πληρωμών. Η πληρωμή όπως αναφέρεται σε ανακοίνωση της ΑΑΔΕ θα πραγματοποιείται μέσω της υπηρεσίας «Προσωποποιημένη Πληροφόρηση» του στη διαδικτυακή πύλη της ΑΑΔΕ. Με τη νέα παρεχόμενη υπηρεσία είναι δυνατή η πληρωμή πολλών οφειλών σε μία συναλλαγή και με την επιτυχή ολοκλήρωσή της οι οφειλές πιστώνονται άμεσα.
REFERENCE SUMMARY: Εντός των προσεχών ημερών αναμένεται η δημοσίευση στην εφημερίδα της κυβερνήσεως, της απόφασης που υπέγραψε ο διοικητής της ανεξάρτητης αρχής δημοσίων εσόδων Γιώργος Πιτσιλής
GENERATED SUMMARY: αναμένεται η δημοσίευση κυβερνήσεως της απόφασης που υπέγραψε ο διοικητής ανεξάρτητης αρχής δημοσίων εσόδων γιώργος πιτσιλής με οποία η ααδε να παρέχει τους φορολογουμένους δυνατότητα να πληρώνουν αρρύθμιστες οφειλές
ROUGE EVALUATION: [{'rouge-1': {'f': 0.6666666618381345, 'p': 0.5625, 'r': 0.8181818181818182}, 'rouge-2': {'f': 0.6551724090071345, 'p': 0.5428571428571428, 'r': 0.8260869565217391}, 'rouge-l': {'f': 0.6252072968483955, 'p': 0.5625, 'r': 0.8181818181818182}}]

Παράρτημα Β

Μέρος λογισμικού υλοποίησης της εισαγωγή του συνόλου δεδομένων στο σύστημα και της προ-επεξεργασίας τους, με χρήση της γλώσσας Python:

```
import sys
import csv
import io
import pandas as pd
import numpy as np
import re
import html as ihtml
from bs4 import BeautifulSoup
import time
import unicodedata
import os
import collections
input_dir = 'drive/MyDrive/Diplomatiki_sum/Data'
file_data = os.path.join(input_dir,'website_1_dataset_215385rows.csv')

#----------Importing the Greek gathered dat into the system-----------#
#Reading of the greek text data (title|text|summary) of online greek news art
icles.
start_time_read = time.time()
title=[]
text=[]
summary=[]
#Read the desired columns
title=pd.read_csv(file_data,sep='\t',usecols=["title_alt"], skip_blank_lines=
True)
print(title.info())
text=pd.read_csv(file_data,sep='\t',usecols=["description"], skip_blank_lines
=True)
print(text.info())
summary=pd.read_csv(file_data,sep='\t',usecols=["short_description"], skip_bl
ank_lines=True)
print(summary.info())
#Create a dataframe with all data
raw_data=pd.concat([title,text,summary], axis=1, keys=['Title', 'Text', 'Summ
ary'])
print (raw_data.info())
#Remove duplicates and NaN - empty rows.
raw_data=raw_data.dropna()
raw_data=raw_data.drop_duplicates()
```

```
print (raw_data.info())
print(raw_data['Title'].iloc[383])
print(raw_data['Text'].iloc[383])
