Ατομική Διπλωματική Εργασία

Εφαρμογή στρατηγικής Active Learning & k-means Clustering για Fine-Tuning NMT από Κυπριακή διάλεκτο σε Νέα Ελληνική γλώσσα

Νικόλας Ζιαρτής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εφαρμογή στρατηγικής Active Learning & k-means Clustering για Fine-Tuning NMT από Κυπριακή διάλεκτο σε Νέα Ελληνική γλώσσα

Νικόλας Ζιαρτής

Επιβλέπων Καθηγητής

Prof. Κωνσταντίνος Παττίχης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2025

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Καθηγητή Κωνσταντίνο Σ. Παττίχη, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου ένα θέμα που με ενδιέφερε βαθύτατα και για την άριστη επικοινωνία που είχαμε καθ' όλη τη διάρκεια της διπλωματικής μου εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω την κ. Ρεβέκκα Παττίχη για τη συνεχή και άμεση βοήθεια που μου παρείχε και για τις γνώσεις που μοιράστηκε μαζί μου καθ' όλη τη διάρκεια της εργασίας. Η εμπειρία της, οι γνώσεις της και η αφοσίωσή της στον τομέα αυτό βοήθησαν να ξεπεραστεί κάθε πρόκληση στην έρευνα μου.

Τέλος, ευχαριστώ τον Δρ. Σπύρο Αρμοστή, ο οποίος συνέβαλε στη δημιουργία του training set, μεταφράζοντας προτάσεις από την Κυπριακή διάλεκτο στην νέα Ελληνική γλώσσα που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου, η βοήθειά του υπήρξε καθοριστική και είχαμε άριστη συνεργασία.

Περίληψη

Η αυτοματοποιήση μετάφρασης από την Κυπριακή διάλεκτο στη Νεοελληνική γλώσσα αποτελεί πρόκληση, καθώς υπάρχουν ελάχιστα διαθέσιμα παράλληλα διγλωσσικά δεδομένα για εκπαίδευση συστημάτων NMT. Για τον λόγο αυτό, η διπλωματική εργασία προτείνει ένα σύστημα ενεργητικής μάθησης εκπαιδεύοντας (fine-tuning) ένα προεκπαιδευμένου NMT μοντέλο, ώστε να μεγιστοποιεί την αποδοτικότητα της μάθησης με περιορισμένα δεδομένα. Η μεθοδολογία περιλαμβάνει ομαδοποίηση των προτάσεων με στρατηγική k-means clustering με βάση τις ενσωματώσεις τους και επιλογή των πιο πληροφοριακών προτάσεων. Στη συνέχεια, το μοντέλο προσαρμόζεται (fine-tuning) με μερική ακινητοποίηση (freezing) των πρώιμων στρωμάτων, έτσι ώστε να διατηρείται η γενική γνώση του δικτύου ενώ επικεντρώνεται στη βελτίωση της μετάφρασης της Κυπριακής διαλέκτου.

Για την αξιολόγηση του συστήματος χρησιμοποιήθηκαν πρότυπες μετρικές MT: BLEU, WER (Word Error Rate), CER (Character Error Rate) και συντελεστής συνημιτόνου (cosine similarity), υπολογιζόμενες ανά εποχή (epoch) και παρτίδα (batch). Τα πειράματα έδειξαν σημαντική βελτίωση της απόδοσης με την ενσωμάτωση της ενεργητικής μάθησης: η τιμή του BLEU αυξήθηκε αισθητά και το μοντέλο συγκλίνει γρηγορότερα σε υψηλό επίπεδο απόδοσης σε σχέση με τα μοντέλα αναφοράς. Επιπλέον, οι δοκιμές επιβεβαίωσαν τη γενικευσιμότητα της προσέγγισης, καθώς το τελικό μοντέλο υπερέβη σταθερά το baseline τόσο σε εντός τομέα όσο και σε εκτός τομέα δεδομένα. Τα κύρια συμπεράσματα δείχνουν ότι το προτεινόμενο σύστημα αυξάνει σημαντικά την αποδοτικότητα και ταχύτητα της εκπαίδευσης (μέσω ενεργητικής μάθησης), επιτυγχάνει ταχύτερη σύγκλιση της απόδοσης και διατηρεί ισχυρές δυνατότητες γενίκευσης σε διαφορετικές γλωσσικές συνθήκες.

Περιεχόμενα

Κατάλογος Σχημάτων / Πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή

Το Machine Translation έχει σημειώσει αναμφίβολα σημαντική πρόοδο τα τελευταία χρόνια, ιδιαίτερα όταν χρησιμοποιείται μεταξύ γλωσσών που διαθέτουν πολλούς πόρους όπως τα αγγλικά, τα γερμανικά και γενικά γλώσσες που χρησιμοποιούνται εκτενώς από μεγάλο αριθμό ατόμων και υπάρχουν πολλά γραπτά κείμενα. Ένα παράδειγμα Machine Translation από υψηλών πόρων γλώσσες είναι το Google Translate και το DeepL τα οποία παρουσιάζουν υψηλά ποσοστά ακρίβειας μετάφρασης. Ωστόσο, η χρήση του από ή προς γλώσσες χαμηλών πόρων, όπως η κυπριακή διάλεκτος δεν υπάρχει ως επιλογή διότι δεν αποδίδει τόσο καλά, καθώς αυτά τα μοντέλα απαιτούν σημαντικές ποσότητες παράλληλων δίγλωσσων δεδομένων για να μπορούν να γενικεύσουν σωστά και να έχουν υψηλό ποσοστό ακρίβειας στις μεταφράσεις.

Η κυπριακή διάλεκτος είναι μια μορφολογικά πλούσια διάλεκτος, η οποία είναι δύσκολο να επιτευχθεί η ακριβής της μετάφραση εξαιτίας των γραμματικών και κλιτικών της μορφών που είναι πιο σύνθετες από εκείνες της τυπικής ελληνικής

γλώσσας, καθιστώντας δυσκολότερη τη δημιουργία μεταφραστικών μοντέλων που μπορούν να γενικεύουν ορθά για διάφορα γλωσσικά μοτίβα.

Αυτή η μελέτη στοχεύει στην αντιμετώπιση των προαναφερθέντων προκλήσεων μέσω της χρήσης της μεθόδου active learning για την δημιουργία ενός machine translation συστήματος προς μετάφραση της κυπριακής διαλέκτου (χαμηλών πόρων γλώσσας) σε ελληνική γλώσσα (υψηλών πόρων γλώσσα). Η μέθοδος αυτή χρησιμοποιώντας τον αλγόριθμο k-means (χωρίζει τις προτάσεις σε clusters) και επιλέγει τις k πιο πιο σημαντικές προτάσεις που υπάρχουν στο περιορισμένο dataset της κυπριακής διαλέκτου, με απώτερο σκοπό την αύξηση της ποικιλομορφίας των προτάσεων έτσι ώστε να υπάρχει δυνατότατο βελτίωσης του μοντέλου. Οι προτάσεις αυτές μεταφράζονται manually και ακολούθως χρησιμοποιούνται για την περαιτέρω εκπαίδευση του μοντέλου για μερικές εποχές.

Με αυτή την προσέγγιση επιδιώκουμε την καλύτερη εκμετάλλευση του περιορισμένου μας dataset για πιο αποδοτική και ακριβείς μετάφραση. Επίσης εξοικονομούμε κόστος και χρόνο εξαιτίας της μειωμένης εργασίας των manual translations που θα χρειαστούν, διότι επιλέγουμε για μετάφραση λιγότερες και σημαντικότερες προτάσεις.

1.2 Το Πρόβλημα

Το πρόβλημα που προσπαθούμε να επιλύσουμε είναι η επίτευξη περισσότερης ακρίβειας μετάφρασης από την Κυπριακή διάλεκτο στην Ελληνική γλώσσα με χρήση περιορισμένων πόρων και χρόνου. Η ανάπτυξη ενός παραδοσιακού μοντέλου νευρωνικής μηχανικής μάθησης από κυπριακή διάλεκτο σε ελληνική γλώσσα δεν θα έχει τα αναμενόμενα αποτελέσματα εξαιτίας του ότι η κυπριακή διάλεκτος είναι μια γλώσσα η οποία δεν έχει αρκετά παράλληλα διαγλωσσικά δεδομένα (κείμενα προτάσεις η λέξεις μεταφρασμένες από κυπριακή σε ελληνική γλώσσα), και γι'αυτό το μοντέλο δεν θα μπορεί να κάνει ακριβή συσχέτιση των λέξεων της κυπριακής με τις αντίστοιχες λέξεις της ελληνικής. Επιπρόσθετα μπορεί να οδηγηθούμε είτε σε υπερεκπαίδευση του μοντέλου βάση των περιορισμένων μας δεδομένων είτε σε αδυναμία γενίκευσης σε περιπτώσεις που η είσοδος είναι κατά βάση άγνωστα δεδομένα (δεδομένα τα οποία το μοντέλο βλέπει για πρώτη φορά).

Έτσι η παρούσα εργασία παρουσιάζει μια διαφοροποίηση της παραδοσιακής μεθόδου μάθησης NMT χρησιμοποιώντας active learning με στρατηγική clustering των περιορισμένων δεδομένων, ουσιαστικά αυτή η στρατηγική ομαδοποιεί τις προτάσεις που έχουν αρκετά παρόμοια γλωσσικά χαρακτηριστικά και από κάθε ομάδα επιλέγει την πρόταση με τα πιο κοινά χαρακτηριστικά σε σχέση με τις προτάσεις της ομάδας της. Αυτό έχει ως στόχο την αποδοτικότερη και αποτελεσματικότερη χρήση των δεδομένων, αφού θα χρησιμοποιούνται δεδομένα που είναι πιο σημαντικά στο όλο εύρος των δεδομένων, έτσι το μοντέλο θα μπορεί να γενικοποιεί καλύτερα και σε λέξεις, φράσεις ή προτάσεις άγνωστες προς αυτό. Με τον όρο "σημαντικές προτάσεις" εννοούμε προτάσεις με ξεχωριστά στοιχεία που προσπαθούμε να εμπλουτίσουμε το μοντέλο μας ώστε να εμπεριέχει όσο το δυνατό περισσότερα μοναδικά χαρακτηριστικά έχουμε στην διάθεση μας (στο dataset).
Συμπερασματικά αυτή η μελέτη έχει ως στόχο την επίτευξη καλύτερης ποιότητας μετάφρασης από ή προς γλώσσες με περιορισμένα δεδομένα απαλείφοντας αυτό το πρόβλημα που υπάρχει.

1.3 Στόχος της Διπλωματικής

Ο στόχος μου είναι να δημιουργήσω ένα clustering-based active learning strategy για μηχανική μετάφραση μεταξύ γλωσσών με ανεπαρκή παράλληλα δίγλωσσα δεδομένα, και πιο συγκεκριμένα μετάφραση από την Κυπριακή διάλεκτο στην Ελληνική Γλώσσα. Αυτό το σύστημα έχει ως στόχο την δημιουργία ενός υψηλής ποιότητας μεταφραστή, ο οποίος θα έχει αρκετά καλά αποτελέσματα χωρίς την ανάγκη ύπαρξης ενός μεγάλου παράλληλου δίγλωσσου dataset. Το σύστημα θα μπορεί να μαθαίνει πιο αποτελεσματικά από μικρότερα και στοχευμένα subsets (από labelled προτάσεις) για κάθε training, με αποτέλεσμα την επιθυμητή μετάφραση ελαχιστοποιώντας την διαδικασία μαζικής μετάφρασης προτάσεων ειδικότερα σε γλώσσες που δεν είναι εφικτό, λόγω των περιορισμένων πόρων που υπάρχουν.

Για να πετύχω αυτό το στόχο, η μελέτη αυτή βασίζεται στους παρακάτω στόχους

1. **Διαμοιρασμός του dataset της κυπριακής διαλέκτου σε ξεχωριστές προτάσεις**. Ουσιαστικά δημιουργίας μεθόδου όπου με χρήση των κανόνων γραμματικής της κυπριακής διαλέκτου διαμοιράζει τις προτάσεις. Αυτό είναι

πολύ σημαντικό να γίνει σωστά, έτσι ώστε το clustering μετά τον προτάσεων να γίνει ορθά, λαμβάνοντας υπόψη όλα τα εννοιολογικά στοιχεία της κάθε πρότασης.

2. **Δημιουργία clustering-based μεθόδου για επιλογή προτάσεων για μετάφραση**

   Η μέθοδος αυτή θα εξαγάγει τις προτάσεις που έχουν την πιο μεγάλη εννοιολογική σημασία και θα επιφέρουν την περισσότερη βελτίωση της ακρίβειας του μοντέλου. Θα χρησιμοποιεί τον αλγόριθμο k-means clustering για να βρίσκει τις k πιο "σημαντικές" προτάσεις για να μεταφράζονται. Στόχος αυτής της μεθόδου είναι η σωστή διαχώριση των προτάσεων σε ομάδες, όπου η κάθε ομάδα θα έχει αρκετά κοινά στοιχεία

3. **Εφαρμογή μιας επαναληπτικής διαδικασίας (active learning approach).**

   Η διαδικασία θα έχει ως στόχο την βελτίωση του μοντέλου, αφού επαναληπτικά θα κάνει cluster τα unlabelled δεδομένα σε k ομάδες και θα επιλέγει τις k πιο "σημαντικές" προτάσεις, όπου ουσιαστικά αυτές οι προτάσεις θα αντιπροσωπεύουν καλύτερα την ομάδα στην οποία ανήκουν σε σχέση με όλες τις άλλες προτάσεις της ίδιας τους ομάδας. Ακολούθως οι επιλεγμένες προτάσεις θα μεταφράζονται και μετά θα γίνεται η περαιτέρω εκπαίδευση του μοντέλου βάση των καινούργιων labelled data(μεταφραζόμενες προτάσεις από κυπριακή διάλεκτο στην ελληνική τους ερμηνεία).

4. **Evaluation του μοντέλου για να εξεταστεί εάν όντος μαθαίνει και βελτιώνεται η ακρίβεια μετάφρασης σε κάθε training step**.

   Επιλέγω κάποιες προτάσεις στην αρχή της διαδικασίας προτού ξεκινήσω και τις αφαιρώ από το dataset, και αυτές τις προτάσεις της μεταφράζω στην ελληνική γλώσσα και αποτελούν το evaluation dataset. Με αυτό το dataset βλέπω κατά πόσο το μοντέλο παράγει καλύτερα αποτελέσματα ακρίβειας ανά training step, συγκρίνοντας τα αποτελέσματα του μοντέλου έχοντας σαν input τις προτάσεις κυπριακής διαλέκτου του evaluation dataset με τις αντίστοιχες ορθές μεταφράσεις.

1.4 Συνεισφορές της Διπλωματικής

Αυτή η μελέτη συνεισφέρει στο τομέα του Machine Translation για γλώσσες χαμηλών πόρων, με έμφαση στην μετάφραση από την Κυπριακή διάλεκτο στην Ελληνική γλώσσα. Οι συνεισφορές είναι οι εξής:

1. Νέα προσέγγιση με χρήση Active Learning: Παρουσιάζεται μια καινούργια μέθοδος, η οποία κάνοντας clustering των προτάσεων, επιλέγει τις πιο "σημαντικές προτάσεις" προς μετάφραση. Το clustering πραγματοποιείται μέσω του αλγορίθμου k-means όπου επιλέγονται οι προτάσεις με τα πιο ξεχωριστά γλωσσικά και εννοιολογικά χαρακτηριστικά, κατά συνέπεια μειώνεται η επιλογή προτάσεων που έχουν γλωσσικά χαρακτηριστικά πολύ παρόμοια με προτάσεις που είδη χρησιμοποιήθηκαν για εκπαίδευση και βελτίωση του μοντέλου
2. Μείωση της ποσότητας προτάσεων που χρειάζονται μετάφραση. Η μέθοδος αυτή ασχολείται με την μετάφραση μόνο των "σημαντικών" προτάσεων του dataset παρά ολόκληρου του dataset.
3. Προσπάθεια επίλυσης προβλημάτων που υπάρχουν σε χαμηλών πόρων Translation συστήματα. Η μελέτη αυτή προσπαθεί να επιλύσει αυτό το πρόβλημα προτείνοντας μια νέα διαδικασία, και να υποδείξει ότι τα translation μοντέλα μπορούν να εκπαιδευτούν με περιορισμένα δεδομένα και να παράγουν αρκετά καλά αποτελέσματα.

