NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Division of Signals, Control and Robotics
Speech and Language Processing Group

Open-Domain Dialogue Systems for Low-Resource Languages: The case of Greek

Diploma Thesis

Andreas Koukounas

Supervisors: Alexandros Potamianos
Associate Professor, NTUA

Athanasios Katsamanis
Principal Researcher, Athena RC

Athens, July 2025

[This page is left intentionally blank.]

National Technical University of Athens
School of Electrical
and Computer Engineering
Division of Signals, Control and Robotics
Speech and Language Processing Group

Open-Domain Dialogue Systems for Low-Resource Languages: The case of Greek

Diploma Thesis

Andreas Koukounas

Supervisors: Alexandros Potamianos
Associate Professor, NTUA

Athanasios Katsamanis
Principal Researcher, Athena RC

Approved by the examination committee on 1st of July 2025.

(Signature)	(Signature)	(Signature)
....	....	....
Alexandros Potamianos	Constantinos Tzafestas	Giorgos Stamou
Associate Professor	Associate Professor	Professor
NTUA	NTUA	NTUA

Athens, July 2025.

(Signature)

....

Andreas Koukounas
Graduate of School of Electrical and Computer Engineering, NTUA

Copyright © Andreas Koukounas, 2025. All rights reserved.

It is prohibited to copy, store and distribute this work, in whole or in part, for commercial purposes. Reproduction, storage, and distribution for a non-profit, educational, or research nature are permitted, provided the source of origin is indicated and the present message maintained. Questions about the use of the work for profit should be directed to the author.

The views and conclusions contained in this document are those of the author and should not be construed as representing the official positions of the National Technical University of Athens.

Περίληψη

Η συνομιλία ανθρώπου-μηχανής αποτελεί κρίσιμο έργο στην Τεχνητή Νοημοσύνη και την Επεξεργασία Φυσικής Γλώσσας. Παρά την πρόοδο στην παραγωγή διαλόγων ανοιχτού τομέα, η ανάπτυξη chatbots για μη αγγλικές γλώσσες έχει μείνει πίσω λόγω έλλειψης δεδομένων. Σε αυτή τη Διπλωματική Εργασία, μελετάμε την παραγωγή διαλόγων στα Ελληνικά, όπου τα δεδομένα εκπαίδευσης και τα προ-εκπαιδευμένα μοντέλα γλώσσας είναι περιορισμένα.

Αρχικά, παρουσιάζουμε θεωρητικό υπόβαθρο για τη μηχανική μάθηση (ML), τη βαθιά μάθηση (DL) και την επεξεργασία φυσικής γλώσσας (NLP). Στη συνέχεια, μελετάμε τα μοντέλα βασισμένα στην αρχιτεκτονική transformers που χρησιμοποιήσαμε: BERT, GPT-2, T5 και XGLM. Παρουσιάζουμε το θεωρητικό υπόβαθρο αυτών των αρχιτεκτονικών και αναλύουμε τη δημιουργία των αντίστοιχων ελληνικών (GREEK-BERT και GPT-2 Greek) ή πολυγλωσσικών (mT5) μοντέλων.

Για την αντιμετώπιση της έλλειψης ελληνικού συνόλου δεδομένων διαλόγου, χρησιμοποιήσαμε μηχανική μετάφραση (MT) για να δημιουργήσουμε ελληνική έκδοση του συνόλου δεδομένων DailyDialog. Διεξάγουμε 4 διαφορετικά πειράματα με τα πολυγλωσσικά μας μοντέλα mT5 και XGLM:

1. **Εγγενής εκπαίδευση**: Προσαρμόσαμε τα πολυγλωσσικά μοντέλα αποκλειστικά στο μεταφρασμένο σύνολο δεδομένων.

2. **Διαγλωσσική μεταφορά μάθησης**: Προσαρμόσαμε τα μοντέλα στην αρχική αγγλική έκδοση του DailyDialog και στη συνέχεια σε περιορισμένο αριθμό χειροκίνητα μεταφρασμένων ελληνικών παραδειγμάτων.

3. **Μάθηση πολλαπλών εργασιών**: Εκπαιδεύσαμε τα μοντέλα ταυτόχρονα στη γλώσσα προέλευσης και στη γλώσσα στόχο.

4. **Μάθηση με προτροπές**: Ενισχύσαμε τις προηγούμενες προσεγγίσεις με συγκεκριμένα prompts που μοιράζονται μεταξύ των γλωσσών.

Αξιολογήσαμε όλα τα μοντέλα χρησιμοποιώντας πολλαπλές μετρικές: Perplexity, BLEU, BertScore και Distinct-n. Τα αποτελέσματα δείχνουν ότι η εγγενής εκπαίδευση επέτυχε τη καλύτερη επίδοση, με το GPT2-Greek να αναδεικνύεται ως το καλύτερο μοντέλο (perplexity: 12.47, BLEU B-1: 25.93, Distinct-1: 23.13%, BertScore F-1: 71.37%). Μεταξύ των πολυγλωσσικών προσεγγίσεων, η εκπαίδευση βασισμένη σε prompts ενίσχυσε σημαντικά την απόδοση του XGLM (F-1: 69.12%), ενώ η πολυεργασιακή μάθηση αποδείχθηκε καλύτερη από τη διαγλωσσική μεταφορά μάθησης.

Διεξήγαμε επίσης ανθρώπινη αξιολόγηση για ποιοτικές πτυχές που οι αυτοματοποιημένες μετρικές ενδέχεται να μην αποτυπώνουν πλήρως. Αυτές οι αξιολογήσεις αποκάλυψαν ότι το XGLM που εκπαιδεύτηκε με πολυεργασιακή μάθηση βασισμένη σε prompts επέτυχε την καλύτερη απόδοση μεταξύ των προσεγγίσεων μας, κατατασσόμενο δεύτερο μόνο μετά το πολύ μεγαλύτερο μοντέλο Meltemi. Αυτό αποδεικνύει αποτελεσματική διαγλωσσική μεταφορά γνώσης παρά τη χρήση σημαντικά λιγότερων ελληνικών δεδομένων εκπαίδευσης.

Με αυτή τη διπλωματική εργασία, επιθυμούμε να ανοίξουμε νέους δρόμους για την εξερεύνηση της παραγωγής διαλόγων ανοιχτού τομέα για γλώσσες περιορισμένων πόρων και προτείνουμε ενδιαφέρουσες μελλοντικές επεκτάσεις για περαιτέρω έρευνα.

Λέξεις Κλειδιά - παραγωγή διαλόγων ανοιχτού πεδίου, γλώσσες περιορισμένων πόρων, Ελληνικά, transformers, BERT, GPT-2, mT5, XGLM, διαγλωσσική μεταφορά μάθησης, μάθηση πολλαπλών εργασιών, μάθηση με προτροπές, ανθρώπινη αξιολόγηση

Abstract

Human-machine conversation has been a critical and challenging task in AI and NLP. Recent years have seen rapid progress in open-domain dialogue generation. However, because vast conversational data are only available in English, the development of generation-based chatbots for non-English languages has lagged behind. This Diploma Thesis studies dialogue generation in a low-resource language, specifically Greek, where training data and pre-trained language models are limited.

We present theoretical background on machine learning (ML), deep learning (DL), and natural language processing (NLP), then study the transformer-based models used: BERT, GPT-2, T5 and XGLM. We analyze how the corresponding Greek (GREEK-BERT and GPT-2 Greek) or multilingual (mT5) models were created.