print(raw_data['Summary'].iloc[383])
print("--- %s seconds ---" % (time.time() - start_time_read))

#--------------Pre-processing/Cleaning Data Function------------#
#Clean all data (Text,Title,Summary) and gather all contractions (apostrofos
use in Greek)
def data_cleaner1(raw_text,c_text):
    #Remove HTML/Javascript tags
    text=ihtml.unescape(raw_text)
    text=BeautifulSoup(text,"lxml").get_text()
    #Remove nonGreek words
    text=re.sub("[a-zA-Z]",'',text)
    #Lowercase all words
    text=text.lower()
    #Remove text in parenthesis
    text=re.sub(r'\([^)]*\)', '', text)
    #Remove special characters
    text=re.sub("[\\\]",' ',text)
    text=re.sub("[\^\\/\{\}\[\]!%__@#$&\*\-_<>-:\"<>+=:«»…]",'',text)
    #Remove multiple spaces
    text = re.sub("(\s+)",' ',text)
    #Get all the contractions (apostrofos use in Greek)
    text=re.sub(r'\’\s','’',text)
    text=re.sub(r'\'\s','\'',text)
    st=re.findall(r'\w+\'\w+',text)
    if st: c_text = c_text.append(st)
    return text

#------------Apply the Pre-processing function---------#
start_time_clean1=time.time()
apostrofos=[]
temp=[]
#Cleaning the texts
for index,rows in raw_data['Text'].iterrows():
    rows = data_cleaner1(str(rows),apostrofos)
    temp.append(rows)
pre_clean_text=pd.DataFrame(temp)
#Cleaning the tiltes
del temp
temp=[]
for index,rows in raw_data['Title'].iterrows():
    rows = data_cleaner1(str(rows),apostrofos)
```

```
    temp.append(rows)
pre_clean_title=pd.DataFrame(temp)
#Cleaning the summaries
del temp
temp=[]
for index,rows in raw_data['Summary'].iterrows():
    rows = data_cleaner1(str(rows),apostrofos)
    temp.append(rows)
pre_clean_sum=pd.DataFrame(temp)
print("--- %s seconds ---" % (time.time() - start_time_clean1))

#Create a dataframe of all clean data
pre_clean_data=pd.concat([pre_clean_title,pre_clean_text,pre_clean_sum], axis
=1, keys=['Title', 'Text', 'Summary'])
print (pre_clean_data.info())
#Find and delete all data that has non Greek text
# and rubbish and are empty after the cleaning.
pre_clean_data['Text'].replace(' ', np.nan, inplace=True)
pre_clean_data['Title'].replace(' ', np.nan, inplace=True)
pre_clean_data['Summary'].replace(' ', np.nan, inplace=True)
pre_clean_data.dropna(inplace=True)
print(pre_clean_data['Text'].iloc[66350])
print(pre_clean_data['Title'].iloc[66350])
print(pre_clean_data['Summary'].iloc[66350])
print(pre_clean_data.info())
```