1.5 Περιεχόμενα Κεφαλαίων

Η μελέτη μου οργανώνεται ως εξής:

1. **Κεφάλαιο 1**: Introduction, Αυτό το κεφάλαιο περιγράφει αρχικά το πρόβλημα που υπάρχει, μετά το στόχο αυτής της μελέτης, ακολούθως αναλύει τις συνεισφορές αυτής της μελέτης στο τομέα γενικά του Machine Translation
2. **Κεφάλαιο 2**: Literature Review, Αυτό το κεφάλαιο αναφέρει προηγούμενες έρευνες και μελέτες που έχουν γίνει στο τομέα του Machine Translation και του Active learning, τις οποίες έχουν άμεση σχέση με την δημιουργία ενός χαμηλών πόρων machine translation συστήματος

3. **Κεφάλαιο 3**: Methodology, Αυτό το κεφάλαιο ουσιαστικά περιγράφει ακριβώς την διαδικασία υλοποίησης ενός τέτοιου συστήματος. Δηλαδή περιγράφονται τα στάδια, τα οποία είναι: preprocessing των δεδομένων, την στρατηγική clustering, την μέθοδο active learning και ακολούθως το evaluation method
4. **Κεφάλαιο 4**: Results, Αυτό το κεφάλαιο θα αναγράφει τα αποτελέσματα της μεθόδου που αναλύεται στην μελέτη αυτή. Η οποία βασίζεται στη σύγκριση των αποτελεσμάτων του μοντέλου σε σχέση με τις μεταφράσεις annotators υπολογίζοντας το Levenshtein distance για την μέτρηση των αλλαγών σε επίπεδο χαρακτήρων και τον υπολογισμό του Word Error Rate στο επίπεδο των λέξεων. Τέλος, το BLEU score, στο οποίο παράγονται n-grams (από unigrams έως four-grams) και υπολογίζεται η ακρίβεια για το καθένα μέσω της καταμέτρησης των επικαλυπτόμενων n-grams. Αυτές οι τιμές ακρίβειας συνδυάζονται χρησιμοποιώντας τον γεωμετρικό μέσο όρο, και εφαρμόζεται ποινή εάν η υπόθεση είναι μικρότερη από την αναφορά, οδηγώντας στο τελικό BLEU score. Έτσι μετά το τέλος της εκπαίδευσης του μοντέλου και θα κρίνεται κατά πόσο βαθμό είναι accurate το αποτέλεσμα της μετάφρασης του μοντέλου.
5. **Κεφάλαιο 5**: Discussion, σε αυτό το κεφάλαιο θα παρουσιάζει τα συμπεράσματα αυτής της μελέτης βάση αποτελεσμάτων συγκεκριμένα από την κυπριακή διάλεκτο στην ελληνική γλώσσα, καθώς και μελλοντικές χρήσης αυτής της μελέτης.

Κεφάλαιο 2

Ανασκόπηση Βιβλιογραφίας

2.1 Machine Translation Overview

Το Machine Translation (MT) είναι μια μέθοδος, η οποία αυτόματα μεταφράζει κείμενα από μια γλώσσα σε άλλη. Ο τομέας αυτός έχει εξελιχθεί πολύ τα τελευταία χρόνια εξαιτίας της ανάπτυξης της τεχνολογίας (υπολογιστικής ισχύος), της μηχανικής μάθησης και συνάμα της ύπαρξης μεγάλων παράλληλων δίγλωσσων dataset σε πιο εξελιγμένα μοντέλα, που ονομάζονται (NMT) neural machine translation μοντέλα [9]. Κάποια παραδείγματα τέτοιου είδους μοντέλων είναι τα ακόλουθα: rule based, statistical και neural methods.

Κάνοντας μια αναδρομή στην ιστορία του Machine Translation, Αρχικά τα πρώτα συστήματα που υλοποιήθηκαν ήταν βασισμένα πάνω σε γλωσσικούς κανόνες,

γραμματική και λεξικό για να δημιουργήσουν την μετάφραση. Τέτοιου είδους συστήματα ήταν τα Rule-based μοντέλα. Τα rule-based συστήματα αποδίδουν καλά μόνο με απλές γραμματικές Ακολούθως με χρήση της στατιστικής από τα μαθηματικά δημιουργήθηκαν τα probabilistic μοντέλα τα οποία χρησιμοποιούσαν στατιστικές μεθόδους για να αποφασίσουν την πιο πιθανή μετάφραση της πρότασης. Τα probabilistic μοντέλα για να έχουν τα επιθυμητά αποτέλεσμα χρειάζονται μεγάλο παράλληλο δίγλωσσο dataset. Έπειτα τα NMT (neural machine translation models), τα οποία χρησιμοποιούσαν deep learning, έτσι ώστε το μοντέλο να μπορεί να μάθει complex language patters, με αποτέλεσμα το μοντέλο να παράγει πιο ποιοτικές και fluent μεταφράσεις

Κάποια από τα προβλήματα που υπάρχουν μέχρι στιγμής στα Machine Translation Systems είναι η μη αποδοτική ανάπτυξη ενός μοντέλου μετάφρασης εάν μια ή και οι δύο γλώσσες έχουν περιορισμένα δεδομένα. Για την αντιμετώπιση αυτού του προβλήματος μπορεί να χρησιμοποιηθεί η τεχνική του active learning και της μεταφοράς της μάθησης, επιτρέποντας το σύστημα να επικεντρώνεται σε πιο ενημερωτικά δεδομένα για καλύτερη εκπαίδευση [1]. Ακόμη ένα πρόβλημα είναι η αντιμετώπιση κυριολεκτικής/μεταφορικής σημασίας των λέξεων σε ένα κείμενο, καθώς η κυριολεκτική μετάφραση συχνά αποτυγχάνει να αποδώσει το νόημα. Αυτό επιλύεται μέσω της NMT όπου τις λέξεις και φράσεις τις μετατρέπει σε vectors στο πολυδιάστατο χώρο με αποτέλεσμα αυτές οι αναπαραστάσεις να λαμβάνουν υπόψη όχι μόνο το νόημα της λέξεις αλλά και το πλαίσιο μέσα στο οποίο βρίσκεται.

2.1.1 Rule-based Machine Translation (RBMT)

Η Μηχανική Μετάφραση Βασισμένη σε Κανόνες (RBMT) είναι μια από τις πρώτες μεθόδους αυτόματης μετάφρασης. Συγκεκριμένα για την επίτευξη της μετάφρασης από μια γλώσσα σε άλλη αυτή η μέθοδος λαμβάνει υπόψη γλωσσικούς κανόνες, γραμματικές και λεξικό που ουσιαστικά είναι λέξη προς λέξη αντιστοιχία των δυο γλωσσών, που όλα αυτά κατασκευάζονται από γλωσσολόγους [2]. Η λειτουργεία της μεθόδου ξεκινά αρχικά με την ανάλυση της πρότασης(μορφολογική και συντακτική ανάλυση), ακολούθως αναγνωρίζει τα χαρακτηριστικά της πρότασης (ρήματα, ουσιαστικά και άλλα), έπειτα εφαρμόζει κανόνες για να κάνει μορφολογικές αλλαγές και τέλος δημιουργεί την πρόταση στην επιθυμητή γλώσσα. Αυτή η μέθοδος απαιτεί

δημιουργία κανόνων, πολλή χρόνο και εξειδικευμένη γνώση για την ορθή της υλοποίηση. Καταληκτικά για την μελέτη μας αυτή η μέθοδος δεν είναι εφικτή καθώς οι μειωμένοι πόροι καταστούν αδύνατη την δημιουργία επαρκών κανόνων που να καλύπτουν όλες τις περιπτώσεις.

2.1.2 Statistical Machine Translation (SMT)

Το statistical Machine translation είναι μια διαφορετική μέθοδος, η οποία βασίζεται στην στατιστική για να υπολογίσει την πιο πιθανή μετάφραση για μια δεδομένη είσοδο. Μια από τις πιο γνωστές μεθόδους της SMT είναι η phrased based translation. Η διαφορά της phrased based σε σχέση με της τυπικής είναι ότι οι προτάσεις αντί να μεταφράζονται λέξη προς λέξη, χωρίζονται σε φράσεις(μια φράση μπορεί να αποτελείται από 1 ή περισσότερες λέξεις). Ακολούθως γίνεται η αντιστοίχιση μεταξύ των φράσεων από την μια γλώσσα στην άλλη ακολουθώντας της υποδείξεις των στατιστικών πιθανοτήτων, έτσι μπορούμε να πετύχουμε μεγαλύτερη ακρίβεια μετάφρασης σε σύγκριση με την μέθοδο λέξη προς λέξη μετάφραση [4]. Τα 3 βασικά μέρη του SMT είναι το μοντέλο μετάφρασης, όπου εδώ γίνεται ο υπολογισμός της πιθανότητας αντιστοιχίας φράσεων από τη μια γλώσσα στην άλλη (τα αποτελέσματα αυτού στηρίζονται στην ανάλυση του παράλληλου dataset), Το μοντέλο γλώσσας, το οποίο εξασφαλίζει την φυσικότητα και γραμματική ορθότητα της μεταφραζόμενης φράσης και ο αλγόριθμος αποκωδικοποίησης, ο οποίος είναι υπεύθυνος για την σύνθεση της τελικής μετάφρασης

Η SMT έχει σημειώσει εξαιρετικά αποτελέσματα για γλώσσες με πολλά παράλληλα δίγλωσσα δεδομένα, Όμως δεν είναι αποτελεσματική σε γλώσσες με μειωμένους πόρους, όπως την Κυπριακή Διάλεκτο, χωρίς αρκετά δεδομένα το μοντέλο δεν μπορεί να μάθει ορθά και να αντιστοιχεί σωστά τις μεταφρασμένες φράσεις. Επιπρόσθετα η SMT έχει πρόβλημα κατανόησης του περιεχομένου της πρότασης, αφού το σύστημα βασίζεται στην μετάφραση μεμονωμένων /ανεξάρτητων φράσεων.

2.1.3 Neural Machine Translation (NMT)

Η Νευρωνική μηχανική μάθηση είναι η πιο σύγχρονη και αποτελεσματική προσέγγιση στη μηχανική μετάφραση, με τη χρήση deep learning η NMT έχει την δυνατότητα

κατανόησης ολόκληρου του περιεχόμενου της πρότασης αντί να γίνεται μετάφραση μεμονωμένων φράσεων της. Το στοιχείο αυτό της NMT συντελεί στην ενίσχυση της φυσικότητας και ακρίβειας στις μεταφράσεις. Στις αρχές τα NMT μοντέλα είχαν αρχιτεκτονικές (Sequence-to-Sequence) με RNNs σε συνδυασμό με LSTM unit για να μπορεί να διατηρεί πληροφορίες για περισσότερο χρόνο, όμως αυτή η μέθοδος παρουσίασε προβλήματα στην διατήρηση μακροχρόνιων εξαρτήσεων [3].

Εξαιτίας των προαναφερθέντων προβλημάτων δημιουργήθηκε το μοντέλο Transformer, όπου εισήγαγε νέο τρόπο κατανόησης των προτάσεων μέσω των μηχανισμών αυτό-προσοχής. Βασικά αυτή η μέθοδος αυτοπροσοχης δίνει την δυνατότητα στο μοντέλο να επεξεργάζεται ταυτόχρονα όλες τις λέξεις της πρότασης, έτσι επιτρέπετε στο μοντέλο να καταλαβαίνει το γενικό πλαίσιο της πρότασης [13].

Ένα πρόβλημα της χρήσης NMT είναι η ανάγκη ενός μεγάλου παράλληλου δίγλωσσου dataset για την εκπαίδευση του μοντέλου, σε γλώσσες όπως την κυπριακή διάλεκτο όπου τα δεδομένα είναι περιορισμένα, η αποτελεσματικότητα του μοντέλου αυτού είναι χαμηλή. Όμως ο συνδυασμός αυτού του μοντέλου με άλλα, πχ transfer learning, active learning μπορεί να έχει καλυτέρα αποτελέσματα.

2.1.4 Challenges in Machine Translation

Παρά την επιτυχία της NMT, εξακολουθούν να υφίστανται αρκετές προκλήσεις, ιδίως για γλώσσες με περιορισμένους πόρους. Οι βασικές προκλήσεις περιλαμβάνουν:

Σπανιότητα δεδομένων: Γλώσσες με χαμηλούς πόρους, όπως η κυπριακή διάλεκτος, έχουν περιορισμένα παράλληλα διαγλωσσικά δεδομένα έτσι καθίσταται δύσκολη την εκπαίδευση αποδοτικών μοντέλων NMT. Τεχνικές όπως το transfer learning και η ενεργός μάθηση στοχεύουν στην αντιμετώπιση αυτού του προβλήματος.

Λέξεις εκτός λεξιλογίου (OOV): Έχουν εισαχθεί τεχνικές τμηματοποίησης επιμέρους λέξεων, όπως η κωδικοποίηση ζεύγους byte (BPE), για τον χειρισμό άγνωστων λέξεων κατά την εκπαίδευση. Αυτές οι τεχνικές χωρίζουν τις σπάνιες λέξεις σε μονάδες υπολέξεων, επιτρέποντας στο μοντέλο να γενικεύει καλύτερα [14].

Προσαρμογή στον τομέα: Τα συστήματα ΜΤ που εκπαιδεύονται σε σώματα γενικού τομέα συχνά αποτυγχάνουν σε συγκεκριμένους τομείς (π.χ. ιατρικούς, νομικούς) λόγω των διαφορών στο λεξιλόγιο και το πλαίσιο.

Υπολογιστικό κόστος: Η εκπαίδευση μοντέλων NMT, ιδίως των Transformers, απαιτεί σημαντικούς υπολογιστικούς πόρους. Αυτό αποτελεί πρόκληση για τους ερευνητές που εργάζονται με περιορισμένο υλικό ή σε περιβάλλοντα με περιορισμένους πόρους.

2.2 Active Learning στο Neural Machine Translation

Η τεχνική του active learning χρησιμοποιείται ευρέως σε εφαρμογές μηχανικής μετάφρασης, εξαιτίας της μειωμένης του ανάγκης για μεγάλου μεγέθους παράλληλων δίγλωσσων δεδομένων. Αυτό καταστείτε εφικτό λόγου της στοχευμένης επιλογής δεδομένων που έχουν ως στόχο την μεγιστοποίηση της πληροφοριακής αξίας για την εκπαίδευση του μοντέλου. Τρείς βασικές προσεγγίσεις οι οποίες αναφέρονται συχνά στην βιβλιογραφία αυτής της μεθόδου είναι οι εξής:

2.2.1 Μέθοδοι Active Learning

Diversity Sampling:

Η στρατηγική αυτή έχει ως στόχο να εξασφαλίσει ότι το επιλεγμένο training set είναι αντιπροσωπευτικό ως προς τα χαρακτηριστικά του συνολικού dataset. Επιλέγονται δείγματα από το συνολικό dataset, τα οποία διαφέρουν μεταξύ τους και αντικατοπτρίζουν την ποικιλία του γλωσσικού περιβάλλοντος (δείγματα με διαφορετικά γλωσσικά, θεματικά και δομικά χαρακτηριστικά). Με αποτέλεσμα να δημιουργεί ένα training set, το οποίο να βοηθά στην γενίκευσει του μοντέλου σε καινούρια μη εξερευνημένες εισόδους, αποτρέποντας την υπερπροσαρμογή του μοντέλου σε ένα μικρό σύνολο από δεδομένα.

Uncertainty Sampling:

Αυτή η προσέγγιση επικεντρώνεται στην επιλογή δειγμάτων για τα οποία το μοντέλο είναι ιδιαίτερα αβέβαιο στις προβλέψεις του. Συγκεκριμένα, επιλέγει δείγματα με χαμηλές confidence scores, υποδεικνύοντας ότι το μοντέλο είτε έχει έλλειψη δεδομένων, είτε έχει δυσκολίες στην ακριβή κατηγοριοποίηση των δεδομένων. Αυτή η στρατηγική συγκεντρώνει τις προσπάθειές της στα αδύνατα σημεία του μοντέλου στο πρόβλημα κατηγοριοποίησης, όπου η επιπλέον εκπαίδευση στα σημεία αυτά θα μπορούσε να οδηγήσει σε σημαντική αύξηση της ακρίβειας του μοντέλου. Τέτοιες τεχνικές όχι μόνο μειώνουν το κόστος συλλογής επισημασμένων δεδομένων, αλλά δημιουργούν επίσης μια πιο στοχευμένη προσέγγιση μάθησης όπου η πληροφορία είναι πιο αναγκαία για τη βελτίωση του συστήματος, αφού επικεντρώνεται στις ανάγκες του μοντέλου στο κάθε training step [15].

Query-by-Committee:

Σε αυτήν την προσέγγιση, μια επιτροπή από διαφορετικά μοντέλα συνεργάζεται για να αναγνωρίσει inputs για τα οποία υπάρχει σημαντική διαφωνία μεταξύ των μοντέλων. Η ύπαρξη διαφωνίας μεταξύ των μοντέλων υποδηλώνει ότι τα δεδομένα που εξελίσσονται μπορεί να είναι ιδιαίτερα ενημερωτικά και πληροφοριακά για την παρούσα γνώση του μοντέλου, καθώς αντιπροσωπεύουν καταστάσεις όπου το μοντέλο αντιμετωπίζει δυσκολία να καταλήξει σε ομοφωνία. Οι μελέτες δείχνουν ότι η χρήση μιας τέτοιας στρατηγικής μπορεί να βελτιώσει δραματικά την απόδοση προσφέροντας περισσότερες πληροφορίες εκεί που είναι πιο απαραίτητες, έτσι ώστε να δοθεί περισσότερη έμφαση στα επόμενα training phases του μοντέλου [15].