Following analysis of research on low-resource dialogue generation, we conduct experiments and discuss results. To address the lack of a Greek dialogue dataset, we used machine translation (MT) to create a Greek version of the DailyDialog dataset. We fine-tune Greek monolingual models (GREEK-BERT and GPT-2 Greek) on the translated dataset, then conduct 4 experiments with multilingual models mT5 and XGLM:

1. **Native training**: Fine-tuned multilingual models exclusively on the translated dataset for direct comparison with monolingual models.

2. **Cross-Lingual transfer learning**: Fine-tuned models using the original English DailyDialog dataset, then further fine-tuned on limited manually translated Greek examples.

3. **Multitask Learning**: Trained models simultaneously on both languages, utilizing the complete English dataset alongside a subset of the translated Greek dataset.

4. **Prompt based Learning**: Enhanced both approaches with specific prompt templates shared across languages to facilitate knowledge transfer from English to Greek dialogues.

We evaluated all models using multiple metrics: Perplexity, BLEU, and BertScore for response quality, and Distinct-n for lexical diversity. Results demonstrate that native training achieved superior performance, with GPT2-Greek as the best-performing model (perplexity: 12.47, BLEU B-1: 25.93, Distinct-1: 23.13%, BertScore F-1: 71.37%). Among multilingual approaches, prompt-based training significantly enhanced XGLM performance (F-1: 69.12%), while multitask learning consistently outperformed cross-lingual transfer learning.

Human evaluations assessed qualitative aspects that automated metrics might not capture. These revealed that our XGLM model trained using prompt-based multitask learning achieved the best performance among our approaches, ranking second only to the much larger Meltemi model trained on substantially more Greek data. This demonstrates effective cross-linguistic knowledge transfer despite using considerably less Greek training data.

This thesis opens new avenues for exploring open-domain dialogue generation for low-resource languages and proposes future extensions for further research.

Keywords - open-domain dialogue generation, low-resource languages, Greek language, transformers, BERT, GPT-2, mT5, XGLM, cross-lingual transfer learning, multitask learning, prompt learning, human evaluation

Ευχαριστίες

Η παρούσα διπλωματική αποτελεί κατόρθωμα προσωπικό και συλλογικό καθώς η ολοκλήρωση της παρούσας δεν θα μπορούσε να είχε επιχτευχθεί χωρίς την συνδρομή πολλαπλών προσώπων.

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον υπεύθυνο καθηγητή Αλέξανδρο Ποταμιάνο και στον κύριο Αθανάσιο Κατσαμάνη από το ερευνητικό κέντρο Αθηνά για την πολύτιμη καθοδήγηση και τους απαραίτητους πόρους που μου παρείχαν κατά την εκπόνηση αυτής της εργασίας.

Ακόμη, θα ήθελα να ευχαριστήσω τους φίλους μου, τους εξ αποστάσεως και δια ζώσης που σταθερά με υποστήριζαν σε όλες μου τις σπουδές. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, Νικόλαο και Βασιλική, και την αδερφή μου Γεωργία, που χωρίς την έμπνευση αλλά και τη στήριξη που μου έδωσαν δεν θα μπορούσα να είχα καταφέρει να ξεκινήσω αλλα και να ολοκληρώσω αυτές τις σπουδές.

Ανδρέας Κούκουνας,
Αθήνα, Ιούλιος 2025

Contents

List of Figures

List of Tables

Chapter 0

Εκτεταμένη Ελληνική Περίληψη

0.1 Εισαγωγή

Η ταχεία πρόοδος των τεχνολογιών επεξεργασίας φυσικής γλώσσας (NLP) έχει φέρει επανάσταση στη διάδραση ανθρώπου-υπολογιστή. Εικονικοί βοηθοί και chatbots εξυπηρέτησης πελατών έχουν γίνει αναπόσπαστα μέρη της καθημερινότητάς μας, βοηθώντας σε πολλαπλούς τομείς και παρέχοντας ψυχαγωγία.

Οι εξελίξεις στη Μηχανική Μάθηση (ML), ιδιαίτερα μέσω της Βαθιάς Μάθησης (DL), έχουν προωθήσει την ανάπτυξη διαφόρων συστημάτων διαλόγου. Τα chatbots ανοιχτού πεδίου αποτελούν σημαντικό πεδίο έρευνας, στοχεύοντας στη μίμηση ανθρώπινων συνομιλιών. Ιστορικά, μερικά από τα πρώτα chatbots ήταν η ELIZA και το PARRY. Η ELIZA μιμούνταν ψυχοθεραπευτή αναδιατυπώνοντας ερωτήσεις [79], ενώ το PARRY προσομοίωνε παρανοϊκό ασθενή [10]. Τα μοντέλα αυτά βασίζονταν σε χειροποίητους κανόνες, όχι σε μάθηση από δεδομένα.

Νεότερες μέθοδοι αξιοποιούν προσεγγίσεις βασισμένες σε δεδομένα, μαθαίνοντας από μεγάλο όγκο ανθρώπινων συνομιλιών. Τα μοντέλα ανοιχτού πεδίου εκπαιδεύονται σε εκτεταμένα σύνολα δεδομένων και βελτιώνονται μέσω προ-εκπαιδευμένων μοντέλων γλώσσας [91], [63]. Παρά τους πόρους για την ανάπτυξη αυτών των μοντέλων [89], [37], [62], οι περισσότεροι είναι κυρίως στα Αγγλικά, παρουσιάζοντας προκλήσεις για άλλες γλώσσες.

Η Ελληνική, με την πλούσια ιστορική και πολιτιστική κληρονομιά, θεωρείται γλώσσα χαμηλών πόρων στο πλαίσιο του NLP λόγω της έλλειψης επισημειωμένων συνόλων δεδομένων. Αυτή η διατριβή διερευνά καινοτόμες προσεγγίσεις για την αποτελεσματική χρήση των διαθέσιμων πόρων για την ανάπτυξη ικανών συστημάτων παραγωγής διαλόγου στα Ελληνικά.

Σε αυτή τη διατριβή, διερευνούμε διάφορες αρχιτεκτονικές για συστήματα διαλόγου, συμπεριλαμβανομένων μοντέλων κωδικοποιητή-αποκωδικοποιητή encoder-decoder και μόνο αποκωδικοποιητή decoder-only. Εξετάζουμε διαφορετικούς τύπους μοντέλων—μονόγλωσσα για τα Ελληνικά και πολύγλωσσα. Εφαρμόζουμε επίσης διάφορες τεχνικές εκπαίδευσης, όπως διαγλωσσική μεταφορά μάθησης cross-lingual transfer learning, μάθηση πολλαπλών εργαρσιών multitask learning και μάθηση μέσω prompt prompt-based learning, επιτρέποντας τη χρήση αγγλικών συνόλων δεδομένων για την ενίσχυση των περιορισμένων ελληνικών δεδομένων. Αυτή η έρευνα στοχεύει στη γεφύρωση του χάσματος στη γλωσσική τεχνολογία για τα Ελληνικά, παρέχοντας γνώσεις που θα μπορούσαν να προωθήσουν συστήματα διαλόγου για παρόμοιες γλώσσες χαμηλών πόρων.