Μέρος λογισμικού υλοποίησης των αριθμητικών αναπαραστάσεων των λέξεων (word embeddings):

```
import pandas as pd
import numpy as np
import os
import pandas
import io
import sys
import re
import numpy as np
import pickle
import csv
```

```
import timeit
import random
import spacy
!python -m spacy download el_core_news_md
import el_core_news_md
nlp=el_core_news_md.load()

#----------Function to Generate TF-IDF features---------#
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
def tf_idf_generate(sentences):
    from sklearn.feature_extraction.text import CountVectorizer, TfidfTransfo
rmer
    #All the available text
    data = sentences
    cv = CountVectorizer()
    #Convert text data into term-frequency matrix
    data = cv.fit_transform(data)
    tfidf_transformer = TfidfTransformer()
    #Convert term-frequency matrix into tf-idf
    tfidf_matrix = tfidf_transformer.fit_transform(data)
    #Create dictionary to find a TF-IDF value for each word
    word2tfidf = dict(zip(cv.get_feature_names(), tfidf_transformer.idf_))
    return word2tfidf

#-------------Function to Generate POS tags-------------#
from nltk import Tree
def get_pos_tags_dict(word_dict):
    pos_list ={}
    for word in word_dict.keys():
        doc=nlp(word)
        for token in doc:
            pos_list[word] = token.pos_
    df = pd.DataFrame(list(pos_list.items()))
    df.columns = ['word', 'pos']
    df.pos = pd.Categorical(df.pos)
    df['code'] = df.pos.cat.codes
    pos_list ={}
    for index, row in df.iterrows():
       pos_list[row['word']] = row['code']
    return pos_list

#-------------Function to Generate DEP tags-------------#
def get_dep_tags_dict(sentence_list):
```

```
    dep_list ={}
    for sentence in sentence_list:
        doc=nlp(sentence)
        for word in doc:
            dep_list[word] = word.dep_
            print(word)
            print(word.dep_)
    df = pd.DataFrame(list(dep_list.items()))
    df.columns = ['word', 'dep']
    df.dep = pd.Categorical(df.dep)
    df['code'] = df.dep.cat.codes
    deps=df["code"]
    dep_list ={}
    for index, row in df.iterrows():
       dep_list[row['word']] = row['code']
    return dep_list,deps

#-----Function to get initial word embeddings and add extra feautures----#
def get_init_embeds(model, text_list, summary_list, embedding_size):
    print("Loading Lists...")
    #Compute TF-IDF features for each words
    print("Loading TF-IDF...")
    tf_idf_list = tf_idf_generate(text_list+summary_list)
    #Compute POS tags for each words
    print("Loading Pos Tags...")
    pos_list = get_pos_tags_dict(model.wv.vocab)
    #Compute DEP tags for each words
    print("Loading Dep Tags...")
    dep_list = get_dep_tags_dict(text_list[:1000])
    used_words = 0
    word_vec_list = list()
    word_embs ={}
    #Add the extra feaurures at the end of the word embeddings
    #if a word dont have a feaure add zeros at the end
    for word in sorted(model.wv.vocab.keys()):
        try:
            word_vec = model.wv[word]
            if word in tf_idf_list:
              v= tf_idf_list[word]
              rich_feature_array = np.array([v,v,v,v,v,v,v,v,v,v])
              word_vec = np.append(word_vec, rich_feature_array)
              word_embs[word]=word_vec
            else:
              v=0
              rich_feature_array = np.array([v,v,v,v,v,v,v,v,v,v])
```

```
            word_vec = np.append(word_vec, rich_feature_array)
            word_embs[word]=word_vec
          if word in pos_list:
            v=pos_list[word]
            rich_feature_array_2 = np.array([v,v,v,v,v,v,v,v,v,v])
            word_vec = np.append(word_vec, rich_feature_array_2)
            word_embs[word]=word_vec
          else:
            v=0
            rich_feature_array_2 = np.array([v,v,v,v,v,v,v,v,v,v])
            word_vec = np.append(word_vec, rich_feature_array_2)
            word_embs[word]=word_vec
          if word in dep_list:
            v=dep_list[word]
            rich_feature_array_3 = np.array([v,v,v,v,v,v,v,v,v,v])
            word_vec = np.append(word_vec, rich_feature_array_3)
            word_embs[word]=word_vec
          else:
            v=0
            rich_feature_array_3 = np.array([v,v,v,v,v,v,v,v,v,v])
            word_vec = np.append(word_vec, rich_feature_array_3)
            word_embs[word]=word_vec
          used_words += 1
        except KeyError:
            word_vec = np.zeros([embedding_size], dtype=np.float32)
            word_embs[word]=word_vec
        word_vec_list.append(np.array(word_vec))
    return word_embs, np.array(word_vec_list)


word_embs={}
word_vecs_array=list()
word_embs,word_vecs_array=get_init_embeds(model,text_list,summary_list, 150)
```