2.2.2 Εφαρμογή σε Μεταφράσεις Χαμηλών Πόρων

Σε περιπτώσεις μετάφρασης μεταξύ γλωσσών με περιορισμένων παράλληλων δίγλωσσων δεδομένων, όπως στην περίπτωση της κυπριακής διαλέκτου, η Ενεργή Μάθηση προσφέρει δύο βασικά πλεονεκτήματα:
Εξοικονόμηση Πόρων: Αντί να μεταφράζεται ολόκληρο το σύνολο δεδομένων, επιλέγεται ένα υποσύνολο που είναι το πιο αντιπροσωπευτικό και ενημερωτικό, για παράδειγμα, 50 προτάσεις. Αυτό μειώνει σημαντικά το κόστος και τον χρόνο που σχετίζονται με τις μεταφράσεις.
Βελτίωση και διατήρηση γνώσεων σταδιακά: Σε κάθε φάση εκπαίδευσης, τα νέα δείγματα που επιλέγονται και μεταφράζονται προστίθενται στο σύνολο εκπαίδευσης ώστε το μοντέλο να μπορεί να χτίσει μια ισχυρή κατανόηση του γλωσσικού τοπίου διατηρώντας ταυτόχρονα τη γνώση από προηγούμενες φάσεις. Αυτή η προσέγγιση βοηθά το μοντέλο να βελτιώνεται συνεχώς. Έρευνες, έχουν δείξει πως οι εν λόγω στρατηγικές βελτιώνουν σημαντικά την ακρίβεια των μεταφράσεων, ενώ ταυτόχρονα μειώνουν το χρόνο και το κόστος που απαιτεί η επισήμανση δεδομένων [1].

2.3 Τεχνικές Clustering για την Επιλογή Δεδομένων

2.3.1 Εισαγωγή στο clustering:

Το clustering αποτελεί μια από τις πιο διαδεδομένες και μελετημένες τεχνικές στην εξερευνητική ανάλυση δεδομένων (EDA). Όταν χρησιμοποιούμε clustering στα δείγματα ενός συνόλου δεδομένων επιδιώκουμε να τα διαχωρίσουμε σε διακριτές ομάδες, έτσι ώστε τα δείγματα εντός κάθε ομάδας να έχουν κοινά χαρακτηριστικά και σε αντίθεση τα δείγματα που ανήκουν σε άλλες ομάδες να διαφέρουν σημαντικά.
Το clustering είναι μια μέθοδος μη επιβλεπόμενης μάθησης που στόχο έχει την εύρεση των πιο αντιπροσωπευτικών στοιχείων κάθε ομάδας, που σε συνέχεια θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου.
Οι διάφορες τεχνικές που μπορούν να χρησιμοποιηθούν όπως Centroid-based Clustering, Hierarchical Clustering οργανώνουν τα δεδομένα σε clusters σύμφωνα με συγκεκριμένα χαρακτηριστικά τα οποία τα αποτελούν. Κάποια από αυτά τα

χαρακτηριστικά μπορεί να είναι οι vector αναπαραστάσεις που παράγονται από μοντέλα word ή sentence embeddings [5].

2.3.2 Αλγόριθμος k-means

Ο αλγόριθμος k-means είναι ένας από τους πιο διαδεδομένους αλγορίθμους για clustering των δεδομένων εξαιτίας της απλότητας του και της ικανότητας του να διαχωρίζει ενα dataset σε έναν σταθερό αριθμό ομάδων, όπου σε κάθε ομάδα περιέχει ένα centroid (κεντρικό σημείο) που αντιπροσωπεύει το μέσο δείγμα όλων των στοιχείων της ομάδας του [16]. Αυτό επιτυγχάνεται με την ελαχιστοποίηση της συνολικής within cluster variance ή του Sum of Squared Errors (SSE), όπου είναι το άθροισμα των τετραγωνικών αποστάσεων των δεδομένων από τα centroits που ανήκουν στην ομάδα τους.

Η λειτουργία του αλγορίθμου διαμορφώνεται σε δυο επαναλαμβανόμενα βήματα:
1. Κάθε δεδομένο του dataset γίνεται assigned σε ένα cluster με βάση την μικρότερη απόσταση του (τη τετραγωνική Ευκλείδεια απόσταση) από τα centroids των cluster. Αυτή η διαδικασία ομαδοποιεί τα δεδομένα σε clusters με βάση την ομοιότητα τους.
2. Μόλις ολοκληρωθεί η αντιστοίχιση μεταξύ των data και των clusters, υπολογίζεται η ανανεωμένη θέση του centoid του κεντρικού σημείου του κάθε cluster ως ο μέσος όρος όλων των data που έχουν ανατεθεί στο αντίστοιχο cluster. Οι νέες αυτές τιμές ορίζουν το ανανεωμένο κέντρο του κάθε cluster [16].

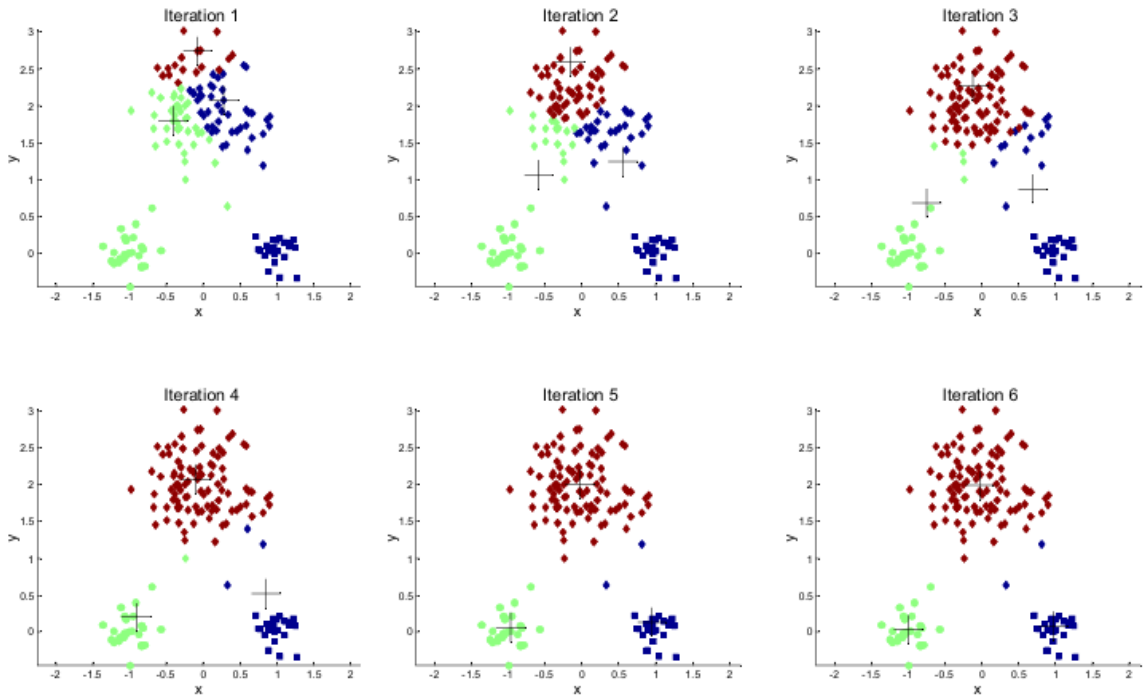
Τα πιο πάνω 2 σημεία επαναλαμβάνονται μέχρι τα centroids να σταθεροποιηθούν ή κάποιο κριτήριο που εμείς θέσαμε να επιτευχθεί, όπως να ορίσουμε ελάχιστη τιμή στο SSE ή να ορίσουμε μέγιστο αριθμό επαναλήψεων

$$\text{SSE: } J = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathbf{C}_i} d\left(\mathbf{x}_j, \bar{\mathbf{x}}_i\right)^2$$

Σχήμα 2.1 Τύπος του Sum of Squared Errors

Ο πιο πάνω τύπος εξηγεί τον υπολογισμό του SSE (το άθροισμα των τετραγωνικών αποστάσεων) όπου αθροίζονται σε όλα τα clusters η απόσταση όλων των δεδομένων τους από το centroid της ομάδας στην οποία ανήκουν [6].

Ο αλγόριθμος αυτός έχει γραμμική πολυπλοκότητα O(N·n·k), όπου N είναι τα συνολικά δεδομένα του dataset, n είναι ο αριθμός των dimentsions και k είναι ο αριθμός των clusters που θέλουμε να δημιουργήσουμε.



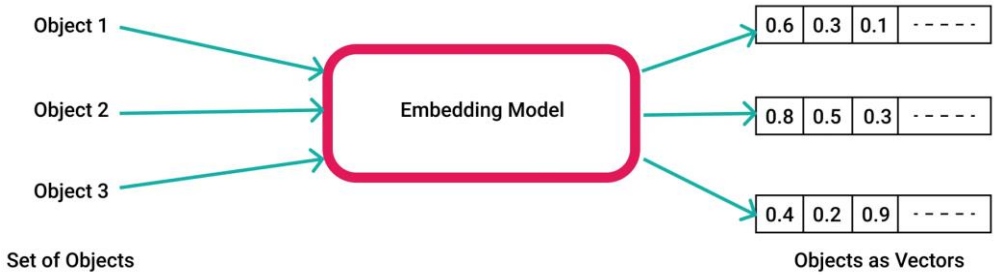
Σχήμα 2.2 K-Means clustering Εξέλιξη διαδοχικών Iterations

Παραπάνω μπορούμε να δούμε διάφορα iterations του αλγορίθμου αυτού για προκαθορισμένο k=3 ανά διαφορετικά iterations και όπως μπορούμε να δούμε την μετακίνηση των κεντρικών σημείων, με αποτέλεσμα την καλύτερη αποτίμηση της ομάδας που αντιπροσωπεύουν.

2.3.3 Εφαρμογή σε Κείμενα (προτάσεις):

Ένας τρόπος επεξεργασίας και υπολογισμού της εννοιολογικής σημασίας της φυσικής γλώσσας, ο οποίος αποδεικνύεται αξιόπιστος για χρήση είναι η μετατροπή των προτάσεων σε high-dimensional vector spaces, όπου αυτό μπορεί να γίνει με την χρήση προεκπαιδευμενων μοντέλων. Τα μοντέλα αυτά δημιουργούν embeddings (vector of

floating point numbers), τα οποία αποτυπώνουν τα πιο σημαντικά εννοιολογικά και γραμματικά στοιχεία της κάθε πρότασης.



Σχήμα 2.3 Embedding Model: Αναπαράσταση Objects ως Vectors

Έχοντας αυτά τα αριθμητικά διανύσματα της κάθε πρότασης που παρακτηκαν με χρήση προεκπαιδευμενου μοντέλου μπορείς εύκολα να συγκρίνεις την ομοιότητα (απόσταση μεταξύ διανυσμάτων) μεταξύ των προτάσεων στον πολυδιάστατο χώρο με χρήση κάποιου μετρικού όπως για παράδειγμα το cosine similarity

$$similarity(A, B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Σχήμα 2.4 Cosine Similarity: υπολογισμός ομοιότητας Vectors μέσω cos(θ)

Πιο πάνω μπορούμε να δούμε τον τρόπο σύγκρισης δυο vectors με cosine similarity, όπου A · B είναι το εσωτερικό γινόμενο μεταξύ των διανυσμάτων A και B, Το ||A|| αντιπροσωπεύει το μέτρο του διανύσματος που υπολογίζεται ως $||A|| = \sqrt{(A_1^2 + A_2^2 + ... + A_n^2)}$, θ είναι η γωνία μεταξύ των διανυσμάτων.

Ακολουθώντας την παραπάνω προσέγγιση σε συνδυασμό με αλγορίθμους clustering (όπως k-means) επιλέγονται προτάσεις, οι οποίες αντιπροσωπεύουν πλήρως το ολικό dataset. Ουσιαστικά η λειτουργία του αλγορίθμου k-means είναι η ομαδοποίηση των vectors αυτών σε ομάδες όπου το centroid της κάθε ομάδας θα είναι ο μέσος όρος των vector που την αποτελούν. Όσο πιο μικρή απόσταση βρίσκεται ένα vector από το centroid τόσο πιο αντιπροσωπευτικό, διότι συνοψίζει καλύτερα τα κύρια κοινά

χαρακτηριστικά της ομάδας στην οποία ανήκει. Ακολούθως επιλέγονται η πιο αντιπροσωπευτική πρόταση(vector) από κάθε ομάδα και προστίθεται στο training set μαζί με την αντίστοιχη της μετάφραση στα Ελληνικά, όπου αυτό εξασφαλίζει ότι το training set θα αποτελείται από ψηλής πληροφορίας δεδομένα και ότι το μοντέλο θα έχει την ικανότητα να γενικεύει

2.4 Fine Tuning στην Αυτόματη Μετάφραση

Η τεχνική του fine-tuning είναι μια διαδικασία στην οποία χρησιμοποιείς ένα προ-εκπαιδευμένο μοντέλο, το οποίο έχει ήδη γλωσσικές γνώσεις από μια ευρεία βάση δεδομένων και ουσιαστικά το εκπαιδεύεις σε ένα πιο συγκεκριμένο σύνολο δεδομένων με αποτέλεσμα να εξειδικευτεί σε ειδικές απαιτήσεις. Το τελικό fine-tune μοντέλο διατηρεί τις γενικές γνώσεις που είχε αποκτήσει αρχικά, ενώ παράλληλα βελτιώνεται στις νέες απαιτήσεις. Σύμφωνα με τις έρευνες που έγιναν για το fine-tuning στη μηχανική μετάφραση γλωσσικών προ-εκπαιδευμένων μοντέλων έδειξαν ότι μια ορθή προσέγγιση για fine-tuning γλωσσικών μοντέλων βασισμένα σε transformers είναι το freezing των αρχικών layers, τα οποία έχουν αποκτήσει ήδη γενικές γλωσσικές γνώσεις, με αυτό το τρόπο εξασφαλίζεται ότι η γενική γνώση που είχε το μοντέλο δεν αφαιρείται κατά την προσαρμογή στο νέο στόχο, και η εκπαίδευση των τελευταίων επιπέδων όπως τα τελευταία layers του encoder, ο decoder και το output(lm_head), διότι τα τελευταία αυτά επίπεδα έχουν μεγαλύτερη ικανότητα να προσαρμόζονται σε domain specific χαρακτηριστικά (πχ. Της κυπριακής διαλέκτου) [17]. Αυτό έχει ως αποτέλεσμα την βελτίωση της ακρίβειας του μοντέλου, και την εξοικονόμηση υπολογιστικών πόρων εξαιτίας των συγκεκριμένων layers τα οποία εκπαιδεύονται.

Κεφάλαιο 3

Μεθοδολογία

3.1 Επισκόπηση μεθοδολογίας

Η εργασία αυτή επικεντρώνεται στην ανάπτυξη ενός συστήματος NMT (neural machine translation) για την μετάφραση κειμένου από την κυπριακή διάλεκτο στην Κοινή Νέα Ελληνική γλώσσα. Παρόλο που η κυπριακή διάλεκτος έχει κάποια κοινά στοιχεία με την Νέα Ελληνική γλώσσα, έχουν και αρκετές διαφορές τόσο στο λεξιλόγιο, φωνολογία και σύνταξη. Στόχος του συστήματος είναι να αυτοματοποιήσει αυτή την διαδικασία μετάφρασης, έτσι ώστε κείμενα και λέξεις της κυπριακής

διαλέκτου να μπορούν να κατανοηθούν από άτομα γνώστες της νέας Ελληνικής γλώσσας

Για τους σκοπούς της εργασίας αυτής χρησιμοποιήθηκε ένα σύγχρονο μοντέλο NMT το οποίο βασίζεται σε αρχιτεκτονική (transformers). Εκπαιδεύουμε το σύστημα με απώτερο σκοπό την καλύτερη αντιστοίχιση προτάσεων Κυπριακής διαλέκτου (input) σε προτάσεις νέας Ελληνικής γλώσσας (output). Εξαιτίας των περιορισμένων παράλληλων δίγλωσσων δεδομένων (προτάσεις σε κυπριακή μεταφραζόμενες σε Ελληνική γλώσσα) η προσέγγιση που χρησιμοποιήσαμε βασίζεται στη μεταφορά μάθησης (transfer learning) και στη στρατηγική ενεργής μάθησης (active learning). Συγκεκριμένα χρησιμοποιήσαμε ένα ήδη προεκπαιδευμένο μοντέλο νευρωνικής μηχανικής μετάφρασης, το οποίο το προσαρμόσαμε (fine tuned) στο σκοπό της εργασίας αυτής με χρήση των διαθέσιμων μεταφράσεων της κυπριακής διαλέκτου. Παράλληλα, επαναληπτικά επιλέγαμε νέες προτάσεις από το corpus στο οποίο είχαμε στην κατοχή μας και τις μεταφράζαμε στη νέα ελληνική γλώσσα και αυτές τις νέες προτάσεις με τις μεταφράσεις τους τις ενσωματώναμε στο συνολικό training set, έτσι ώστε το σύστημα να βελτιώνεται σε κάθε επόμενο training phase με όσο πιο αντιπροσωπευτικά γλωσσικά στοιχεία. Στις υποενότητες 3.2 – 3.6 περιγράφεται αναλυτικά η διαδικασία που ακολουθήσαμε.