0.2 Μηχανική Μάθηση

Σε αυτή την υποενότητα θα εξετάσουμε τα γλωσσικά μοντέλα, καθώς και βασικές τεχνικές εκπαίδευσης αυτών, όπως μεταφορά μάθησης (Transfer Learning), εκμάθηση πολλαπλών εργασιών (Multitask Learning), και μάθηση βάσει prompt.

0.2.1 Γλωσσικά Μονέλα

Τα γλωσσικά μοντέλα είναι μοντέλα που αναθέτουν πιθανότητες σε ακολουθίες λέξεων. Αποτελούν το θεμέλιο της Επεξεργασίας Φυσικής Γλώσσας, καθώς προσφέρουν μια μέθοδο μετατροπής ποιοτικών πληροφοριών κειμένου σε κατανοητά από τη μηχανή ποσοτικά δεδομένα. Πιο συγκεκριμένα, ένα γλωσσικό μοντέλο δεδομένης μιας ακολουθίας λέξεων ως είσοδο, προσπαθεί να προβλέψει την επόμενη λέξη. Με αυτόν τον τρόπο λοιπόν, το μοντέλο μπορεί να δημιουργήσει αναπαραστάσεις λέξεων με βάση το περιεχόμενο/ιστορικό. Με την πρόοδο της βαθιάς μάθησης, τα παραδοσιακά γλωσσικά μοντέλα που βασίζονται στη στατιστική αντικαταστάθηκαν από γλωσσικά μοντέλα που χρησιμοποιούν νευρωνικά δίκτυα, τα οποία τελικά οδήγησαν στα σημερινά μεγάλα προεκπαιδευμένα γλωσσικά μοντέλα, όπως το Bert [14] και το GPT-3 [8]. Τα προ-εκπαιδευμένα γλωσσικά μοντέλα αξιοποιούν τις τεράστιες ποσότητες δεδομένων κειμένου χωρίς ετικέτες για να εκπαιδευτούν, μέσω μη επιβλεπόμενης ή αυτο-επιβλεπόμενης μάθησης, προκειμένου να αποκτήσουν μια γενική κατανόηση της φυσικής γλώσσας. Μετά την εκπαίδευσή τους, μπορούν να προσαρμοστούν σε επιμέρους προβλήματα περνώντας από μερικούς ακόμη γύρους εκπαίδευσης, με εφαρμογή της μεθόδου βελτιστοποίησης (fine-tuning), χρησιμοποιώντας μικρότερα σύνολα δεδομένων με ετικέτες.

0.2.2 Μεταφορά Μάθησης

Η μεταφορά μάθησης (Transfer learning) είναι μια τεχνική μηχανικής μάθησης που περιλαμβάνει την αξιοποίηση γνώσης που αποκτήθηκε από την επίλυση ενός προβλήματος και την εφαρμογή της σε ένα διαφορετικό, συνήθως συναφές. Μια εικονογραφημένη έκδοση της τεχνικής μεταφοράς μάθησης φαίνεται στο Σχήμα 2.11. Σε πολλά προβλήματα βαθιάς μάθησης, για την κατασκευή ενός μοντέλου που επιλύει μια σύνθετη εργασία, απαιτείται τεράστια ποσότητα επισημειωμένων δεδομένων. Ωστόσο, η συλλογή επαρκών δεδομένων εκπαίδευσης μπορεί να είναι δαπανηρή, χρονοβόρα ή ακόμη και ανέφικτη σε πολλές περιπτώσεις. Αντί να ξεκινά η διαδικασία μάθησης από το μηδέν για μια νέα εργασία, η μεταφορά μάθησης εκμεταλλεύεται προϋπάρχουσα γνώση ή μοντέλα που έχουν εκπαιδευτεί σε μεγάλα σύνολα δεδομένων. [99]

Η μεταφορά μάθησης συνήθως περιλαμβάνει δύο στάδια: προ-εκπαίδευση και βελτιστοποίηση. Στο στάδιο προ-εκπαίδευσης, ένα μοντέλο εκπαιδεύεται σε μεγάλο όγκο δεδομένων. Αυτή η αρχική εκπαίδευση βοηθά το μοντέλο να μάθει γενικά χαρακτηριστικά και μοτίβα που μπορούν να φανούν χρήσιμα για πολλές διαφορετικές εργασίες. Στο στάδιο βελτιστοποίησης, το προεκπαιδευμένο μοντέλο εκπαιδεύεται περαιτέρω σε ένα μικρότερο σύνολο δεδομένων που είναι ειδικό για την εργασία στόχου. Η βελτιστοποίηση περιλαμβάνει την εκπαίδευση του μοντέλου στο νέο σύνολο δεδομένων διατηρώντας τα αρχικά βάρη σταθερά ή τροποποιώντας τα σύμφωνα με τη νέα εργασία. Ωστόσο, αξίζει να σημειωθεί ότι η μεταφερόμενη γνώση δεν ευνοεί πάντα τη νέα εργασία, καθώς μπορεί να αποτύχει αν υπάρχουν λίγα κοινά στοιχεία μεταξύ του τομέα προέλευσης και του τομέα στόχου.

Σε ειδικές περιπτώσεις, η αρχική εργασία περιλαμβάνει μη επιβλεπόμενη μάθηση. Αυτό είναι πολύ συνηθισμένο στην Επεξεργασία Φυσικής Γλώσσας, όπου προ-εκπαιδεύουμε μεγάλα μοντέλα γλώσσας με μεγάλες ποσότητες μη επισημειωμένων δεδομένων. Αυτά τα μοντέλα μαθαίνουν να προβλέπουν την επόμενη λέξη σε μια πρόταση και καταφέρνουν να συλλάβουν πλούσιες πληροφορίες περιεχομένου. Αργότερα, αυτά τα μοντέλα γλώσσας μπορούν να βελτιστοποιηθούν σε εργασίες όπως η μηχανική μετάφραση, η δημιουργία διαλόγων και η απάντηση ερωτήσεων.

0.2.3 Εκμάθηση Πολλαπλών Εργασιών

Η εκμάθηση πολλαπλών εργασιών (Multi-task learning) είναι μια προσέγγιση μηχανικής μάθησης που περιλαμβάνει την από κοινού εκπαίδευση ενός μοντέλου σε πολλαπλές συναφείς εργασίες. Αντί να εκπαιδεύονται διαφορετικά μοντέλα για κάθε εργασία, η πολλαπλή μάθηση εργασιών χρησιμοποιεί κοινές αναπαραστάσεις και γνώσεις μεταξύ των εργασιών με στόχο τη βελτίωση της απόδοσης σε κάθε μεμονωμένη εργασία. Η βασική ιδέα πίσω από την εκμάθηση πολλαπλών εργασιών είναι ότι η μάθηση από πολλαπλές εργασίες ταυτόχρονα μπορεί να παρέχει οφέλη σε κάθε μεμονωμένη εργασία.

Υπάρχουν διάφορες προσεγγίσεις για την υλοποίηση της εκμάθηση πολλαπλών εργασιών. Μια προσέγγιση είναι να μοιράζονται τα χαμηλότερου επιπέδου επίπεδα του μοντέλου μεταξύ των εργασιών, διατηρώντας παράλληλα ειδικά για κάθε εργασία επίπεδα στην κορυφή. Αυτή η στρατηγική επιτρέπει στο μοντέλο να μάθει τόσο κοινές αναπαραστάσεις όσο και χαρακτηριστικά ειδικά για κάθε εργασία.