Βιβλιογραφία

[1] Samrat Babar, M. Tech-Cse and Rit, “Text Summarization: An Overwiew“, 2013

[2] Dr. Michael J. Garbade , “A Quick Introduction to Text Summarization in Machine Learning”, 19/09/2018 [Ηλεκτρονικό]. Available: https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f

[3] Weihs, “Abstract (summary)”, 2004 [Ηλεκτρονικό]. Available: https://www.newworldencyclopedia.org/entry/Abstract_(summary)

[4] «Summarizing» [Ηλεκτρονικό]. Available: https://www.readingrockets.org/strategies/summarizing

[5] A. Kołcz, V. Prabakarmurthi, and J. Kalita, “Summarization as feature selection for text categorization,” in *International Conference on Information and Knowledge Management, Proceedings*, 2001.

[6] J. Tan, X. Wan, and J. Xiao, “From neural sentence summarization to headline generation: A coarse-to-fine approach,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2017.

[7] M. Eyal, T. Baumel, and M. Elhadad, “Question answering as an automatic evaluation metric for news article summarization,” in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.

[8] P. Gupta, R. Tiwari, and N. Robert, “Sentiment analysis and text summarization of online reviews: A survey,” in *International Conference on Communication and Signal Processing, ICCSP 2016*, 2016.

[9] B. Dorr, D. Zajic, and R. Schwartz, “Hedge trimmer: A parse-and-trim approach to headline generation,” *Proc. HLT-NAACL 03 Text …*, 2003.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *J. Mach. Learn. Res.*, 2011.

[11] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for sentence summarization,” in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015.

[12] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016.

[13] S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT*

2016 - Proceedings of the Conference, 2016.

[14] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, 2016.

[15] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2016.

[16] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017.

[17] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[18] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep communicating agents for abstractive summarization," in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018.

[19] Y. S. Wang and H. Y. Lee, "Learning to encode text as human-readable summaries using generative adversarial networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020.

[20] L. Liu, W. Du, H. Wang, and W. Song, "Automatic summarization in Chinese product reviews," *Telkomnika (Telecommunication Computing Electronics and Control)*. 2017.

[21] N. Desai, "AUTOMATIC TEXT SUMMARIZATION USING SUPERVISED MACHINE LEARNING TECHNIQUE FOR HINDI LANGAUGE," *Int. J. Res. Eng. Technol.*, 2016.

[22] K. V. Kumar, D. Yadav, and A. Sharma, "Graph based technique for hindi text summarization," in *Advances in Intelligent Systems and Computing*, 2015.

[23] T. Vladislav and S. Denis, "Combination of abstractive and extractive approaches for summarization of long scientific texts," *arXiv*. 2020.

[24] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - Video to text," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[25] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[26] R. Khandelwal, "Attention: Sequence 2 Sequence model with Attention Mechanism," *Medium*, 2020. .

[27] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems*, 2014.

[28] Z. Tu, Z. Lu, L. Yang, X. Liu, and H. Li, “Modeling coverage for neural machine translation,” in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2016.

[29] “Ελληνική γλώσσα (Η αρχαιότερη – τελειότερη και μητέρα όλων των γλωσσών”, Αρχαίων Τόπος, 08/04/2015, [Ηλεκτρονικό]. Availlable: https://theancientwebgreece.wordpress.com/2015/04/08

[30] Γ. Μπαμπινιώτης, “ Γλωσσική σχέση τής Ελληνικής με την Αγγλική. ”, 27/06/2019, [Ηλεκτρονικό]. Availlable: https://www.babiniotis.gr/dimosieumata/glossika-themata/391-glossiki-sxesi-tis-ellinikis-me-tin-aggliki

[31] Y. Kim, “Convolutional Neural Networks for Sentence Classification”,Aug. 2014.

[32] R. Johnson and T. Zhang, “Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings,” Feb. 2016.

[33] "What is Python? Executive Summary" [Ηλεκτρονικό]. Availble: https://www.python.org/doc/essays/blurb/

[34] David Robinson, "The Incredible Growth of Python", 06/09/2017 [Ηλεκτρονικό]. Available: https://stackoverflow.blog/2017/09/06/incredible-growth-python/

[35] "ARIS Documentation" [Ηλεκτρονικό]. Available: https://doc.aris.grnet.gr/

[36] "Tensorboard: Tensorflow's visualization toolkit" [Ηλεκτρονικό]. Available: https://www.tensorflow.org/tensorboard

[37] Denis Metev,"39+ Smartphone Statistics You Should Know in 2020", 18/02/2020 [Ηλεκτρονικό]. Available: https://review42.com/resources/smartphone-statistics/

[38] "TensorFlow Lite guide", 18/02/2020 [Ηλεκτρονικό]. Available: https://www.tensorflow.org/lite/guide