3.2 Προεπεξεργασία των δεδομένων

Εξαιτίας της περιορισμένης διαθεσιμότητας κειμένων γραμμένων σε Κυπριακή διάλεκτο, θέλαμε να τα χωρίσουμε όλα τα κείμενα που είχαμε στην κατοχή μας, τα οποία αποτελούνταν από εκθέσεις, ποιήματα, διαλόγους σε προτάσεις έτσι ώστε να τα χρησιμοποιήσουμε ως βάση για ορθή ανάλυση και κατά συνέπεια επιλογή αντιπροσωπευτικών δειγμάτων

3.2.1 Χωρισμός του dataset σε προτάσεις:

Οι κανόνες που χρησιμοποιήσαμε για τη ορθή διάσπαση του κειμένου σε προτάσεις ήταν ο χαρακτήρας τελείας ('.'), ως ο πιο κύριος οριοθέτης, ενώ ταυτόχρονα το σύμβολο της παύλας ('-'), το οποίο δηλώνει διάλογο, με αποτέλεσμα κάθε επικοινωνία μέσω διαλόγου να συμβολίζει μια ξεχωριστή πρόταση. Επιπλέον στο dataset μας είχαμε και

ποιήματα τα οποία διαχωρίζονταν με κόμματα(','), ενώναμε τους στοίχους όλου του ποιήματος, με αποτέλεσμα να ενσωματώσουμε σε μια πρόταση το συνολικό νόημα του ποιήματος

3.2.2 Καθαρισμός του dataset:

Ακολούθως μετά από το διαχωρισμό του κειμένου σε προτάσεις αφαιρέσαμε περιττά κενά ανάμεσα στις προτάσεις που μπορούσαν να προκύψουν και ακολούθως διαγράψαμε σύμβολα ή χαρακτήρες τα οποία δεν ανήκαν στο περιεχόμενο της πρότασης. Με αποτέλεσμα να έχουμε ένα ομοιόμορφο αποτέλεσμα στο οποίο κάθε γραμμή να αντιπροσωπεύει μια πρόταση ορθά σύμφωνα με τους κανόνες της ελληνικής γλώσσας.

Η ορθή διαχώριση των προτάσεων συμβάλει στην εξαγωγή high dimensional vector (embeddings), τα οποία έχουν ακριβείς πληροφορία για τα στοιχεία της κάθε πρότασης με αποτέλεσμα την καλύτερη ομαδοποίηση προτάσεων και επιλογή αντιπροσωπευτικών δειγμάτων

3.3 Επιλογή προτάσεων από το dataset μέσω Clustering algorithm

Σε αυτό το στάδιο ο στόχος μας είναι να επιλέξουμε τις πιο αντιπροσωπευτικές προτάσεις που έχουμε στο unlabeled dataset, διότι εξαιτίας χρονικών περιορισμών δεν έχουμε την δυνατότητα να μεταφράσουμε όλες τις προτάσεις. Ακολούθως να τις μεταφράσουμε σε Ελληνική γλώσσα, έτσι ώστε να έχουμε ένα σύνολο από προτάσεις της κυπριακής διαλέκτου με την αντίστοιχη τους ερμηνεία στην ελληνική γλώσσα και να ενσωματώσουμε αυτές τις νέες προτάσεις στο συνολικό training set. Η επαναληπτική διαδικασία αυτή βασίζεται στις εξής 2 μεθοδολογίες:

3.3.1 Εξαγωγή high dimensional vectors :

Εδώ ουσιαστικά για κάθε πρόταση κυπριακής διαλέκτου που ανήκει στο dataset μας και δεν έχει ήδη χρησιμοποιηθεί σε κάποιο στάδιο εκπαίδευσης του μοντέλου μετατρέπεται σε πολυδιάστατο vector χρησιμοποιώντας το προεκπαιδευμένο μοντέλο "ilsp/Meltemi-7B-Instruct-v1.5". Το Meltemi 7B είναι ένα LLM ειδικά προσαρμοσμένο στην

ελληνική γλώσσα, το οποίο μπορεί να παράγει σημασιολογικά πλούσιες αναπαραστάσεις για δοσμένο κείμενο [7]. Αυτό έχει ως αποτέλεσμα κάθε vector να αναπαριστά τα πιο βασικά εννοιολογικά και γραμματικά χαρακτηριστικά της πρότασης της οποία αντιπροσωπεύει. Με αυτήν την μετατροπή επιτρέπεται η σύγκριση ομοιότητας μεταξύ προτάσεων και κατά συνέπεια η ομαδοποίηση τους.

3.3.2 Εφαρμογή του αλγορίθμου k-means :

Τώρα που έχουμε τα embeddings, μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο k-means για να μας τα ομαδοποίηση σε ομάδες με κοινά στοιχεία. Ο αριθμός ομάδων που θα ομαδοποιηθούν οι προτάσεις αυτές είναι προκαθορισμένος και των ορίζουμε εμείς, για τους σκοπούς αυτής τις μελέτης δημιουργούσαμε 50 ομάδες προτάσεων κάθε φορά. Κάθε ομάδα έχει ένα centroid, αυτό το vector αναπαριστά τα average στοιχεία της κάθε πρότασης στην ίδια ομάδα. Επιλέγουμε μια πρόταση από κάθε ομάδα αφού τρέξουμε τον αλγόριθμο αυτό η οποία απέχει την μικρότερη ευκλείδεια απόσταση από το centroid συγκριτικά με τις άλλες προτάσεις στην ίδια ομάδα, αποφεύγοντας παράλληλα την πλεονάζουσα πληροφορία

3.3.3 Δημιουργία του επόμενου training batch:

Τώρα που έχουμε τις 50 επόμενες εννοιολογικά και γραμματικά πιο σημαντικές προτάσεις τις μεταφράζουμε και επεκτείνουμε το training dataset έχοντας αυτές και τις αντίστοιχες τους μεταφράσεις στην Ελληνική γλώσσα και εκπαιδεύουμε το μοντέλο. Το batch 1 αποτελείται από 50 προτάσεις και ακολούθως κάθε επόμενο περιέχει 50 νέες προτάσεις από το αμέσως προηγούμενο. Τα πρώτα 4 batches τα μετάφρασα εγώ και ακολούθως τα επόμενα 5,6,7,8,9 με την βοήθεια annotators. Ακολουθώντας αυτή την διαδικασία ενισχύουμε το γνωστικό υπόβαθρο του μοντέλου σε περιπτώσεις που το μοντέλο είναι αβέβαιο, αφού επιλέγουμε προτάσεις που αντανακλούν επαρκώς την ποικιλία του αρχικού συνόλου δεδομένων και παράλληλα ενισχύεται η ικανότητα του μοντέλου για γενίκευση και ακρίβεια μετάφρασης , μέσω επαναληπτικών φάσεων εκπαίδευσης.

3.4 Προεκπαιδευμένο Μοντέλο Μετάφρασης και Fine-Tuning

Για την υλοποίηση αυτού του αυτόματου μεταφραστή επιλέχθηκε το μοντέλο **"Helsinki-NLP/opus-mt-en-grk",** το οποίο δημιουργήθηκε από το πανεπιστήμιο Ελσίνκι, και πιο συγκεκριμένα βασίζεται σε Marian NMT (Transformer) και έχει εκπαιδευτή σε παράλληλα δίγλωσσα δεδομένα από ζεύγη αγγλικών σε ελληνικών προτάσεων [8]. Το προεκπαιδευμένο αυτό μοντέλο που επιλέξαμε έχει πλούσια γνώση της ελληνικής γλώσσας στην έξοδο(output) κάτι που συνάδει με τους στόχους μας. Παρόλο που το input του μοντέλου αυτού είναι προτάσεις της Αγγλικής γλώσσας (τις οποίες μπορεί να μεταφράσει με ψηλή ακρίβεια σε νέα ελληνική γλώσσα), η χρήση του ως βάση για προσαρμογή κρίθηκε σημαντική, καθώς προσφέρει μια προ εκπαίδευση των γλωσσικών χαρακτηριστικών της ελληνικής γλώσσας. Έτσι χρησιμοποιήσαμε το μοντέλο αυτό ώστε να το προσαρμόσουμε σε μετάφραση από Κυπριακή σε Ελληνική γλώσσα.

Το fine-tuning του μοντέλου αυτού, πραγματοποιήθηκε με την χρήση της βιβλιοθήκης **HuggingFace Transformers**, η οποία παρείχε εργαλεία για αποδοτική εκπαίδευση τέτοιου είδους μοντέλων. Αρχικά φορτώσαμε το μοντέλο αυτό, με τις παραμέτρους τις οποίες είχε εκπαιδευτεί και ακολούθως το εκπαιδεύσαμε σε training phases με επιλεγμένα ζεύγη προτάσεων κυπριακής-ελληνικής. Οι κύριες παράμετροι που εφαρμόστηκαν στην εκπαίδευση του μοντέλου είναι οι εξής:

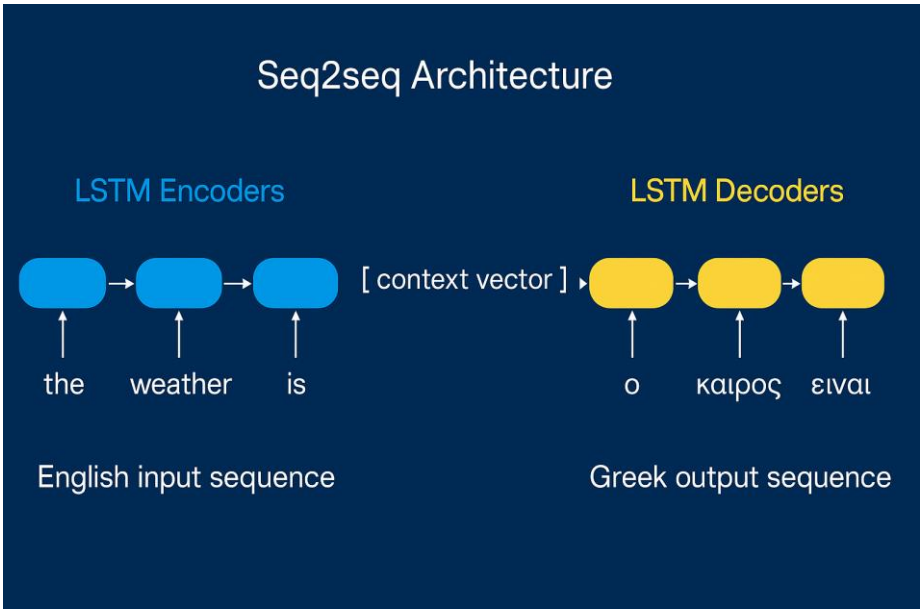
```
# training arguments
def get_training_args(output_dir):
    return TrainingArguments(
        output_dir=output_dir,
        num_train_epochs=50,
        per_device_train_batch_size=4,
        gradient_accumulation_steps=8,
        max_steps=50,
        eval_steps=50,
        save_total_limit=1,
        learning_rate=1e-5,
        fp16=True,
        logging_dir=output_dir + "/logs",
        logging_steps=50,
        report_to="none",
    )
```

Σχήμα 3.1 TrainingArguments: παράμετροι εκπαίδευσης μοντέλου

Οι παραπάνω παράμετροι εξασφαλίζουν ότι το fine-tuning γίνεται με μικρό learning rate, μικρά batch αποφεύγοντας μεγάλες απότομες αλλαγές, οι οποίες μπορούν να προκαλέσουν catastrophic forgetting της προυπάρχουσας γνώσης του μοντέλου που χρησιμοποιούμε. Επιπρόσθετα η εκπαίδευση του μοντέλου γίνεται τμηματικά ενσωματώνοντας 50 νέα ζεύγη προτάσεων στο training set κάθε training phase του fine tuning(βλ. ενότητα 3.4). Η διαδικασία αυτή του active learning – fine tuning επιτρέπει την βελτίωση του μοντέλου και παράλληλα της διατήρησης της γνώσης που απόκτησε σε προηγούμενα training phase.

3.5 Freezing των layers του μοντέλου κατά το fine-tuning

Το μοντέλο Helsinki-NLP/Opus-MT (μετάφραση από αγγλικά σε ελληνικά) αποτελεί μια σύγχρονη μορφή νευρωνικού δικτύου με Seq2Seq αρχιτεκτονική. Πιο συγκεκριμένα το MarianMT μοντέλο έχει 2 κύρια μέρη: Έναν encoder (κωδικοποιητή) και ένα decoder (αποκωδικοποιητή) όπως μπορούμε να δούμε και στην εικόνα πιο κάτω



Σχήμα 3.2 Seq2Seq Architecture: Encoders & Decoders για Translation

Ο encoder έχει multiple layers of self-attention and feed-forward neural networks, τα οποία μετασχηματίζουν σταδιακά την πρόταση εισόδου σε μια ενδιάμεση αναπαράσταση. Ακολούθως ο decoder με παρόμοια δομή, παίρνει το αποτέλεσμα του encoder και μέσω μηχανισμών cross-attention (attention προς την έξοδο του encoder)

και feed-forward neural networks συνθέτει την μετάφραση στη γλώσσα στόχο. Τέλος το lm-head (language model head) που είναι το τελευταίο επίπεδο εξόδου, το οποίο παίρνει τα αποτελέσματα από τον decoder και για κάθε λέξη υπολογίζει την πιθανότητα να είναι η επόμενη στην πρόταση, δηλαδή αποφασίζει ποια λέξη είναι πιο πιθανό να ακολουθεί.

Κατά το fine-tuning παγώσαμε κάποια layers του μοντέλου αυτού με στόχο να προσαρμόσουμε το μοντέλο στα νέα δεδομένα και παράλληλα να εκμεταλλευτούμε όσο το δυνατό περισσότερο τις γνώσεις που προ είχε το μοντέλο αυτό.

```
# Step 2: Freeze all layers except the last encoder layers, decoder, and lm_head
for param in model.parameters():
    param.requires_grad = False

# Unfreeze the last 3 encoder layers
for param in model.model.encoder.layers[-3:].parameters():
    param.requires_grad = True

# Unfreeze the decoder
for param in model.model.decoder.parameters():
    param.requires_grad = True

# Unfreeze the output projection layer
for param in model.lm_head.parameters():
    param.requires_grad = True
```

Σχήμα 3.3 κώδικας για Layer Freezing & Unfreezing

Αρχικά κάναμε freeze όλα τα layers έτσι ώστε να δηλώσουμε στο μοντέλο ακριβώς ποια από τα layers του decoder, encoder και αν το lm.head θα χρησιμοποιηθούν στην εκπαίδευση

3.5.1 Αποδέσμευση των τελευταίων τριών layers του Encoder:

Ακολούθως, κάναμε (unfreezing) σε κάποια από τα ανώτερα στρώματα του encoder, με αποτέλεσμα να εκπαιδευτούν. Συγκεκριμένα τα τρία τελευταία layers του encoder έγιναν unfreezed, καθιστώντας τα βάρη τους προσαρμόσιμα για περαιτέρω εκπαίδευση, ενώ τα αρχικά layers του encoder παρέμειναν freezed. Με αυτόν τον τρόπο, το μοντέλο άρχισε να μαθαίνει να τροποποιεί τις υψηλού επιπέδου αναπαραστάσεις της εισόδου, προκειμένου να αντανακλούν όσο το δυνατό καλύτερα

στις ιδιαιτερότητες της κυπριακής διαλέκτου, διατηρώντας παράλληλα τις πιο βασικές γλωσσικές δομές από τα αρχικά layers, πιο γενικά μοτίβα.

3.5.2 Αποδέσμευση ολόκληρου του Decoder:

Αφού σταθεροποιήθηκε η εκπαίδευση με τον μερικώς ελεύθερο encoder, αποδεσμεύθηκε πλήρως και ο decoder. Όλα τα layers του decoder έγιναν πλέον εκπαιδεύσημα, επιτρέποντας στο μοντέλο να προσαρμόσει τον τρόπο που συνθέτει την έξοδο (μεταφρασμένη πρόταση στην Ελληνική γλώσσα). Το στάδιο αυτό, ήταν κρίσιμο για να μάθει το μοντέλο να αποδίδει τις συντακτικές και λεξιλογικές ιδιαιτερότητες της ελληνικής γλώσσας(γλώσσας στόχου), πέρα από τις γενικές ικανότητες μετάφρασης που ήδη διέθετε.

3.5.3 Αποδέσμευση του επιπέδου εξόδου (lm_head):

Τέλος, κάναμε unfreeze και το τελικό επίπεδο εξόδου του μοντέλου, το lm_head. Με αυτό το τρόπο επιτρέπουμε στα βάρη του να εκτεθούν σε εκπαίδευση κατά το fine-tuning, έτσι ώστε το μοντέλο να μαθαίνει απευθείας ποιες λέξεις είναι πιο σημαντικές για το εξειδικευμένο μας domain και να βελτιστοποιεί τις προβλέψεις του, προσφέροντας ακριβέστερες και πιο σχετικές λέξεις

Με την μερική αποδέσμευση του encoder και στη συνέχεια του decoder, δίνεται στο μοντέλο η δυνατότητα να μαθαίνει νέες πληροφορίες (π.χ. διάφορα γλωσσικά χαρακτηριστικά του νέου dataset) με ελεγχόμενο τρόπο, πρώτα προσαρμόζοντας τα υψηλότερα επίπεδα κατανόησης της εισόδου και έπειτα τον ίδιο τον σχηματισμό της εξόδου. Αυτό περιορίζει τον κίνδυνο του φαινομένου catastrophic forgetting, όπου η μάθηση νέου περιεχομένου θα μπορούσε να διαγράψει τη γνώση της γενικής μετάφρασης που ήδη έχει αποκτηθεί.

Συνολικά, η στρατηγική αυτή εξισορροπεί τη διατήρηση της γενικής μεταφραστικής ικανότητας του προτύπου και την εστιασμένη προσαρμογή του στα ειδικά χαρακτηριστικά του νέου σετ εκπαίδευσης, οδηγώντας σε ένα πιο αποδοτικό και σταθερό fine-tuning.