Εναλλακτικά, μπορεί να χρησιμοποιηθεί ένα μοναδικό κοινό μοντέλο, χρησιμοποιώντας πολλαπλές κεφαλές εξόδου όπου κάθε κεφαλή προβλέπει την αντίστοιχη εργασία. Κατά την εκπαίδευση, τα κοινά επίπεδα και οι κεφαλές ειδικές για κάθε εργασία βελτιστοποιούνται από κοινού. Τα κοινά επίπεδα ενημερώνονται με βάση τις παραγώγους από όλες τις εργασίες, επιτρέποντας στο μοντέλο να συλλάβει τα κοινά χαρακτηριστικά μεταξύ των εργασιών. Οι κεφαλές ειδικές για κάθε εργασία ενημερώνονται με βάση τις παραγώγους που είναι ειδικές για κάθε εργασία, επιτρέποντάς τους να εξειδικευτούν στην ακριβή πρόβλεψη για τις αντίστοιχες εργασίες τους.

Αυτή η προσέγγιση μειώνει τη συνολική πολυπλοκότητα του μοντέλου και τις απαιτήσεις μνήμης σε σύγκριση με την εκπαίδευση ξεχωριστών μοντέλων για κάθε εργασία και συχνά οδηγεί σε καλύτερη γενίκευση, καθώς το μοντέλο πρέπει να βρει μια κοινή αναπαράσταση που βελτιώνει την απόδοση σε όλες τις μεμονωμένες εργασίες. Μια απεικόνιση αυτής της προσέγγισης πολλαπλής μάθησης εργασιών φαίνεται στο Σχήμα 2.12.

0.2.4 Μάθηση βάσει prompt

Η μάθηση βασισμένη σε prompt χρησιμοποιείται συχνά ως μια ελαφρύτερη εναλλακτική σε σύγκριση με τη βελτιστοποίηση. Ωστόσο, τα prompts μπορούν να χρησιμοποιηθούν παράλληλα με τη βελτιστοποίηση για την ενίσχυση της απόδοσης. Στις ακόλουθες παραγράφους, παρουσιάζουμε αυτές τις 2 εναλλακτικές.

Εναλλακτική της βελτιστοποίησης

Όπως αναφέρθηκε προηγουμένως, στη μάθηση βασισμένη σε prompts, οι παράμετροι του προ-εκπαιδευμένου μοντέλου συνήθως διατηρούνται παγωμένες, ειδικά για μεγάλα προεκπαιδευμένα μοντέλα γλώσσας, και μόνο οι παράμετροι του prompt θ_p αλλάζουν. Εκπαιδεύοντας αποκλειστικά τις παραμέτρους του prompt, η μάθηση βασισμένη σε prompt προσφέρει μια πιο αποδοτική εναλλακτική λύση από τη βελτιστοποίηση όσον αφορά τους υπολογιστικούς πόρους και τις απαιτήσεις αποθήκευσης.

Η μάθηση βασισμένη σε prompt αποδεικνύεται ιδιαίτερα πλεονεκτική σε σενάρια όπου τα διαθέσιμα δεδομένα για μια συγκεκριμένη εργασία είναι περιορισμένα. Αυτό συμβαίνει επειδή οι παράμετροι του προ-εκπαιδευμένου μοντέλου παραμένουν αμετάβλητες, διατηρώντας τις ικανότητες κατανόησης γλώσσας που αποκτήθηκαν κατά τη φάση προ-εκπαίδευσης. Κατά συνέπεια, η συγκεκριμένη διαδικασία μάθησης καθοδηγεί αποκλειστικά το μοντέλο προς τη συγκεκριμένη εργασία χωρίς να επηρεάζει τις υποκείμενες ικανότητες κατανόησης και παραγωγής του.

Συμπληρωματική της βελτιστοποίησης

Ενώ είναι συνηθισμένη πρακτική να διατηρούνται οι παράμετροι του προ-εκπαιδευμένου μοντέλου γλώσσας παγωμένες, ειδικά όταν πρόκειται για εξαιρετικά μεγάλα προ-εκπαιδευμένα μοντέλα, αυτό δεν συμβαίνει πάντα. Ορισμένοι ερευνητές χρησιμοποιούν prompts ως συμπληρωματικές πληροφορίες για τη βελτίωση της απόδοσης, ενώ παράλληλα βελτιστοποιούν μερικές ή όλες τις παραμέτρους του μοντέλου [45], [5]. Αυτό είναι ιδιαίτερα συνηθισμένο όταν εργαζόμαστε με μικρότερα μοντέλα, επειδή τότε η βελτιστοποίηση απαιτεί λιγότερους πόρους και χώρο. Η απόφαση για πάγωμα ή προσαρμογή συγκεκριμένων παραμέτρων καθορίζεται τελικά από μια σειρά κριτηρίων, συμπεριλαμβανομένης της συγκεκριμένης εργασίας, της διαθεσιμότητας δεδομένων και των διαθέσιμων πόρων.

0.2.5 Προεκπαιδευμένα γλωσσικά μοντέλα

BERT

Το BERT μοντέλο βασίζεται αρχιτεκτονικά στην ιδέα του Transformer [14]. Αποτελείται από μια στοίβα κωδικοποιητών χρησιμοποιώντας την δυνατότητα αυτοπροσοχής πολλών κεφαλών (multi-head self-attention) σε δύο κατευθύνσεις. Ένας από τους κύριους λόγους για την καλή απόδοση του BERT σε διαφορετικές εργασίες (tasks) είναι η προεκπαίδευσή του σε δύο μη εποπτευόμενες εργασίες. Με αυτόν τον τρόπο, το μοντέλο έχει τη δυνατότητα να κατανοεί τα μοτίβα της γλώσσας. Η πρώτη εργασία στην οποία το μοντέλο είναι προεκπαιδευμένο ονομάζεται ¨μοντελοποίηση γλώσσας με κενά' (masked language modeling - MLM) [14]. Σε αυτήν την εργασία, το 15% των λέξεων κάθε ακολουθίας καλύπτεται τυχαία και το μοντέλο προσπαθεί να προβλέψει τις λέξεις αυτές. Η δεύτερη εργασία ονομάζεται ¨πρόβλεψη επόμενης πρότασης' (Next Sentence Prediction - NSP), όπου το μοντέλο, δεδομένης μιας πρότασης, προσπαθεί να βρει την ακόλουθή της.

Το GREEK-BERT [30] προσαρμόζει το BERT-base για την επεξεργασία της ελληνικής γλώσσας. Προ-εκπαιδεύτηκε σε 29 GB ελληνικού κειμένου από τη Wikipedia, τα Πρακτικά του Ευρωπαϊκού Κοινοβουλίου και το OSCAR (καθαρή έκδοση του Common Crawl). Βελτιστοποιημένο για αναγνώριση μερών του λόγου, αναγνώριση οντοτήτων και συμπερασμό φυσικής γλώσσας, το GREEK-BERT ξεπέρασε τα πολύγλωσσα μοντέλα (mBERT, XLM-R [11]) στις εργασίες αναγνώριση οντοτήτων (Named Entity Recognition) και εξαγωγής συμπερασμάτων φυσικής γλώσσας (Natural Language Inference), ενώ πέτυχε συγκρίσιμα αποτελέσματα στην αναγνώριση μερών του λόγου (POS tagging).

GPT-2

Το GPT-2 [60], που αναπτύχθηκε από την OpenAI, είναι ένα ευρέως αναγνωρισμένο μοντέλο γλώσσας που χρησιμοποιείται σε διάφορες εργασίες με αποτελέσματα αιχμής. Η αρχιτεκτονική του, χρησιμοποιεί τη δομή του αποκωδικοποιητή της αρχιτεκτονικής των transformer, λειτουργώντας ως αυτο-παλινδρομικό μοντέλο που παράγει ένα σύμβολο κάθε φορά, προσθέτοντάς το στην ακολουθία εισόδου για την επόμενη πρόβλεψη. Προ-εκπαιδεύτηκε στο WebText, που περιλαμβάνει πάνω από 8 εκατομμύρια έγγραφα (40 GB κειμένου), για να προβλέψει την επόμενη λέξη σε μια πρόταση δεδομένου του προηγούμενου πλαισίου. Η βασική διαφορά μεταξύ του BERT και του GPT-2 είναι η χρήση από το GPT-2 της αυτο-προσοχής πολλαπλών κεφαλών με μάσκα (masked multi-head self-attention), η οποία εμποδίζει την πληροφορία από λέξεις στα δεξιά της θέσης που υπολογίζεται. Κάθε κεφαλή προσοχής επικεντρώνεται σε διαφορετικές πτυχές του κειμένου εισόδου (σύνταξη, σημασιολογία), επιτρέποντας πιο ακριβείς προβλέψεις και καλύτερη κατανόηση του πλαισίου.

Το GPT-2 Greek [38] προσαρμόζει το αρχικό μικρό μοντέλο GPT-2 για την ελληνική γλώσσα μέσω βελτιστοποίησης με σταδιακό ξεκλείδωμα επιπέδων. Αυτή η αποτελεσματική προσέγγιση για γλώσσες με περιορισμένους πόρους βελτιστοποιεί την απόδοση χωρίς εκπαίδευση από την αρχή. Το μοντέλο δημιουργήθηκε χρησιμοποιώντας δείγμα 23,4 GB από ελληνικά σώματα κειμένων συμπεριλαμβανομένων των CC100, Wikimatrix, Tatoeba, Books, SETIMES και GlobalVoices. Η διαθεσιμότητά του ανοίγει νέες ευκαιρίες για εφαρμογές επεξεργασίας φυσικής γλώσσας στην ελληνική γλώσσα.

Text-To-Text Transfer Transformer (T5)

Το μοντέλο Text-to-Text Transfer Transformer, (T5)) παρουσιάστηκε το 2019 από ερευνητές της Google [61], αντιπροσωπεύοντας μια σημαντική πρόοδο στη μεταφορά μάθησης για εργασίες επεξεργασίας φυσικής γλώσσας. Η βασική καινοτομία του T5 είναι η προσέγγιση κάθε προβλήματος επεξεργασίας κειμένου ως πρόβλημα μετατροπής κειμένου σε κείμενο, επιτρέποντας την εφαρμογή του ίδιου μοντέλου, στόχου, διαδικασίας εκπαίδευσης και διαδικασίας αποκωδικοποίησης σε διάφορες εργασίες επεξεργασίας φυσικής γλώσσας., και είναι βασισμένο στο πλαίσιο κωδικοποιητή-αποκωδικοποιητή της αρχικής αρχιτεκτονικής του transformer.

Για την προ-εκπαίδευση, οι ερευνητές ανέπτυξαν το Κολοσσιαίο Καθαρό Σώμα Κειμένων Διαδικτύου (Colossal Clean Crawled Corpus, C4), μια καθαρισμένη έκδοση του Common Crawl διπλάσιου μεγέθους από τη Wikipedia. Το T5 χρησιμοποιεί μια τροποποίηση της μοντελοποίησης γλώσσας με μάσκα που ονομάζεται ¨εύρος διαφθοράς' (corruption span), αντικαθιστώντας διαδοχικά διαστήματα λέξεων με ένα μόνο κενό, αντί να καλύπτει μεμονωμένες λέξεις όπως στο BERT. Οι ερευνητές δοκίμασαν τρεις στρατηγικές: κάλυψη τυχαίων λέξεων, κάλυψη διαδοχικών λέξεων και απόρριψη λέξεων, με τη τεχνική κάλυψης διαδοχικών λέξεων να αποδεικνύεται πιο αποτελεσματική.

Μια βασική καινοτομία στην προ-εκπαίδευση του T5 ήταν η χρήση προθέματος κειμένου για συγκεκριμένες εργασίες (π.χ., ¨μετάφραση από Αγγλικά σε Γερμανικά:¨) για να βοηθήσει το μοντέλο να προσαρμοστεί σε συγκεκριμένες εργασίες. Αυτή η προσέγγιση μάθησης πολλαπλών εργασιών περιόρισε το πεδίο παραγωγής, βελτιώνοντας την απόδοση και την εξειδίκευση εργασιών.

Το (Multilingual Text-To-Text Transfer Transformer, mT5) [85] ακολουθεί την αρχιτεκτονική του T5 αλλά με διαφορετική διαδικασία προ-εκπαίδευσης. Διαθέσιμο σε πέντε παραλλαγές με 300 εκατομμύρια έως 13 δισεκατομμύρια παραμέτρους, το mT5 προ-εκπαιδεύτηκε αποκλειστικά στο mC4 χωρίς εποπτευόμενη εκπαίδευση, απαιτώντας βελτιστοποίηση πριν τη χρήση σε μεταγενέστερες εργασίες. Σε αντίθεση με το T5, τα προθέματα εργασιών στο mT5 απαιτούνται μόνο για βελτιστοποίηση πολλαπλών εργασιών. Το mT5 έχει επιτύχει κορυφαία απόδοση σε διάφορα πολύγλωσσα εργασίες αναφορικά με την επεξεργασίας φυσικής γλώσσας, καθιστώντας το πολύτιμο για την πολύγλωσση κατανόηση και παραγωγή κειμένου.

XGLM

Τα αυτοπαλινδρομικά μοντέλα γλώσσας μεγάλης κλίμακας όπως το GPT-3 μπορούν να προσαρμοστούν σε διάφορες εργασίες μέσω μάθησης λίγων και μηδενικών παραδειγμάτων (few- and zero-shot learning) με μικρότερο κόστος από την πλήρη βελτιστοποίηση. Ωστόσο, τα δεδομένα εκπαίδευσής τους που κυριαρχούνται από τα αγγλικά ενδεχομένως περιορίζουν τη διαγλωσσική γενίκευση. Για την αντιμετώπιση αυτού του περιορισμού, η Meta AI παρουσίασε το μοντέλο XGLM [39], ένα πολύγλωσσο αυτοπαλινδρομικό μοντέλο γλώσσας εμπνευσμένο από το GPT-3 [8].

Το XGLM εκπαιδεύτηκε σε ένα ισορροπημένο σώμα κειμένων που καλύπτει διάφορες γλώσσες για να διερευνήσει τις πολύγλωσσες δυνατότητες μάθησης λίγων και μηδενικών παραδειγμάτων σε διάφορες εργασίες. Ως ένα από τα πρώτα πολύγλωσσα αυτοπαλινδρομικά μοντέλα, η αρχιτεκτονική του XGLM μοιάζει με το GPT-3 και το GPT-2.