3.6 Αξιολόγηση μοντέλου

Για την καλύτερη αξιολόγηση της ποιότητας των παραγόμενων προτάσεων χρησιμοποιήσαμε την τεχνική cosine similarity, η οποία βασίζεται στην αναγνώριση της σημασιολογικής ομοιότητας των προτάσεων και όχι κατά ανάγκη στην ακριβή αντιστοιχία λέξεων. Μετά από κάθε φάση εκπαίδευσης το fine-tuned μοντέλο περνούσε από αξιολόγηση με τις διαθέσιμες προτάσεις τόσο του trianing set όσο και του validation set. Συγκεκριμένα παραγόταν η αντίστοιχη μετάφραση των κυπριακών προτάσεων από το μοντέλο σε κάθε φάση ανά εποχή και συγκρινόταν με τις ορθές μεταφράσεις ως reference.

Για την επίτευξη της σύγκρισης αυτής, υπολογίστηκαν τα embedding vectors (τα οποία αντικατοπτρίζουν τις προτάσεις στον πολυδιάστατο χώρο) τόσο για τις predicted προτάσεις του μοντέλου όσο και για τις προτάσεις reference. Αυτό έγινε με την βοήθεια ενός προ υπάρχον πολύγλωσσικού μοντέλου μετατροπής προτάσεων σε διανύσματα (βλ ενότητα 3.7). Έπειτα η ομοιότητα των διανυσμάτων αυτών συγκρίθηκε με δυο μετρικές.

- Cosine similarity: Υπολογίστηκε το συνημίτονο της γωνιάς μεταξύ των δυο διανυσμάτων, το οποίο παίρνει τιμή 1 όταν οι δυο προτάσεις έχουν ταυτόσημες ενσωματώσεις (είναι ευθυγραμμισμένα τα διανύσματα) και παίρνει τιμή 0 όταν δεν έχουν κάποια σχέση μεταξύ τους (ορθογώνια διανύσματα)
- Dot product: Το εσωτερικό γινόμενο, το μη κανονικοποιημένο γινόμενο των δυο embeddings, το οποίο λάμβανε υπόψη το μέγεθος (norm) των διανυσμάτων

Οι παραπάνω μετρήσεις επιτρέπουν να εκτιμηθεί πόσο κοντά βρίσκεται το νόημα της μετάφρασης του fine-tuned μοντέλου την δεδομένη στιγμή σε σχέση με το νόημα της ορθής μετάφρασης της κυπριακής πρότασης. Ψηλή τιμή του cosine similarity, μας δηλώνει ότι η πρόταση που πάραξε το μοντέλο χωρίς κατά ανάγκη να είναι ακριβώς ίδια με την πρόταση reference, μεταφέρει σωστό νόημα μετάφρασης. Με αποτέλεσμα αυτές οι μετρικές να αντικατοπτρίζουν καλύτερα την ποιότητα του μοντέλου.

3.6.1 Αξιολόγηση ανά εποχή εκπαίδευσης στα Batch 1-9

Για να υπολογίσουμε τα embeddings των προτάσεων χρησιμοποιήσαμε το προεκπαιδευμένο μοντέλο "paraphrase-multilingual-mpnet-base-v2", το οποίο ανήκει στην οικογένεια των sentence Transformers. Αυτό το πολύγλωσσο μοντέλο κατασκευάστηκε για την μετάφραση προτάσεων σε διανύσματα [10]. Το μοντέλο αυτό έχει εκπαιδευτεί σε περισσότερες από 50 γλώσσες , συμπεριλαμβανομένων και της ελληνικής. Επιπρόσθετα με την χρήση του μοντέλου αυτού κάθε πρόταση αναπαρίσταται με την ίδια χωρική αναπαράσταση (768 dimensional space) ανεξαρτήτως γλώσσας, με αποτέλεσμα να μπορούν να εφαρμοστούν άμεσα τεχνικές εύρεσης ομοιότητας μεταξύ προτάσεων(π.χ cosine similarity).

Κατά την εκπαίδευση του μοντέλου (fine-tuning του **Helsinki-NLP/opus-mt-en-grk**), οι τιμές των μετρικών αξιολόγησης που χρησιμοποιήθηκαν παρακολουθούνταν ανά εποχή (epoch) τόσο για το training set όσο και για το validation set. Συγκεκριμένα μετά το τέλος της εκπαίδευσης κάθε εποχής, χρησιμοποιούσαμε τα checkpoints του μοντέλου στη παρούσα φάση για να μεταφράσουμε όλες τις προτάσεις του training set (εκείνης της επανάληψης), καθώς και τις προτάσεις του validation set (για σκοπούς επικύρωσης) . Ακολούθως υπολογιζόταν η μέση τιμή του cosine similarity και dot product μεταξύ των μεταφράσεων του μοντέλου και των αντίστοιχων ορθών μεταφράσεων. Το validation set αποτελείται από ένα ζεύγος 50 προτάσεων κυπριακής διαλέκτου και τις αντίστοιχες τους μεταφράσεις στην Ελληνική γλώσσα και οι προτάσεις αυτές επιλέχτηκαν στην αρχή από το ολικό dataset και δεν συμπεριλαμβάνονται στην εκπαίδευση, έτσι ώστε οι μετρικές ως προς το validation set να αντανακλούν την ικανότητα γενίκευσης του μοντέλου σε νέες προτάσεις, που το μοντέλο δεν γνωρίζει.

Αυτή η διαδικασία ανάλυσης ανά εποχή παρείχε χρήσιμη πληροφορία για την πορεία μάθησης, αφού παρατηρώντας την μπορούσες να βγάλεις συμπεράσματα για την ποιότητα υλοποιήσεις του μοντέλου. Συγκεκριμένα η ανοδική τιμή του cosine similarity στο training set επιβεβαιώνει ότι το μοντέλο μαθαίνει αποτελεσματικά το ζεύγος προτάσεων που του δίνουμε (training set). Επιπρόσθετα η δυνατότητα σύγκρισης με την τιμή του cosine similarity για το validation set ανά εποχή μας επιτρέπει να

ανιχνεύσουμε φαινόμενα overfitting (η ομοιότητα των προτάσεων που παράγει το μοντέλο σε σχέση με τις ορθές μεταφράσεις στο training set έχουν ψηλότερη cosine similarity τιμή ενώ αντιθέτως η ομοιότητα των προτάσεων που παράγει το μοντέλο σε σχέση με τις ορθές προτάσεις στο validation set παρουσιάζει μια ύφεση ή στασιμότητα), με αποτέλεσμα το μοντέλο να προσαρμόζεται υπερβολικά στα παραδείγματα εκπαίδευσης.

Αυτές οι μετρικές βοήθησαν στην επιλογή των εποχών που επιλέξαμε να εκπαιδεύουμε το μοντέλο σε κάθε training phase (κάθε training phase εκπαιδεύαμε το μοντέλο για 1200 εποχές) για να αποφευχθεί η υπερπροσαρμογή. Επιπλέον, κατέδειξαν την συνολική βελτίωση στην ποιότητα μεταφράσεων του μοντέλου καθώς προχωρούσαν οι επαναληπτικές προσθήκες δεδομένων: σε κάθε νέο batch, οι αρχικές τιμές ομοιότητας (πριν το fine-tuning της εκάστοτε επανάληψης) ήταν υψηλότερες από τις αντίστοιχες αρχικές της προηγούμενης, υποδηλώνοντας ότι το μοντέλο διατηρούσε και αξιοποιούσε την γνώση από τις προηγούμενες φάσεις εκπαίδευσης.

3.6.2 Αξιολογηση ανα εποχη εκπαιδευσης στα Batch 5-9

Από το batch 5 μέχρι το batch 9, κάθε batch συνοδευόταν από μεταφράσεις προτάσεων που παρασχέθηκαν από μεταφραστές, (*annotators*) και αποφασίσαμε να κάνουμε κάποιους ακόμη ελέγχους αξιολόγησης με διαφορετικές τεχνικές για να συγκρίνουμε και παράλληλα να επιτρέπουν την ενδιάμεση αξιολόγηση του μοντέλου πριν την ένταξη των νέων δεδομένων στην εκπαίδευση. Εχτός από τις πιο πάνω αξιολογήσεις που κάναμε στο μοντέλο μας (βλ. 3.6.1) για όλα τα batches ακολουθήσαμε τα βήματα που αναφέρονται πιο κάτω:

Συγκεκριμένα πριν από το training του μοντέλου με το batch N, το ήδη εκπαιδευμένο μοντέλο μέχρι και το προηγούμενο batch N-1 αξιολογείται χρησιμοποιώντας τις μεταφραζόμενες προτάσεις του batch N. Η διαδικασία για κάθε batch N (για N=5,6,7,8) έχει ως εξής:

1. **Μεταφράσεις με το τρέχον μοντέλο**: Πήραμε τα checkpoints του μοντέλου, όπως έχει διαμορφωθεί έως το batch N-1, και δώσαμε ως input την κάθε

πρόταση Κυπριακής διαλέκτου που βρίσκεται στο batch N, και το μοντέλο ακολούθως παράγει την αντίστοιχη μετάφραση στα Νέα Ελληνικά.

2. Σύγκριση με μεταφράσεις από annotators: Οι μεταφράσεις που πάραξε το μοντέλο για το batch N συγκρίνονται με τις αντίστοιχες ορθές μεταφράσεις από τους anottators. Η σύγκριση αυτή πραγματοποιείται με τον υπολογισμό τριών βασικών μετρικών αξιολόγησης:

- **Η απόσταση Levenshtein(σε επίπεδο χαρακτήρων):** Αυτή η απόσταση μετρά τον ελάχιστο αριθμό στοιχειωδών εισαγωγών, διαγραφών ή αντικαταστάσεων που απαιτούνται ώστε η παραγόμενη μετάφραση του μοντέλου να ταυτιστεί με την αντίστοιχη μετάφραση των μεταφραστών
- **Ποσοστό Σφάλματος λέξεων(Word Error Rate, WER):** το ποσοστό αυτό είναι ο λόγος των ελάχιστων απαραίτητων αλλαγών σε επίπεδο λέξης (αντικαταστάσεις, διάγραφες ή εισαγωγές λέξεων) που χρειάζονται για να ταυτιστεί με την αντίστοιχη μετάφραση των μεταφραστών, ως προς το συνολικό πλήθος λέξεων της ορθής πρότασης (αναφοράς). Αυτό εκφράζεται σε ποσοστό. Το WER μας δίνει το ποσοστό των λέξεων που χρειάζονται να διορθωθούν
- **BLEU:** Το BLEU score (Bilingual Evaluation Understudy) αξιολογεί την ποιότητα μετάφρασης του μοντέλου σε σχέση με την ορθές μεταφράσεις των annotators μέσω της σύγκρισης μικρών αλληλουχιών λέξεων(n-grams). Υπολογίζεται η ακρίβεια αντιστοίχισης για n=1 μέχρι και n=4 και ακολούθως υπολογίζεται ο γεωμετρικός μέσος των επιμέρους αυτών τιμών και παράλληλα εφαρμόζεται και ένας συντελεστής ποινής μήκους ώστε να ληφθεί υπόψη σημαντική απόκλιση στο μήκος της πρότασης

3. Υπολογισμός μετρικών: Ο υπολογισμός των πιο πάνω τεχνικών αξιολόγησης του μοντέλου πραγματοποιήθηκαν με την χρήση της βιβλιοθήκης Evaluate του Hugging Face (απόσταση Levenshtein, το WER και το BLUE)

4. Ενημέρωση του μοντέλου με το νέο batch: Αφού αξιολογήσουμε το μοντέλο με τα checkpoints μέχρι και το batch N-1 (ακολουθώντας τα βήματα

που αναφέρονται πιο πάνω). Εκπαιδεύουμε το μοντέλο εκ νέου με την χρήση του batch N (διευρυμένο σύνολο προτάσεων που περιέχει τα 50 νέα μεταφρασμένα ζεύγη προτάσεων, τα οποία έχουν ελεγχτεί από μεταφραστές). Η παραπάνω διαδικασία εφαρμόστηκε διαδοχικά για τα batch 5, 6, 7 και 8.

Κεφάλαιο 4

Αποτελέσματα

4.1 Ανάλυση αποτελεσμάτων

Κατά την εκπαίδευση του μοντέλου πήραμε μετρήσεις για την βελτίωση των αποτελεσμάτων μετάφρασης του μοντέλου ανά training phase, και τις συνοψίσαμε στις παρακάτω γραφικές παραστάσεις και πίνακες. Τα αποτελέσματα απεικονίζονται στις **(α)**γραφικές (cosine similarity) μεταξύ της πρόβλεψης του μοντέλου και της πραγματικής μετάφρασης, για το training set αλλά και για το validation set (*το validation set που χρησιμοποιήθηκε είναι ένα σύνολο προτάσεων κυπριακής με τις αντίστοιχες τους μεταφράσεις που δεν συμπεριλήφθηκε στην εκπαίδευση του μοντέλου και χρησιμοποιήθηκε ως μέσο αξιολόγησης άγνωστων προτάσεων για σκοπούς επικύρωσης*) ανά batch από το 1ο έως το 9ο. Επίσης παρουσιάζεται **(β)** πίνακας με μετρήσεις αξιολόγησης των BLUE scores, WER, CER για τα batch 5 έως 9 συγκρίνοντας τις προβλέψεις του μοντέλου με τις μεταφράσεις που πάρθηκαν από μεταφραστές. Ακολούθως γίνεται μια αποτίμηση των μετρήσεων απόδοσης του μοντέλου

4.2 Μετρήσεις Semantic Similarity ανά batch

Στις γραφικές παραστάσεις πιο κάτω 5.2.1 - 5.2.9 παρουσιάζεται η εξέλιξη του semantic similarity του μοντέλου ανά batch εκπαίδευσης. Όπως μπορούμε να

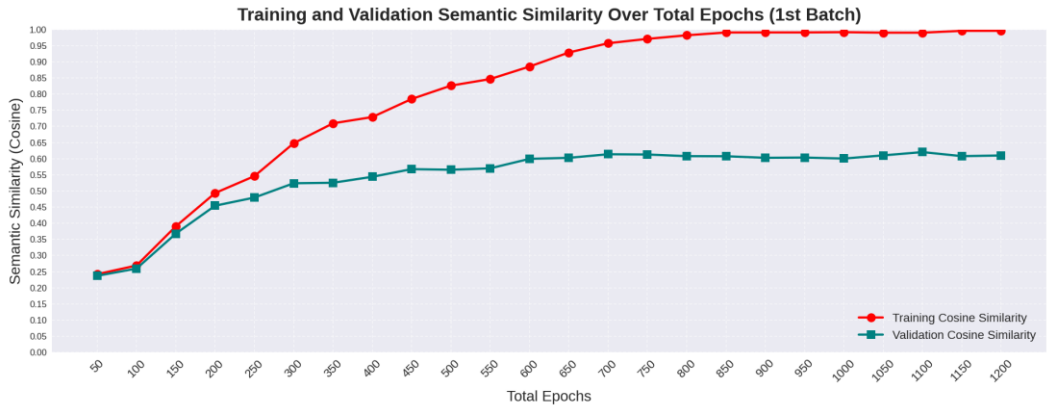
παρατηρήσουμε η ομοιότητα των αποτελεσμάτων του μοντέλου ως προς το σύνολο εκπαίδευσης (training set) σε κάθε φάση έχει μια αυξητική πορεία και προσεγγίζει τιμή κοντά στο 1, το οποίο υποδηλώνει ότι το μοντέλο μαθαίνει την νέα πληροφορία (νέο batch από προτάσεις) και προσαρμόζεται έτσι ώστε η πρόβλεψη του να ταυτίζεται με τις ορθές μεταφράσεις των προτάσεων της Κυπριακής διαλέκτου. Η αντίστοιχη ομοιότητα του συνόλου επικύρωσης επίσης βελτιώνεται κατά την εκπαίδευση, αλλά τείνει να σταθεροποιείται σε ελαφρώς χαμηλότερη τιμή σε σύγκριση με του training set.

Στα αρχικά στάδια εκπαίδευσης (1ο – 3ο batch), παρατηρείται ότι η καμπύλη του training set αυξάνεται πολύ πιο απότομα σε σχέση με τις υπόλοιπες φάσεις εκπαίδευσης εξαιτίας του γεγονός της πρώτης επαφής του μοντέλου Helsinki με πολύ διαφορετικά γλωσσικά στοιχεία σε σχέση με αυτά που ήταν εκπαιδευμένο. Επιπρόσθετα αυτό οδηγεί σε ένα προβάδισμα της ομοιότητας του training set έναντι του validation set και αντανακλά το αναμενόμενο φαινόμενο ότι το μοντέλο προσαρμόζεται πιο εύκολα στα δεδομένα πάνω στα οποία έχει εκπαιδευτεί άμεσα.

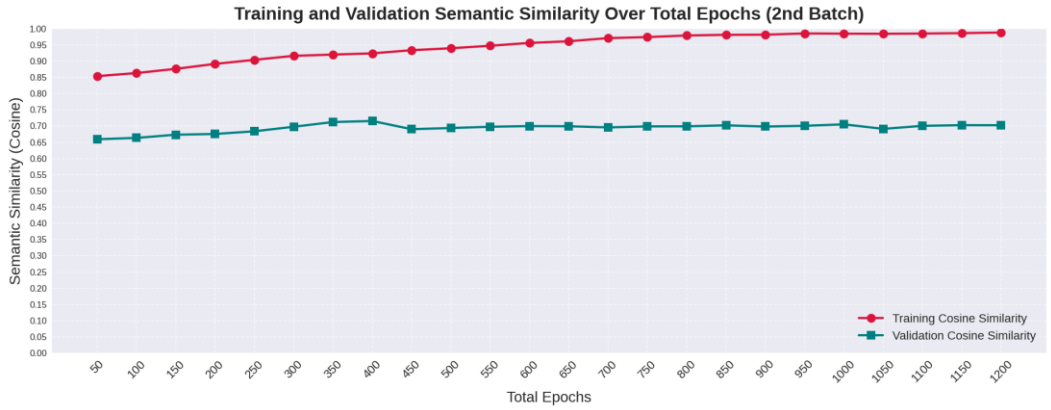
Καθώς προχωρά η εκπαίδευση στα επόμενα batch (4ο batch και έπειτα), το μοντέλο εκτίθεται σε πιο ποίκιλα δεδομένα που αντικατοπτρίζουν διαφορετικά χαρακτηριστικά των γλωσσών και γι'αυτό παρατηρείται ότι οι δυο καμπύλες έχουν μια πιο οξεία αυξητική παράλληλη κατεύθυνση, διατηρώντας ένα μικρό κενό μεταξύ τους. Επίσης μπορούμε να παρατηρήσουμε ότι το semantic similarity του σετ επικύρωσης αυξάνετε σε κάθε training phase, έτσι καταλήγουμε στο συμπέρασμα ότι η στρατηγική που επιλέξαμε για εύρεση προτάσεων για ένταξη στο training set έχει θετική επιρροή στην βελτίωση των αδυναμιών του μοντέλου ανά εκπαίδευση

Τέλος στα τελευταία batch 8ο και 9ο βλέπουμε μια σταθεροποίηση της μάθησης, το μοντέλο έχει προσεγγίσει πλέον ένα επίπεδο απόδοσης όπου οι βελτιώσεις ακολουθώντας την διαδικασία του active learning είναι μηδαμινές σύμφωνα με το corpus προτάσεων που είχαμε στην κατοχή μας.

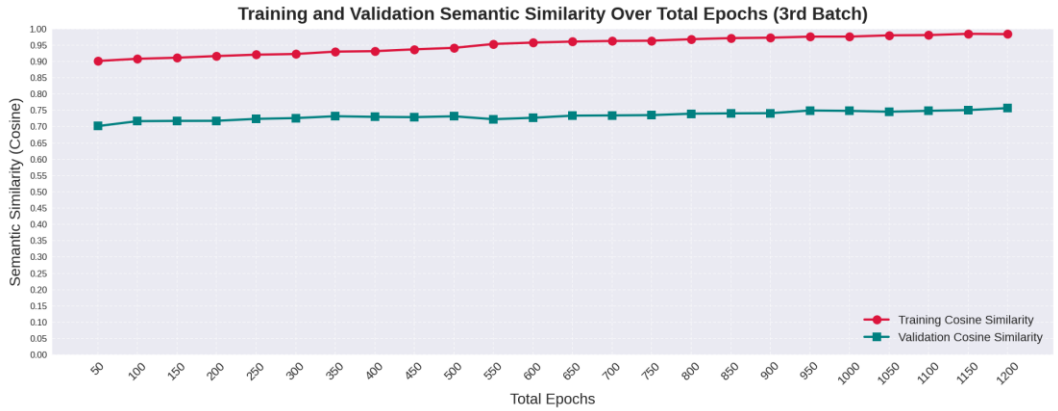
4.2.1 Γραφικές παραστάσεις Semantic Similarity



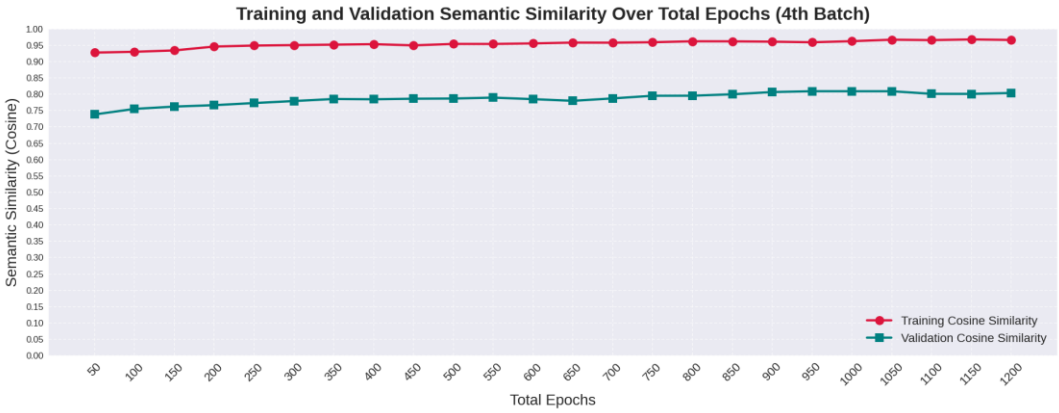
Σχήμα 4.1 Γραφική αποτελεσμάτων από την πρώτη εκπαίδευση του μοντέλου



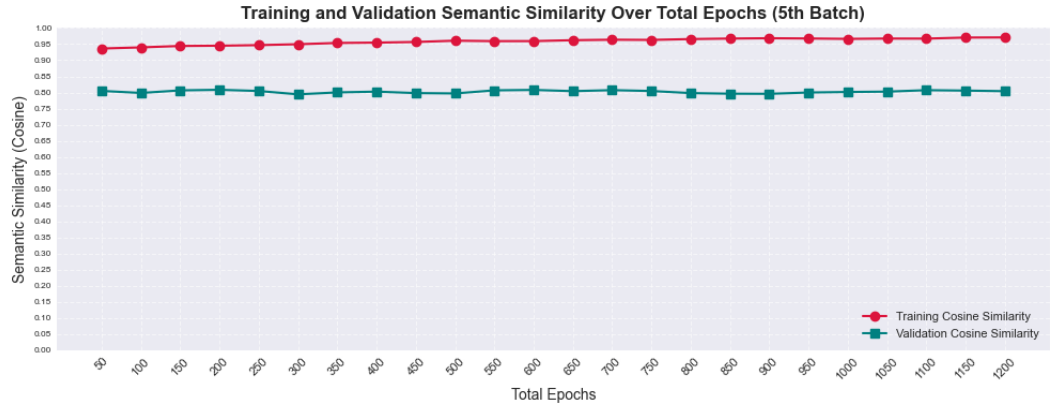
Σχήμα 4.2 Γραφική αποτελεσμάτων από την δεύτερη εκπαίδευση του μοντέλου



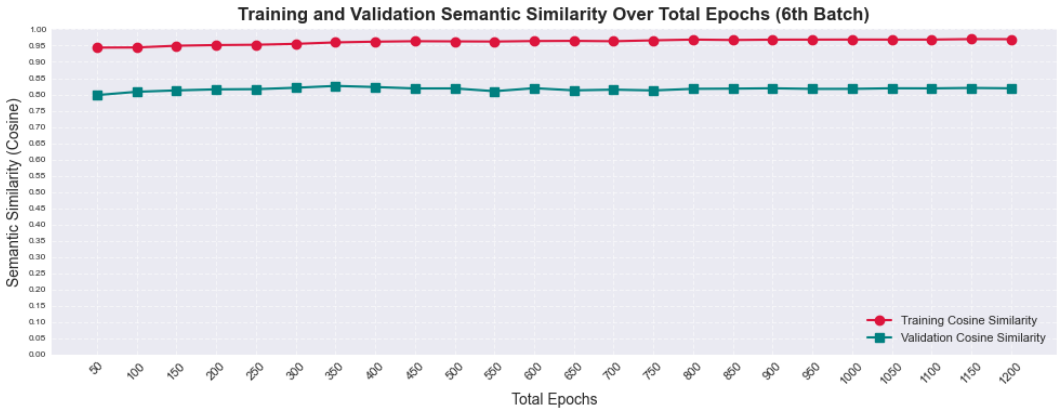
Σχήμα 4.3 Γραφική αποτελεσμάτων από την τρίτη εκπαίδευση του μοντέλου



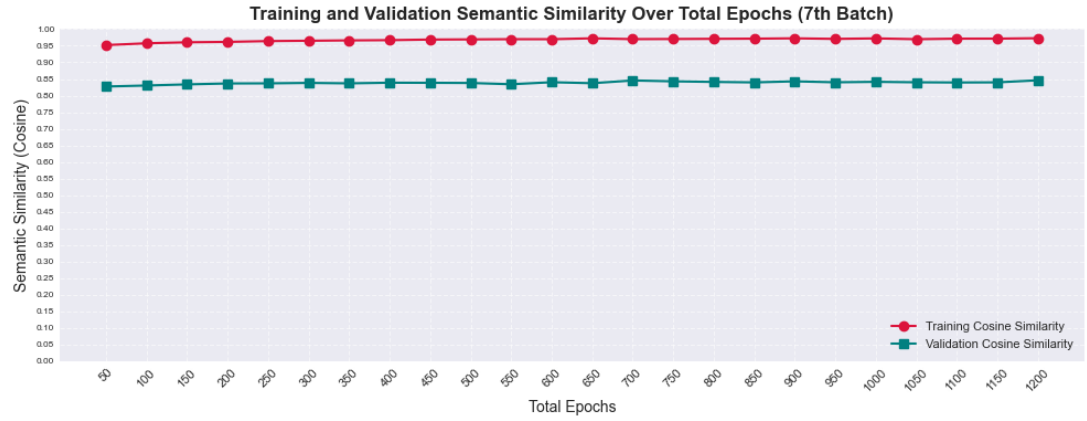
Σχήμα 4.4 Γραφική αποτελεσμάτων από την τέταρτη εκπαίδευση του μοντέλου



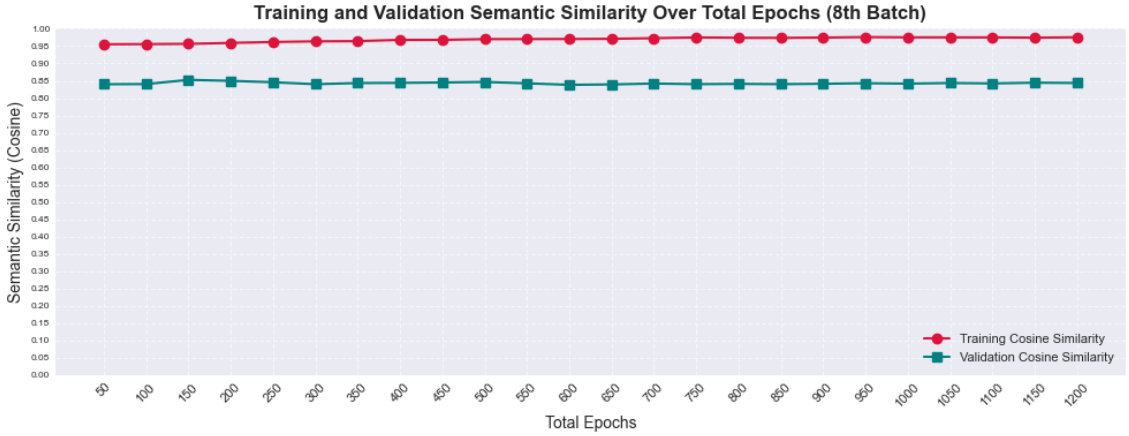
Σχήμα 4.5 Γραφική αποτελεσμάτων από την Πέμπτη εκπαίδευση του μοντέλου



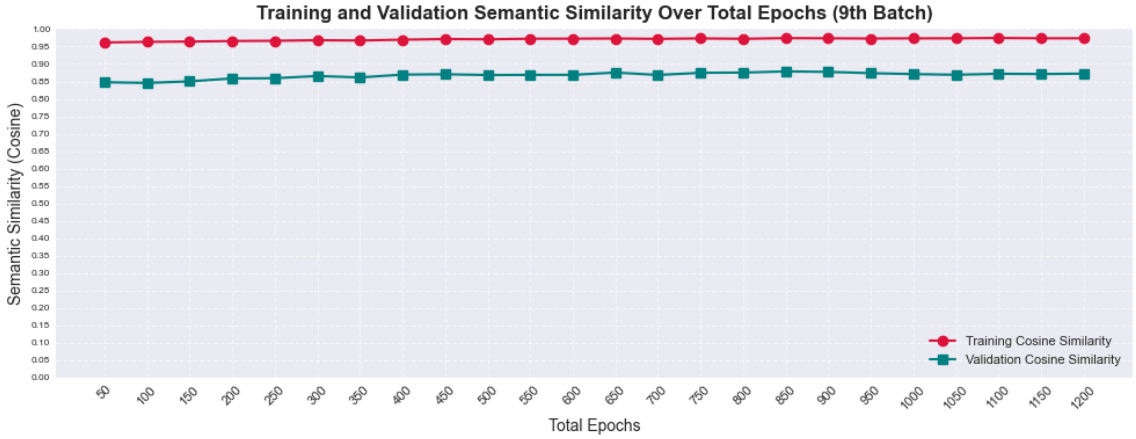
Σχήμα 4.6 Γραφική αποτελεσμάτων από την έκτη εκπαίδευση του μοντέλου



Σχήμα 4.7 Γραφική αποτελεσμάτων από την έβδομη εκπαίδευση του μοντέλου



Σχήμα 4.8 Γραφική αποτελεσμάτων από την όγδοη εκπαίδευση του μοντέλου



Σχήμα 4.9 Γραφική αποτελεσμάτων από την ένατη εκπαίδευση του μοντέλου

4.3 Μετρήσεις BLEU, WER, CER

Εκτός από την σημασιολογική ομοιότητα ο πίνακας (4,10), παρουσιάζει τρεις μετρικές αξιολόγησης BLEU, WER και CER που υπολογίστηκαν μετά το τέλος διαδοχικών φάσεων εκπαίδευσης (4,5,6,7,8). Συγκεκριμένα κάθε γραμμή του πίνακα αντιστοιχεί στην αξιολόγηση του μοντέλου πριν την ένταξη του επόμενου διαδοχικά batch (training set). Ο έλεγχος ποιότητας με αυτές τις τεχνικές γίνεται συγκρίνοντας τις μεταφράσεις του μοντέλου με τα checkpoints μέχρι την φάση X και το training set που θα χρησιμοποιηθεί για να εκπαιδεύσει το μοντέλο για την φάση X+1. Με αυτό το τρόπο μπορούμε να επιβεβαιώσουμε την βελτίωση των μεταφράσεων του μοντέλου σε νέα ανεξερεύνητα δεδομένα.

Αναλύοντας τα πιο κάτω δεδομένα παρατηρούμε ότι με το τέλος κάθε φάσης εκπαίδευσης το μοντέλο βελτιωνόταν και στις 3 μετρήσεις στα πρώτα στάδια και ακολούθως σταθεροποιούταν στα τελευταία batch.

Το BLEU score: είναι ένας δείκτης που μετράει πόσο κοντά είναι η μετάφραση του μοντελου σε μία ή περισσότερες μεταφράσεις απο τους annotators συγκρίνοντας αλληλουχίες λέξεων μέχρι μήκους 4. Υπολογίζει την ακρίβεια (precision) των n-γραμμάτων που εμφανίζονται και στις δύο μεταφράσεις, ενώ επιβάλλει και ένα "πέναλτι συντομίας" (brevity penalty) αν η πρόταση-υποψήφια είναι πολύ μικρότερη από την αναφορά. Το τελικό σκορ κυμαίνεται από 0 έως 100 (ποσοστό): όσο πιο υψηλό είναι, τόσο περισσότερη συμφωνία (και επομένως ποιότητα) έχει η μετάφραση [11].

Το WER (Word Error Rate), που μετρά το ποσοστό σφαλμάτων σε επίπεδο λέξης

Το CER (Character Error Rate), που υπολογίζει το ποσοστό σφαλμάτων σε επίπεδο χαρακτήρα [12].

Checkpoint (Batches Trained)	Evaluated on Batch	BLEU (%)	WER (%)	CER (%)
4 (1–4)	5	53.89	39.10	30.30
5 (1–5)	6	59.17	35.23	27.88
6 (1–6)	7	56.44	37.06	29.16
7 (1–7)	8	59.82	33.64	26.58
8 (1–8)	9	59.23	33.62	26.46

Σχήμα 4.10 Πίνακας αποτελεσμάτων BLEU,WER,CER

Μετά την ολοκλήρωση της εκπαίδευσης του μοντέλου μέχρι και το 4ο batch, το μοντέλο είχε BLEU score 53,9% παίρνοντας σαν είσοδο το batch 5(training set) πριν εκπαιδευτεί σε αυτό, ενώ το error σε επίπεδο λέξης ήταν στο 39,1% και το error σε επίπεδο χαρακτήρα ήταν στο 30,3%. Αυτές οι τιμές αποτελούν την αφετηρία αξιολόγησης της γενίκευσης του μοντέλου μετά από 4 φάσης εκπαίδευσης.

Καθώς το μοντέλο εμπλουτίστηκε με τα δεδομένα του 5ου batch (μέσω fine-tuning χρησιμοποιώντας το 5ο batch), η επίδοσή του στο επόμενο άγνωστο σύνολο (batch 6) παρουσίασε σημαντική άνοδο: το BLEU score αυξήθηκε στο 59,2%, ενώ οι τιμές σφάλματος ανά λέξη και CER ανά χαρακτήρα σε ~35,2% και ~27,9% αντίστοιχα. Η βελτίωση αυτή (άνοδος ~5 μονάδων BLEU και μείωση ~4 μονάδων WER) καταδεικνύει ότι η προσθήκη του νέου training set ήταν ωφέλιμη, καλύπτοντας κενά γνώσης του μοντέλου και επιτρέποντάς του να μεταφράζει πιο σωστά τα επόμενα δεδομένα. Η μεγαλύτερη βελτίωση ποιότητας μεταφράσεων σημειώθηκε σε αυτό το στάδιο σύμφωνα και με τις 3 μετρήσεις.