Για τα δεδομένα προ-εκπαίδευσης, οι ερευνητές επέκτειναν την διαδικασία εξόρυξης του σώματος κειμένων CC100 για να δημιουργήσουν το CC100-XL, ένα πολύγλωσσο σύνολο δεδομένων που καλύπτει 68 μηνιαία στιγμιότυπα του Common Crawl. Τα δεδομένα προ-εκπαίδευσης περιλαμβάνουν 30 γλώσσες που καλύπτουν 16 γλωσσικές οικογένειες.

Οι συγγραφείς δοκίμασαν το XGLM σε διάφορες μεταγενέστερες εργασίες, κυρίως χρησιμοποιώντας μάθηση λίγων παραδειγμάτων. Σε αυτά τα σενάρια χαμηλών πόρων, η απόδοση του μοντέλου εξαρτάται σε μεγάλο βαθμό από την κατασκευή των prompts—μια πρόκληση που περιπλέκεται περαιτέρω σε πολύγλωσσα περιβάλλοντα όπου είναι απαραίτητη η εύρεση βέλτιστων prompts για διαφορετικές γλώσσες. Διερευνήθηκαν τρεις προσεγγίσεις για μη αγγλικά prompts: δημιουργία από φυσικούς ομιλητές, μετάφραση από αγγλικά prompts και διαγλωσσικά prompts (εφαρμογή αγγλικών prompts απευθείας σε μη αγγλικά παραδείγματα). Ενώ τα γνήσια και μεταφρασμένα prompts πέτυχαν καλύτερα αποτελέσματα, τα διαγλωσσικά prompts έδειξαν ανταγωνιστική απόδοση, ιδιαίτερα για γλώσσες χαμηλών πόρων.

Το XGLM πέτυχε εντυπωσιακά αποτελέσματα σ μάθηση λίγων παραδειγμάτων σε πάνω από 20 γλώσσες (συμπεριλαμβανομένων γλωσσών μεσαίων και χαμηλών πόρων) σε εργασίες συμπερασμού φυσικής γλώσσας και μηχανικής μετάφρασης. Ένας βασικός παράγοντας στην απόδοση του XGLM ήταν οι ισχυρές διαγλωσσικές του ικανότητες, επιδεικνύοντας βαθιά κατανόηση πολλαπλών γλωσσών και αποτελεσματική μεταφορά γνώσης μεταξύ τους.

0.3 Παραγωγή Διαλόγου για γλώσσες με περιορισμένους πόρους

Η φυσική γλώσσα κατέχει τεράστια σημασία στον ανθρώπινο πολιτισμό, καθώς εξελίχθηκε για να διευκολύνει τη συνύπαρξη και την επικοινωνία. Ο διάλογος, ως σημαντικό μέρος της γλώσσας, συνδέει τους ανθρώπους μέσω συνομιλιών. Στον τομέα της τεχνητής νοημοσύνης, τα συστήματα διαλόγου έχουν αναδειχθεί ως ένα απαιτητικό πεδίο που επιτρέπει την επικοινωνία μεταξύ πρακτόρων συνομιλίας και ανθρώπων.

Οι πράκτορες συνομιλίας χωρίζονται σε δύο κύριες κατηγορίες: τους προσανατολισμένους σε εργασίες (task-oriented) και τους μη προσανατολισμένους σε εργασίες (non-task-oriented). Οι πρώτοι σχεδιάζονται για συγκεκριμένες εργασίες και εμπλέκονται σε σύντομες συνομιλίες, ενώ οι δεύτεροι επικεντρώνονται σε ελεύθερες συζητήσεις. Τα συστήματα διαλόγου ανοιχτού πεδίου (open-domain dialogue systems) έχουν την ικανότητα να κατανοούν εισόδους φυσικής γλώσσας και να παράγουν απαντήσεις που μοιάζουν με ανθρώπινες.

Τα πρώτα παραδείγματα συστημάτων συνομιλίας περιλαμβάνουν το ELIZA [79] και το PARRY [10], που βασίζονταν σε κανόνες και μοτίβα. Οι σύγχρονες μέθοδοι αξιοποιούν προσεγγίσεις βασισμένες σε δεδομένα, επιτρέποντας στα συστήματα να μαθαίνουν από τεράστιες ποσότητες συνομιλιών μεταξύ ανθρώπων.

0.3.1 Παραγωγή Διαλόγου

Τα μοντέλα παραγωγής διαλόγου βασίζονται συνήθως στο μοντέλο ακολουθία-προς-ακολουθία (sequence-to-sequence, seq2seq), μια αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή όπου και οι δύο μπορεί να είναι είτε επαναλαμβανόμενα νευρωνικά δίκτυα (RNNs) είτε Transformers με μπλοκ αυτοπροσοχής (self-attention blocks). Τελευταία, τα αυτοπαλίνδρομα μοντέλα, που χρησιμοποιούν μόνο τον αποκωδικοποιητή της transformer αρχιτεκτονικής έχουν βρει σημαντική επιτυχία ως η προτιμώμενη μέθοδος για την κατασκευή συστημάτων παραγωγής διαλόγου.

Η εισαγωγή της αρχιτεκτονικής Transformers επέφερε επανάσταση στον τομέα της επεξεργασίας φυσικής γλώσσας, βελτιώνοντας την ποιότητα των απαντήσεων και μειώνοντας το υπολογιστικό κόστος. Πολλά σύγχρονα συστήματα διαλόγου ανοιχτού πεδίου βασίζονται σε αυτή και προεκπαιδευμένα σύνολα δεδομένων. Σημαντικά μοντέλα όπως το Meena [1], το DialoGPT [91], το LaMDA [72] και το ChatGPT [55] έχουν σημειώσει σημαντική πρόοδο στην παραγωγή διαλόγου. Το μοντέλο Meltemi [77] αποτελεί ένα παράδειγμα προσαρμογής των μεγάλων γλωσσικών μοντέλων (LLMs) στα ελληνικά, βασιζόμενο στο μοντέλο Mistral-7B [25].

0.3.2 Τεχνικές για Παραγωγή Διαλόγου σε Γλώσσες με Περιορισμένους Πόρους

Τόσα τα σύνολα δεδομένων διαλόγου ανοιχτού πεδίου, όσο και τα προεκπαιδευμένα γλωσσικά μοντέλα σε γλώσσες εκτός της αγγλικής και της κινεζικής είναι σπάνια. Για την αντιμετώπιση αυτού του προβλήματος, έχουν αναπτυχθεί διάφορες τεχνικές.

Μετάφραση και Εγγενής Εκπαίδευση

Μια προσέγγιση περιλαμβάνει την εκπαίδευση μοντέλων σε μεταφράσεις γνωστών αγγλικών συνόλων δεδομένων. Στο [53], οι συγγραφείς πρότειναν έναν μετασχηματιστή κωδικοποιητή-αποκωδικοποιητή (BERT2BERT) αρχικοποιημένο με παραμέτρους του AraBERT [2]. Ωστόσο, η έρευνα στο [69] ανέδειξε τους περιορισμούς της μετάφρασης συνόλων δεδομένων από γλώσσες με πολλούς πόρους σε γλώσσες με λίγους πόρους, οι οποίες εντοπίζονται στη πολιτιστική ιδιαιτερότητα των διαλόγου, αφού οι άμεσες μεταφράσεις μπορεί να μην αποτυπώνουν τις αποχρώσεις της γλώσσας-στόχου, οδηγούν- σε αφύσικες γενιές διαλόγου.