Στο επόμενο στάδιο, πριν την ένταξη του 7ου batch, το BLEU score στο batch 7 σημείωσε μια μικρή πτώση στο 56,4%, παρόλο που ο WER συνέχισε να βελτιώνεται (μειώθηκε στο 33,5% από 35,2%) και ο CER αυξήθηκε ελαφρά στο 29,2% από 27,9%. Η ελαφρά αυτή ασυνέπεια (μείωση BLEU παρά τη μείωση του WER) υποδηλώνει ότι το 7ο batch περιείχε πιθανώς πιο εξειδικευμένες προτάσεις, στις οποίες το μοντέλο (έως το 6ο batch) δυσκολεύτηκε να πετύχει υψηλό βαθμό ακρίβειας στις μεταφράσεις σε σύγκριση με τις αναφορές. Με άλλα λόγια, τα νέα δεδομένα που επιλέχθηκαν μέσω ενεργητικής μάθησης στο 7ο batch φαίνεται να ήταν ιδιαιτέρως δύσκολα ή απρόβλεπτα για το μοντέλο, γεγονός που οδήγησε σε ελαφρά χαμηλότερο BLEU πριν τα μάθει. Αυτό είναι σύμφωνο με το στόχο της ενεργητικής μάθησης: επιλογή παραδειγμάτων στα οποία το μοντέλο είναι πιο αβέβαιο ή κάνει λάθη, ώστε εκπαιδεύοντάς το σε αυτά να βελτιωθεί εκεί που υστερεί.

Πράγματι, αφού το μοντέλο εκπαιδεύτηκε στο 7ο batch, η απόδοσή του στο 8ο batch ανέκαμψε και πάλι σε υψηλά επίπεδα. Ο BLEU στο batch 8 έφτασε το 59,8%, δηλαδή ανέκτησε και ξεπέρασε το προηγούμενο υψηλό, ενώ οι δείκτες σφάλματος WER και CER βελτιώθηκαν περαιτέρω (33,6% και 26,6% αντίστοιχα, οι χαμηλότερες τιμές

σφάλματος που καταγράφηκαν). Αυτή η ανάκαμψη επιβεβαιώνει ότι η εκπαίδευση στο 7ο batch όντως βοήθησε το μοντέλο να καλύψει τα κενά του και να γενικεύσει καλύτερα, πετυχαίνοντας την καλύτερη ποιότητά του μέχρι εκείνο το σημείο.

Τέλος, πριν την εκπαίδευση στο 9ο (τελευταίο) batch, η αξιολόγηση στο batch 9 έδειξε τιμές παρόμοιες με του προηγούμενου βήματος: BLEU ~59,2%, WER ~33,6% και CER ~26,5%, ουσιαστικά αμετάβλητες συγκριτικά με το batch 8. Αυτό υποδηλώνει ότι η προσθήκη του 8ου batch είχε φέρει το μοντέλο σε ένα επίπεδο όπου η απόδοση του είχε πλησιάσει σε κορεσμό ως προς τη συγκεκριμένη διαδικασία ενεργητικής μάθησης. Η πολύ μικρή διαφορά (π.χ. μια ανεπαίσθητη μείωση 0,6 στο BLEU από 8->9 και σχεδόν ταυτόσημοι WER/CER) καταδεικνύει ότι τα περιεχόμενα του 9ου batch δεν πρόσθεσαν σημαντικά νέα γνώση που να μεταφράζεται σε άμεση βελτίωση στους δείκτες απόδοσης σύμφωνα με τις 3 μετρήσεις αυτές.

4.4 Συνολική Αποτίμηση:

Συνοψίζοντας, τα αποτελέσματα επιβεβαιώνουν ότι η στρατηγική εκπαίδευσης με ενεργητική μάθηση και σταδιακό fine-tuning βελτίωσε αισθητά την απόδοση του μεταφραστικού μοντέλου. Σε κάθε νέο batch, το μοντέλο κατόρθωνε γενικά να αυξήσει τη σημασιολογική ομοιότητα των μεταφράσεών του (όπως φάνηκε από τις υψηλές τιμές cosine similarity) και να μειώσει τα ποσοστά λάθους (WER, CER) διατηρώντας ή αυξάνοντας το BLEU στις περισσότερες περιπτώσεις. Οι μεγαλύτερες βελτιώσεις σημειώθηκαν στα πρώτα επιλεγμένα batches, όπου το μοντέλο μάθαινε εντελώς άγνωστες έως τότε γλωσσικές δομές. Καθώς περισσότερα δεδομένα προστέθηκαν και το μοντέλο έγινε πιο ολοκληρωμένο, οι αποδόσεις του προσέγγισαν ένα όριο, παρουσιάζοντας μια τάση σύγκλισης. Η απουσία αξιοσημείωτου χάσματος μεταξύ απόδοσης σε training και validation υποδηλώνει ότι το μοντέλο δεν υπέρεκπαιδεύτηκε στα δεδομένα του κάθε batch εις βάρος της γενίκευσης. Αντίθετα, η εκπαίδευση φαίνεται να έχει συγκλίνει σε ένα σημείο όπου το μοντέλο μαθαίνει μεν τα νέα δεδομένα, αλλά οι βελτιώσεις είναι πλέον οριακές. Αυτό το αποτέλεσμα είναι σύμφωνο με τη θεωρία της ενεργητικής μάθησης, όπου τα μεγαλύτερα οφέλη παρουσιάζονται νωρίς, ενώ μετά από αρκετούς κύκλους οι απομένουσες μη γνωστές πτυχές του προβλήματος μειώνονται. Συμπερασματικά, η απόδοση του μοντέλου κρίνεται

ικανοποιητική και σταθερή στους τελευταίους ελέγχους, γεγονός που καταδεικνύει την επιτυχή εφαρμογή αυτής της εκπαιδευτικής στρατηγικής. Οι τάσεις αυτές υποδηλώνουν ότι για περαιτέρω ουσιώδη βελτίωση θα απαιτηθεί είτε περισσότερη ποικιλία δεδομένων είτε διαφορετικές προσεγγίσεις μοντελοποίησης, καθώς το υπάρχον μοντέλο φαίνεται να έχει αξιοποιήσει στο έπακρο τις πληροφορίες που του παρείχαν τα διαθέσιμα επιλεγμένα δεδομένα

4.5 Πρόσβαση στο μοντέλο

Για άμεση πρόσβαση στο fine-tuned μοντέλο διατίθεται το παρακάτω link: https://huggingface.co/ZiartisNikolas/NMT-cypriot-dialect-to-greek

Κεφάλαιο 5

Συμπεράσματα

5.1 Γενικά Συμπεράσματα

Η παρούσα διπλωματική εργασία ανέπτυξε ένα προσαρμοστικό σύστημα Νευρωνικής Μηχανικής Μετάφρασης (NMT) που στοχεύει στη μετάφραση από την Κυπριακή διάλεκτο στην κοινή Ελληνική. Το σύστημα βασίστηκε στη σταδιακή εκπαίδευση (fine-tuning) ενός προεκπαιδευμένου μοντέλου Marian NMT, με διαδοχική ενσωμάτωση επιπλέον μεταφρασμένων προτάσων. Για την επιλογή των πιο αντιπροσωπευτικών προτάσεων χρησιμοποιήθηκε ο αλγόριθμος k-μέσων clustering, αξιοποιώντας τα embeddings του ελληνικού LLM Meltemi 7B. Με αυτόν τον τρόπο εφαρμόστηκαν τεχνικές ενεργής μάθησης ώστε να επιλεγούν προς μετάφραση δείγματα που μεγιστοποιούν τη μάθηση του μοντέλου με το ελάχιστο δυνατό κόστος επισήμανσης δεδομένων. Οι πειραματικές αξιολογήσεις έδειξαν ότι το σύστημα υπερέβη σημαντικά τις αρχικές επιδόσεις του baseline μοντέλου. Συγκεκριμένα, παρατηρήθηκε σταδιακή βελτίωση στις μετρικές μετάφρασης: η τιμή του BLEU αυξήθηκε με κάθε βρόγχο ενεργού μάθησης, ενώ οι τιμές WER και CER μειώθηκαν αντίστοιχα, υποδηλώνοντας αύξηση της ποιότητας των μεταφράσεων. Επιπλέον, ο υπολογισμός της συνημιτονοειδούς ομοιότητας μεταξύ των ενσωματωμάτων των μεταφράσεων επιβεβαίωσε τη διατήρηση του βασικού νοήματος, γεγονός που δείχνει ότι η μεθοδολογία συμβάλλει σε ποιοτικές μεταφράσεις. Τα αποτελέσματα αυτά

υπογραμμίζουν την αποτελεσματικότητα της επιλεγμένης μεθοδολογίας στην αντιμετώπιση των ιδιαιτεροτήτων της κυπριακής διαλέκτου.

5.2 Περιορισμοί και Προκλήσεις

Το προτεινόμενο σύστημα αντιμετωπίζει ορισμένους περιορισμούς και προκλήσεις:

- **Περιορισμένη διαθεσιμότητα δεδομένων:** Η συλλογή επαρκών παράλληλων κειμένων Κυπριακής διαλέκτου – Ελληνικών είναι δύσκολη. Η αρχική βάση δεδομένων είναι σχετικά μικρή, περιορίζοντας την κάλυψη ιδιωματικών εκφράσεων, με αποτέλεσμα το μοντέλο να μην εκπροσωπεί πλήρως όλες τις πτυχές της διαλέκτου.
- **Εξάρτηση από το αρχικό μοντέλο:** Το σύστημα βασίζεται στο fine-tuning του υπαρκτού Marian NMT. Ελλείψεις ή σφάλματα στο αρχικό μοντέλο ή στα δεδομένα εκπαίδευσης μπορούν να μεταφερθούν και στις βελτιωμένες εκδόσεις, περιορίζοντας τη μέγιστη ακρίβεια που μπορεί να επιτευχθεί.
- **Πολυπλοκότητα της Κυπριακής διαλέκτου:** Η διάλεκτος διαθέτει μοναδικό λεξιλόγιο, φωνολογία και γραμματικές ιδιαιτερότητες. Η σωστή απόδοση ιδιωμάτων, τοπικών εκφράσεων και σύνθετων γραμματικών δομών παραμένει πρόκληση, καθώς πολλές φορές δεν ακολουθούν τα πρότυπα της κοινής Ελληνικής.
- **Περιορισμοί μετρικών αξιολόγησης:** Οι κλασικές μετρικές BLEU, WER και CER ενδέχεται να μην αποτυπώνουν πλήρως την ποιότητα της μετάφρασης, ειδικά όταν υπάρχουν εναλλακτικές ορθές μεταφράσεις. Συγκεκριμένα, διαφορετικές φράσεις με το ίδιο νόημα μπορεί να θεωρηθούν λάθη, γεγονός που μπορεί να υποεκτιμήσει τη βελτίωση που επιτυγχάνει το μοντέλο.
- **Υπολογιστικές απαιτήσεις:** Η διαδικασία ενεργού μάθησης με διαδοχική επανεκπαίδευση του μοντέλου είναι απαιτητική σε υπολογιστικούς πόρους και χρόνο. Καθώς αυξάνονται τα δεδομένα και η πολυπλοκότητα του μοντέλου, το κόστος εκπαίδευσης μπορεί να αυξηθεί σημαντικά, απαιτώντας σχεδιασμό βελτιστοποίησης των πόρων.

5.3 Πρακτικές Εφαρμογές

Το σύστημα μετάφρασης παρουσιάζει πολλαπλές πρακτικές εφαρμογές σε κυπριακά συμφραζόμενα:

- **Κυβέρνηση:** Χρήση για τη μετάφραση εγγράφων, αιτημάτων και επικοινωνιών που υποβάλλονται στην Κυπριακή διάλεκτο από πολίτες προς κρατικές υπηρεσίες. Με τον τρόπο αυτό διασφαλίζεται η ισότιμη πρόσβαση σε δημόσιες υπηρεσίες και διευκολύνεται η επικοινωνία μεταξύ πολιτών και διοίκησης.
- **Εκπαίδευση:** Υποστήριξη δημιουργίας εκπαιδευτικού υλικού που γεφυρώνει τη διάλεκτο με την καθομιλουμένη Ελληνική, όπως μεταφρασμένα κείμενα, λεξιλόγια και βοηθητικά εγχειρίδια. Μαθητές και φοιτητές που χρησιμοποιούν τη διάλεκτο στο καθημερινό τους λεξιλόγιο θα επωφεληθούν μέσω βελτιωμένης κατανόησης επίσημων διδακτικών κειμένων.
- **Πολιτισμός και κληρονομιά:** Διατήρηση και προβολή της κυπριακής πολιτιστικής κληρονομιάς μέσω της ψηφιοποίησης και μετάφρασης λαογραφικών κειμένων, τραγουδιών και παραδοσιακών αφηγήσεων στην κοινή Ελληνική. Μεταφράζοντας τέτοιο περιεχόμενο, προωθείται η διάχυση της παράδοσης σε ευρύτερο κοινό και διασφαλίζεται η προστασία της γλωσσικής κληρονομιάς.
- **Μέσα ενημέρωσης και επικοινωνία:** Διευκόλυνση πρόσβασης σε τοπικά ειδησεογραφικά άρθρα, αναρτήσεις και κοινωνικά δίκτυα που γράφονται στη διάλεκτο, με μεταφράσεις στην κοινή Ελληνική. Αυτό επιτρέπει σε ομιλητές της καθομιλουμένης να ενημερώνονται και να κατανοούν περιεχόμενο που αρχικά απευθύνεται σε χρήστες της κυπριακής.
- **Τουρισμός και φιλοξενία:** Παροχή εφαρμογών μετάφρασης σε πραγματικό χρόνο για τουρίστες που επισκέπτονται την Κύπρο, ώστε να κατανοούν τοπικές επιγραφές, οδηγίες και εκφράσεις στη διάλεκτο. Αυτό ενισχύει την εμπειρία των επισκεπτών και προωθεί το πολιτιστικό αποτύπωμα της τοπικής ταυτότητας.

5.4 Μελλοντικές Κατευθύνσεις

Με βάση τα αποτελέσματα και τους περιορισμούς της εργασίας, προτείνονται οι εξής κατευθύνσεις για μελλοντική έρευνα:

- **Χρήση πιο σύγχρονων μοντέλων (embeddings):** Η αξιοποίηση νεότερων ή πολυγλωσσικών LLM (π.χ. νεότερες εκδόσεις του Meltemi, πολύγλωσσα BERT/GPT) για την παραγωγή ενσωματωμάτων μπορεί να βελτιώσει την ακρίβεια του clustering και της σημασιολογικής αξιολόγησης. Ισχυρότερα embeddings θα επιτρέψουν καλύτερη αναπαράσταση των εννοιών και θα ενισχύσουν τη διαδικασία επιλογής δειγμάτων.
- **Διεύρυνση γλωσσικής κάλυψης:** Συλλογή και ενσωμάτωση περισσότερων δεδομένων από διαφορετικές πηγές της κυπριακής διαλέκτου και της ευρύτερης Ελληνικής. Η πληρέστερη βάση δεδομένων θα ενισχύσει τη γενίκευση του μοντέλου, επιτρέποντάς του να ανταποκρίνεται καλύτερα σε ποικίλα γλωσσικά φαινόμενα και άγνωστες εκφράσεις.

Οι προτεινόμενες κατευθύνσεις εναρμονίζονται με τις σύγχρονες τάσεις έρευνας στην NMT και αναμένεται να βελτιώσουν περαιτέρω την ακρίβεια και την ευελιξία του συστήματος. Συμπερασματικά, η παρούσα εργασία θέτει ισχυρές βάσεις για την εξέλιξη της μετάφρασης διαλέκτων με απτό όφελος σε επιστημονικό και κοινωνικό επίπεδο.