Διαγλωσσική Μεταφορά Μάθησης

Η διαγλωσσική μεταφορά μάθησης ολίγων παραδειγμάτων (cross-lingual transfer learning) προσφέρει μια υποσχόμενη λύση και περιλαμβάνει δύο φάσεις: την εκπαίδευση πηγής (source-training) και την προσαρμογή στόχου (target-adapting). Η έρευνα από τους Ϊνιτ ετ αλ. [31] έδειξε ότι, παρά τις εντυπωσιακές δυνατότητες μεταφοράς, η απόδοση των πολυγλωσσικών μετασχηματιστών μειώνεται σημαντικά για γλώσσες που είναι γλωσσικά απομακρυσμένες ή έχουν μικρότερα σύνολα εκπαίδευσης.

Πολυεργασιακή Μάθηση

Η πολυεργασιακή μάθηση (multitask learning, MTL) είναι μια αποτελεσματική προσέγγιση επαγωγικής μεταφοράς που βελτιώνει τη γενίκευση μέσω της κοινής εκμάθησης μίας ή περισσότερων βοηθητικών εργασιών μαζί με την εργασία-στόχο. Στο πλαίσιο της παραγωγής διαλόγου σε γλώσσες με περιορισμενους πόρους, όπου η έλλειψη δεδομένων αποτελεί σημαντική πρόκληση, η πολυεργασιακή μάθηση προσφέρει την δυνατότητα για τον εμπλουτισμό των συστημάτων διαλόγου με δανεισμό γνώσης από συναφείς εργασίες. Οι Magooda et al. [49] διερεύνησαν τη πολυεργασιακή μάθηση για αφηρημένη περίληψη σε γλώσσες με λίγους πόρους και διαπίστωσαν ότι εργασίες όπως η ανίχνευση παραφράσεων και η ανίχνευση εννοιών θα μπορούσαν να βελτιώσουν την ποιότητα της περίληψης. Εφαρμόζοντας παρόμοιες αρχές στην εργασία της παραγωγής διαλόγου, όπου η παραγωγή παραφράσεων και η ανίχνευση σχετικών εννοιών είναι εξίσου ζωτικής σημασίας, θα μπορούσε ομοίως να βελτιώσει την απόδοση.

Μάθηση με prompts

Η μάθηση με prompts (prompt learning) αντιπροσωπεύει μια στρατηγική προσαρμογή των προεκπαιδευμένων γλωσσικών μοντέλων σε εξειδικευμένες εργασίες τροποποιώντας την είσοδο των μοντέλων με συγκεκριμένες τροποποιήσεις. Οι Madotto et al. [48] εισήγαγαν τη μάθηση ολίγων παραδειγμάτων με βάση τα prompts για συστήματα διαλόγου, αποδεικνύοντας ότι μπορεί να επιτευχθεί σημαντική απόδοση ενσωματώνοντας prompts συγκεκριμένων εργασιών στην είσοδο του μοντέλου.

0.3.3 Μετρικές Αξιολόγησης

Οι μετρικές αξιολόγησης για τα συστήματα παραγωγής διαλόγου χωρίζονται σε αυτόματες και μετρικές βασισμένες σε ανθρώπους.

Αυτόματες Μετρικές

- **Word Perplexity**: Υπολογίζει την πιθανότητα το μοντέλο να προβλέψει σωστά την επόμενη λέξη σε μια συνομιλία. Χαμηλότερο perplexity υποδηλώνει καλύτερο μοντέλο.

- **BLEU**: Βαθμολογεί μια απάντηση με βάση το πόσο καλά ταιριάζει με ακολουθίες λέξεων n-gram που βρίσκονται σε μια πρότυπη απάντηση.

- **SacreBLEU** [58]: Παρέχει μια τυποποιημένη μεθοδολογία για τον υπολογισμό του σκορ BLEU.

- **BERTScore** [90]: Αξιοποιεί τα contextual embeddings από το BERT και υπολογίζει την ομοιότητα μεταξύ των embeddings των λέξεων στα παραγόμενα και τα κείμενα αναφοράς.

- **Response Diversity**: Το Distinct-1 και Distinct-2 μετρούν τον αριθμό των διακριτών μονογραμμάτων και διγραμμάτων των παραγόμενων απαντήσεων [34].

Μετρικές Βασισμένες σε Ανθρώπους

- Συγκρίσεις κατά ζεύγη για να επιλέξουν οι άνθρωποι ποια από τις δύο απαντήσεις είναι πιο κατάλληλη, πιο αρμόζουσα και πιο χρήσιμη [68].

- Αξιολόγηση συνάφειας: Οι άνθρωποι βαθμολογούν τις παραγόμενες απαντήσεις ανάλογα με το αν φαίνονται σχετικές με τη συνομιλία και εντός θέματος [62].

- Αξιολόγηση ρευστότητας/συνοχής: Οι άνθρωποι βαθμολογούν τις απαντήσεις ανάλογα με το αν φαίνονται κατανοητές, λογικά και συντακτικά ορθές [59].

Συμπερασματικά, η αξιολόγηση των συστημάτων συνομιλίας ανοιχτού πεδίου απαιτεί μια ολοκληρωμένη προσέγγιση, συνδυάζοντας ποικιλία μετρικών. Η χρήση συνδυασμού μετρικών όπως το perplexity, το BLEU, το Distinct-N, το BERTScore, μαζί με το sacreBLEU για τυποποιημένες συγκρίσεις, προσφέρει πολύτιμες γνώσεις σε διάφορες πτυχές της ποιότητας συνομιλίας. Ωστόσο, οι περιορισμοί αυτών των μετρικών υπογραμμίζουν τη σημασία των αξιολογήσεων από ανθρώπους για την καταγραφή των λεπτών αποχρώσεων της φυσικής συνομιλίας.

0.4 Παραγωγή Διαλόγου: Ελληνικά

Η πρόοδος στην Τεχνητή Νοημοσύνη έχει αναζωπυρώσει το ενδιαφέρον για την ανάπτυξη μοντέλων διαλόγων ανοιχτού πεδίου. Αυτά τα μοντέλα συνήθως εκπαιδεύονται χρησιμοποιώντας μεγάλες ποσότητες δεδομένων συνομιλίας και επωφελούνται από προ-εκπαιδευμένα μοντέλα παραγωγής γλώσσας που μπορούν να προσαρμοστούν στην παραγωγή αποκρίσεων ανοιχτού πεδίου. Η ευρεία διαθεσιμότητα τέτοιων πόρων έχει συμβάλει στην ανάπτυξη υψηλής απόδοσης μοντέλων συνομιλίας [91], [63]. Παρά την ύπαρξη σημαντικών πόρων [89], [37], [62], οι περισσότεροι είναι στα Αγγλικά, καθιστώντας δύσκολη την παραγωγή παρόμοιων μοντέλων για άλλες γλώσσες.

Η πρόκληση των περιορισμένων πόρων έχει μελετηθεί για μοντέλα συνομιλίας προσανατολισμένα να διεκπαιρώνουν συγκεκριμένες εργασίες (task-oriented) [83], μηχανική μετάφραση [51], απάντηση ερωτήσεων [56], και άλλες εφαρμογές Επεξεργασίας Φυσικής Γλώσσας [21], [31]. Ωστόσο, ελάχιστες μελέτες έχουν στοχεύσει στο ζήτημα των περιορισμένων πόρων στα μοντέλα συνομιλίας ανοιχτού πεδίου.