Βιβλιογραφία

[1] Active learning for interactive machine translation, pages: 1-8
https://aclanthology.org/E12-1025/

[2] A Survey of Machine Translation Methods, pages: 2-4
file:///C:/Users/ziart/Downloads/A_Survey_of_Machine_Translation_Methods.pdf

[3] Sequence to Sequence Learning with Neural Networks, pages: 2-6
https://arxiv.org/pdf/1409.3215

[4] Active Learning for Multilingual Statistical Machine Translation, pages: 1-5
https://aclanthology.org/P09-1021/

[5] Data Clustering: A Review, pages: 264–323
https://dl.acm.org/doi/pdf/10.1145/331499.331504

[6] Tan, P.-N., Steinbach, M., & Kumar, V. (2018). Introduction to Data Mining (2nd ed.). Pearson, pages: 496-514
https://www.ceom.ou.edu/media/docs/upload/Pang-Ning_Tan_Michael_Steinbach_Vipin_Kumar_-_Introduction_to_Data_Mining-Pe_NRDK4fi.pdf

[7] Meltemi: The first open Large Language Model for Greek, pages: 1-7
https://arxiv.org/pdf/2407.20743

[8] OPUS-MT – Building open translation services for the World, pages: 1-2
https://aclanthology.org/2020.eamt-1.61.pdf

[9] A Survey on Low-Resource Neural Machine Translation, pages: 1-8
https://arxiv.org/abs/2107.04239

[10] Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, pages: 1-4
https://arxiv.org/abs/1908.10084

[11] BLEU: a Method for Automatic Evaluation of Machine Translation, pages: 1-6
https://aclanthology.org/P02-1040.pdf

[12] Word Error Rates: Decomposition over POS Classes and Applications for Error Analysis, pages: 1-5
https://aclanthology.org/W07-0707.pdf

[13] Attention Is All You Need, pages: 4-8
https://arxiv.org/pdf/1706.03762

[14] Six Challenges for Neural Machine Translation, pages: 4-6
https://aclanthology.org/W17-3204.pdf

[15] Active Learning Literature Survey Burr Settles Technical Report #1648 January 2009, pages: 11-15
https://minds.wisconsin.edu/bitstream/handle/1793/60660/TR1648.pdf?sequence=1&isAllowed=y

[16] A K-Means Clustering Algorithm J. A. Hartigan , M. A. Wong, pages: 100-108
https://academic.oup.com/jrsssc/article/28/1/100/6953842

[17] Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping, pages: 3-8
https://arxiv.org/pdf/2002.06305

Παράρτημα Α

Fine-Tuning and Training Script

```
from transformers import MarianMTModel, MarianTokenizer, Trainer,
TrainingArguments, DataCollatorForSeq2Seq, AutoTokenizer, AutoModel
import json
from torch.utils.data import Dataset
from google.colab import drive
import torch
import os
import shutil
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import cosine_similarity

# Mount Google Drive
drive.mount('/content/drive')

# Set the output directories
output_dir = "./fine_tuned_results"
# Temporarily save checkpoints locally
model_save_dir = "/content/drive/MyDrive/Thesis-
Coding/fine_tuned_model"
# Save the final model to Google Drive

# Check if GPU is available
device = "cuda" if torch.cuda.is_available() else "cpu"

# Step 1: Load the Fine-Tuned Model or Original Model
if os.path.exists(model_save_dir):
    print("Loading fine-tuned model from Google Drive...")
    model = MarianMTModel.from_pretrained(model_save_dir,
local_files_only=True).to(device)
    tokenizer = MarianTokenizer.from_pretrained(model_save_dir,
local_files_only=True)
else:
    print("Loading base model for initial training...")
    model_name = "Helsinki-NLP/opus-mt-en-grk"
    tokenizer = MarianTokenizer.from_pretrained(model_name)
    model = MarianMTModel.from_pretrained(model_name).to(device)

# Step 2: Freeze all layers except the last encoder layers, decoder,
and lm_head
for param in model.parameters():
    param.requires_grad = False

# Unfreeze the last 3 encoder layers
for param in model.model.encoder.layers[-3:].parameters():
    param.requires_grad = True

# Unfreeze the decoder
for param in model.model.decoder.parameters():
    param.requires_grad = True

# Unfreeze the output projection layer
for param in model.lm_head.parameters():
    param.requires_grad = True

# Step 3: Define the dataset class
class TranslationDataset(Dataset):
```

```
    def __init__(self, tokenizer, data, max_length=512):
        self.tokenizer = tokenizer
        self.data = data  # In-memory data
        self.max_length = max_length

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

    def __getitem__(self, idx):
        source = self.data[idx]["source"]
        target = self.data[idx]["target"]

        inputs = self.tokenizer(
            source, max_length=self.max_length, truncation=True,
padding="max_length", return_tensors="pt"
        )
        targets = self.tokenizer(
            target, max_length=self.max_length, truncation=True,
padding="max_length", return_tensors="pt"
        )

        inputs["labels"] = targets["input_ids"]
        return {key: val.squeeze(0) for key, val in inputs.items()}

# Step 4: Load the datasets
with open("training_set.json", "r", encoding="utf-8") as f:
    train_data = json.load(f)

with open("validation_set.json", "r", encoding="utf-8") as f:
    validation_data = json.load(f)

train_dataset = TranslationDataset(tokenizer, train_data)
validation_dataset = TranslationDataset(tokenizer, validation_data)

# training arguments
def get_training_args(output_dir):
    return TrainingArguments(
        output_dir=output_dir,
        num_train_epochs=50,
        per_device_train_batch_size=4,
        gradient_accumulation_steps=8,
        max_steps=50,
        eval_steps=50,
        save_total_limit=1,
        learning_rate=1e-5,
        fp16=True,
        logging_dir=output_dir + "/logs",
        logging_steps=50,
        report_to="none",
    )

# Step 6: Validation function using semantic similarity
# Load a pretrained multilingual Sentence Transformer model for
semantic similarity
semantic_model_name = "sentence-transformers/paraphrase-multilingual-
mpnet-base-v2"
semantic_tokenizer =
AutoTokenizer.from_pretrained(semantic_model_name)
semantic_model =
AutoModel.from_pretrained(semantic_model_name).to(device)
```

```
def calculate_semantic_similarity(reference, hypothesis):
    # Tokenize and encode the reference and hypothesis sentences
    ref_tokens = semantic_tokenizer(reference, return_tensors="pt",
truncation=True, padding=True).to(device)
    hyp_tokens = semantic_tokenizer(hypothesis, return_tensors="pt",
truncation=True, padding=True).to(device)

    # Extract embeddings from the model
    ref_embedding =
semantic_model(**ref_tokens).last_hidden_state.mean(dim=1)
    hyp_embedding =
semantic_model(**hyp_tokens).last_hidden_state.mean(dim=1)

    # Calculate cosine similarity
    cosine_sim = cosine_similarity(
        ref_embedding.cpu().detach().numpy(),
        hyp_embedding.cpu().detach().numpy()
    )[0][0]

    return cosine_sim

def validate_with_similarity(model, tokenizer, validation_data):
    similarities = []

    for item in validation_data:
        source = item["source"]
        target = item["target"]

        # Generate hypothesis
        inputs = tokenizer(source, return_tensors="pt",
max_length=512, truncation=True).to(device)
        outputs = model.generate(**inputs)
        decoded_output = tokenizer.decode(outputs[0],
skip_special_tokens=True)

        # Calculate semantic similarity
        sim = calculate_semantic_similarity(target, decoded_output)
        similarities.append(sim)

    # Compute average similarity
    avg_similarity = sum(similarities) / len(similarities)

    return avg_similarity, similarities

# Step 7: Train and validate
def train_and_validate(model, tokenizer, train_dataset,
validation_dataset, validation_data, train_data, epochs, output_dir,
model_save_dir):
    train_losses, validation_losses, validation_similarities,
training_similarities = [], [], [], []

    for epoch in range(epochs):
        print(f"Starting epoch {epoch + 1}...")
        training_args = get_training_args(output_dir)
        data_collator = DataCollatorForSeq2Seq(
            tokenizer=tokenizer, model=model,
label_pad_token_id=tokenizer.pad_token_id
        )
```

```
        trainer = Trainer(
            model=model,
            args=training_args,
            train_dataset=train_dataset,
            eval_dataset=validation_dataset,
            data_collator=data_collator,
            tokenizer=tokenizer,
        )

        # Train
        train_result = trainer.train()
        train_losses.append(train_result.training_loss)

        # Evaluate validation loss
        validation_metrics = trainer.evaluate(validation_dataset)
        validation_loss = validation_metrics["eval_loss"]
        validation_losses.append(validation_loss)

        # Compute semantic similarity for validation
        avg_val_sim, _ = validate_with_similarity(model, tokenizer,
validation_data)
        validation_similarities.append(avg_val_sim)

        # Compute semantic similarity for training
        avg_train_sim, _ = validate_with_similarity(model, tokenizer,
train_data)
        training_similarities.append(avg_train_sim)

        print(
            f"Epoch {epoch + 1} - Train Loss:
{train_result.training_loss:.4f}, "
            f"Validation Loss: {validation_loss:.4f}, "
            f"Validation Cosine Similarity: {avg_val_sim:.4f}, "
            f"Training Cosine Similarity: {avg_train_sim:.4f}"
        )

    # Plot Training vs. Validation similarity
    plot_training_vs_validation_similarity(training_similarities,
validation_similarities, epochs)

    # Save the final model
    model.save_pretrained(model_save_dir)
    tokenizer.save_pretrained(model_save_dir)
    shutil.rmtree(output_dir, ignore_errors=True)
    print("Model saved to Google Drive.")

train_and_validate(
    model=model,
    tokenizer=tokenizer,
    train_dataset=train_dataset,
    validation_dataset=validation_dataset,
    validation_data=validation_data,
    train_data=train_data,
    epochs=24,
    output_dir=output_dir,
    model_save_dir=model_save_dir
)
```

Παράρτημα Β

Cluster-Based Selection for Annotation

```
import pandas as pd
import numpy as np
import torch
from sklearn.cluster import KMeans
from sklearn.metrics.pairwise import euclidean_distances
from transformers import AutoTokenizer, AutoModel
import json

# Load Labeled Data from Multiple Files
def load_labeled_data(files):
    labeled_sentences = set()
    for file_path in files:
        with open(file_path, 'r', encoding='utf-8') as file:
            labeled_data = json.load(file)
            labeled_sentences.update(item['source'] for item in
labeled_data)
    return labeled_sentences

# Define labeled data files
labeled_data_files =
['combined_4.json','labeled_data5.json','labeled_data6.json','labeled_
data7.json']
labeled_sentences = load_labeled_data(labeled_data_files)

# Load the Original Dataset
csv_path = 'splitted_sentences_with_authors.csv'
data = pd.read_csv(csv_path)
all_sentences = data['sentence'].tolist()

# Filter Out Labeled Sentences
remaining_sentences = [sentence for sentence in all_sentences if
sentence not in labeled_sentences]
print(f"Remaining sentences: {len(remaining_sentences)}")

# Load Pre-trained Model
model_name = "ilsp/Meltemi-7B-Instruct-v1.5"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name, device_map="auto",
torch_dtype=torch.float16, offload_folder="offload")

# Function to Extract Embeddings
def extract_embeddings(sentences, batch_size=16):
    embeddings = []
    for i in range(0, len(sentences), batch_size):
        batch = sentences[i:i+batch_size]
        inputs = tokenizer(batch, return_tensors="pt", padding=True,
truncation=True, max_length=512)
        inputs = {key: val.to(model.device) for key, val in
inputs.items()}
        with torch.no_grad():
            outputs = model(**inputs)
            torch.cuda.empty_cache()  # Clear GPU cache
            batch_embeddings =
outputs.last_hidden_state.mean(dim=1).cpu().numpy()
            embeddings.extend(batch_embeddings)
    return np.array(embeddings)
```

```
# Extract Embeddings for Remaining Sentences
sentence_embeddings = extract_embeddings(remaining_sentences)

# Perform K-Means Clustering
num_clusters = 50
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(sentence_embeddings)
kmeans_labels = kmeans.labels_
kmeans_centroids = kmeans.cluster_centers_

# Find Closest Sentences to Centroids
def find_closest_sentences(embeddings, centroids, sentences):
    closest_sentences = []
    for centroid in centroids:
        distances = euclidean_distances([centroid], embeddings)
        closest_idx = np.argmin(distances)
        closest_sentences.append(sentences[closest_idx])
    return closest_sentences

next_batch_sentences = find_closest_sentences(sentence_embeddings,
kmeans_centroids, remaining_sentences)

# Print and Save the Selected Sentences
print("Next Batch of Sentences:")
for idx, sentence in enumerate(next_batch_sentences):
    print(f"Cluster {idx}: {sentence}")

# Save the Cluster Assignments and Next Batch Sentences to JSON Files
clustered_data = [
    {"sentence": sentence, "cluster": int(cluster)}
    for sentence, cluster in zip(remaining_sentences, kmeans_labels)
]

next_batch_data = [
    {"source": sentence, "target": "", "cluster": idx}
    for idx, sentence in enumerate(next_batch_sentences)
]

output_clustered_path = 'clustered_sentences.json'
output_next_batch_path = 'next_batch_sentences.json'

with open(output_clustered_path, 'w', encoding='utf-8') as file:
    json.dump(clustered_data, file, ensure_ascii=False, indent=4)

with open(output_next_batch_path, 'w', encoding='utf-8') as file:
    json.dump(next_batch_data, file, ensure_ascii=False, indent=4)

print(f"Cluster assignments saved to {output_clustered_path}")
print(f"Next batch saved to {output_next_batch_path}")
```

Παράρτημα Γ

Evaluation with BLEU, WER, and CER Metrics

```
from google.colab import drive
drive.mount('/content/drive')

# 2. Εισαγωγές
import os
import json
import math
from collections import Counter

import torch
from transformers import MarianMTModel, MarianTokenizer

# 3. Ορισμός μετρικών (Levenshtein, CER, WER, BLEU)

def edit_distance(a, b):
    """Υπολογίζει την απόσταση Levenshtein μεταξύ δύο ακολουθιών
(χαρακτήρες ή tokens)."""
    n, m = len(a), len(b)
    dp = [[0]*(m+1) for _ in range(n+1)]
    for i in range(n+1):
        dp[i][0] = i
    for j in range(m+1):
        dp[0][j] = j
    for i in range(1, n+1):
        for j in range(1, m+1):
            cost = 0 if a[i-1] == b[j-1] else 1
            dp[i][j] = min(
                dp[i-1][j] + 1,       # διαγραφή
                dp[i][j-1] + 1,       # εισαγωγή
                dp[i-1][j-1] + cost   # αντικατάσταση
            )
    return dp[n][m]

def cer(preds, refs):
    """Υπολογίζει το Character Error Rate (%) για λίστα προβλέψεων
και αναφορών."""
    total_edits = 0
    total_chars = 0
    for p, r in zip(preds, refs):
        total_edits += edit_distance(p, r)
        total_chars += len(r)
    return (total_edits / total_chars) * 100 if total_chars > 0 else
0.0

def wer(preds, refs):
    """Υπολογίζει το Word Error Rate (%) για λίστα προβλέψεων και
αναφορών."""
    total_edits = 0
    total_words = 0
    for p, r in zip(preds, refs):
        p_tokens = p.split()
        r_tokens = r.split()
        total_edits += edit_distance(p_tokens, r_tokens)
        total_words += len(r_tokens)
    return (total_edits / total_words) * 100 if total_words > 0 else
0.0
```

```
def corpus_bleu(preds, refs, max_n=4):
    """Υπολογίζει το corpus-level BLEU score (%) με n-grams από 1 έως
max_n."""
    precisions = []
    for n in range(1, max_n + 1):
        clipped = 0
        total   = 0
        for p, r in zip(preds, refs):
            p_ngrams = Counter(tuple(p.split()[i:i+n]) for i in
range(len(p.split())-n+1))
            r_ngrams = Counter(tuple(r.split()[i:i+n]) for i in
range(len(r.split())-n+1))
            for ng, cnt in p_ngrams.items():
                clipped += min(cnt, r_ngrams.get(ng, 0))
            total += sum(p_ngrams.values())
        precisions.append(clipped/total if total > 0 else 0.0)

    # Brevity penalty
    pred_len = sum(len(p.split()) for p in preds)
    ref_len  = sum(len(r.split()) for r in refs)
    bp = math.exp(1 - ref_len/pred_len) if (pred_len > 0 and pred_len
< ref_len) else 1.0

    # Geometric mean of precisions
    if all(p > 0 for p in precisions):
        geo_mean = math.exp(sum(math.log(p) for p in precisions) /
len(precisions))
    else:
        geo_mean = 0.0

    return bp * geo_mean * 100

def evaluate_metrics(preds, refs):
    """Επιστρέφει dict με BLEU, WER, CER."""
    return {
        "bleu": corpus_bleu(preds, refs),
        "wer" : wer(preds, refs),
        "cer" : cer(preds, refs)
    }

# 4. Διαδρομές για checkpoint και JSON (batch 5 μετά
model_checkpoint_4)
checkpoint_dir =
"/content/drive/MyDrive/model_checkpoint_8/fine_tuned_model"
combined_json   =
"/content/drive/MyDrive/model_checkpoint_8/combined_9.json"

# 5. Φόρτωση μοντέλου & tokenizer (local_files_only για τοπικό drive)
device    = "cuda" if torch.cuda.is_available() else "cpu"
tokenizer = MarianTokenizer.from_pretrained(checkpoint_dir,
local_files_only=True)
model     = MarianMTModel.from_pretrained(checkpoint_dir,
local_files_only=True).to(device)

# 6. Διαβάζουμε τις μεταφράσεις αναφοράς
with open(combined_json, "r", encoding="utf-8") as f:
    data = json.load(f)
sources    = [entry["source"] for entry in data]
references = [entry["target"] for entry in data]
```

```
# 7. Παράγουμε τις προβλέψεις του μοντέλου
predictions = []
for src in sources:
    inputs     = tokenizer(src, return_tensors="pt", truncation=True,
padding=True).to(device)
    output_ids = model.generate(**inputs)
    pred       = tokenizer.decode(output_ids[0],
skip_special_tokens=True)
    predictions.append(pred)

# 8. Υπολογίζουμε και εκτυπώνουμε τα metrics
metrics = evaluate_metrics(predictions, references)
print(f"BLEU score: {metrics['bleu']:.2f}")
print(f"WER       : {metrics['wer']:.2f}%")
print(f"CER       : {metrics['cer']:.2f}%")

# 9. (Προαιρετικά) Αποθήκευση των αποτελεσμάτων
results = {
    "metrics": metrics,
    "samples": [
        {"source": s, "prediction": p, "reference": r}
        for s, p, r in zip(sources, predictions, references)
    ]
}
output_path =
"/content/drive/MyDrive/model_checkpoint_8/eval_batch9_results.json"
with open(output_path, "w", encoding="utf-8") as f:
    json.dump(results, f, ensure_ascii=False, indent=2)

print("Evaluation results saved to:", output_path)
```