Οι Yang et al. [87] μελέτησαν το πρόβλημα της παραγωγής αποκρίσεων με περιορισμένους πόρους χρησιμοποιώντας 360K ζεύγη εκφράσεων-αποκρίσεων στα Κινέζικα, προτείνοντας την εκτίμηση προτύπων από μη επισημειωμένα δείγματα. Οι Naous et al. [53] πέτυχαν υψηλή απόδοση στην παραγωγή αποκρίσεων στα Αραβικά, προσαρμόζοντας ένα μοντέλο transformer σε 36K δείγματα αυτόματα μεταφρασμένα από τα Αγγλικά [54]. Στην παρούσα εργασία αντιμετωπίζουμε το πρόβλημα της παραγωγής αποκρίσεων ανοιχτού πεδίου στα Ελληνικά.

0.4.1 Σύνολα Δεδομένων

Το σύνολο δεδομένων DailyDialog [37] αποτελεί μια δημοσίως διαθέσιμη συλλογή πολυστροφικών διαλόγων που καλύπτουν ποικίλα θέματα. Περιλαμβάνει πάνω από 13.000 συνομιλίες με μέσο όρο 7,9 προτάσεις ανά διάλογο και μέση έκταση 15 λέξεις ανά πρόταση. Οι συνομιλίες συλλέχθηκαν από ιστοσελίδες εξάσκησης Αγγλικών και αντικατοπτρίζουν καθημερινούς διαλόγους με σκοπό την ανταλλαγή πληροφοριών και την κοινωνική επαφή. Καλύπτουν διάφορα σενάρια όπως συζητήσεις για διακοπές, εξυπηρέτηση σε καταστήματα και εστιατόρια, με κυριότερες θεματικές τις Σχέσεις (33,33%), την Καθημερινή Ζωή (28,26%) και την Εργασία (14,49%).

Για τη δημιουργία της ελληνικής έκδοσης του συνόλου δεδομένων, χρησιμοποιήθηκε νευρωνική μηχανική μετάφραση. Συγκεκριμένα, το μοντέλο μετάφρασης από τη συλλογή OPUS [73] αποτέλεσε το κύριο εργαλείο μετάφρασης από τα Αγγλικά στα Ελληνικά, προσφέροντας υψηλή απόδοση στη συγκεκριμένη γλωσσική κατεύθυνση.

0.4.2 Προτεινόμενα Μοντέλα

Η μελέτη αξιολογεί τέσσερα διαφορετικά γεννητικά μοντέλα για την παραγωγή ελληνικών συνομιλιών. Το πρώτο είναι το GPT2-Greek, ένα μονόγλωσσο αυτοπαλινδρομικό μοντέλο με 117M παραμέτρους που αναπτύχθηκε προσαρμόζοντας την αγγλική έκδοση με σταδιακό ξεκλείδωμα επιπέδων. Το δεύτερο είναι το GREEK-BERT2GREEK-BERT, μια μονόγλωσση υλοποίηση seq2seq με 224M παραμέτρους όπου κωδικοποιητής και αποκωδικοποιητής αρχικοποιούνται με τα βάρη του GREEK-BERT. Το τρίτο είναι το mT5, ένα πολύγλωσσο μοντέλο που ακολουθεί την αρχιτεκτονική T5 με 300M παραμέτρους. Το τέταρτο είναι το XGLM, ένα πολύγλωσσο μοντέλο αποκωδικοποιητή με 564M παραμέτρους. Η σημαντική διαφορά στον αριθμό παραμέτρων οφείλεται στους περιορισμούς διαθεσιμότητας μοντέλων εκπαιδευμένων σε ελληνικά κείμενα.

0.4.3 Προσεγγίσεις εκπαίδευσης

Για τα πειράματά μας, εφαρμόσαμε τέσσερις κύριες προσεγγίσεις εκπαίδευσης, βασισμένες στις ιδέες προηγούμενων εργασιών στον τομέα αυτό, όπως περιγράφεται στην Ενότητα 0.3.2: native training, cross-lingual transfer learning, multitask learning, και prompt learning.

Native training

Ακολουθώντας τη μεθοδολογία που προτάθηκε από τους συγγραφείς στο [54], η αρχική μας προσέγγιση περιελάμβανε εκπαίδευση σε ένα ελληνικό σύνολο δεδομένων παραγωγής διαλόγου. Δεδομένης της απουσίας τέτοιων συνόλων δεδομένων, χρησιμοποιήσαμε τις ελληνικές μεταφράσεις του συνόλου δεδομένων Daily Dialog. Η εκπαίδευση διεξήχθη αποκλειστικά στα ελληνικά, επιτρέποντας και στα τέσσερα μοντέλα, τόσο τα μονόγλωσσα όσο και τα πολύγλωσσα, να εκπαιδευτούν χρησιμοποιώντας την ίδια συνάρτηση απώλειας μοντελοποίησης γλώσσας. Οι διαδικασίες εκπαίδευσης για τα μοντέλα decoder-only διέφεραν ελαφρώς από εκείνες για τα μοντέλα seq2seq.

Cross-lingual transfer learning

Όπως περιγράφεται στην Ενότητα 4.3.2, η διαγλωσσική μεταφορά μάθησης με λίγα παραδείγματα (cross-lingual transfer learning) είναι μια αποτελεσματική στρατηγική για τη μεταφορά γνώσης από μια γλώσσα πηγή, στην προκειμένη περίπτωση τα αγγλικά, σε μια γλώσσα στόχο, τα ελληνικά. Για να διευκολυνθεί αυτό, είναι απαραίτητες τόσο οι αρχικές όσο και οι μεταφρασμένες εκδόσεις του συνόλου δεδομένων DailyDialog. Κατά συνέπεια, καθώς αυτό το στάδιο περιλαμβάνει εκπαίδευση τόσο σε αγγλικά όσο και σε ελληνικά δεδομένα, μόνο τα μοντέλα 3 και 4, τα οποία είναι πολύγλωσσα, εκπαιδεύτηκαν υπό αυτές τις συνθήκες. Η τυπική προσέγγιση περιελάμβανε αρχικά τη ρύθμιση (fine-tuning) όλων των παραμέτρων του μοντέλου στο αγγλικό σύνολο δεδομένων για να επιτρέψει στο μοντέλο να μάθει την παραγωγή διαλόγου με δεδομένα υψηλής ποιότητας. Στη συνέχεια, το μοντέλο εκπαιδεύτηκε ξανά με τον ίδιο τρόπο, χρησιμοποιώντας διάφορα υποσύνολα του ελληνικού συνόλου δεδομένων, με $k = 32, 64, 128, 512, 1024$ παραδείγματα. Αυτή η μέθοδος στοχεύει στη μεταφορά των αναπαραστάσεων που έχουν μαθευτεί από το αρχικό στάδιο εκπαίδευσης στη γλώσσα στόχο, όπου διατίθενται λιγότερα δεδομένα.

Multitask learning

Η έννοια και οι ρυθμίσεις για την πολυεργασιακή μάθηση (multitask learning) είναι παρόμοιες με εκείνες της μάθησης με λίγα παραδείγματα. Εδώ, ο στόχος είναι η εκμάθηση της ίδιας εργασίας σε διαφορετικές γλώσσες ταυτόχρονα. Ένα σημαντικό μέρος των αγγλικών δεδομένων χρησιμοποιείται για την προσαρμογή των παραμέτρων των μοντέλων, ενώ παράλληλα ενσωματώνεται μια μικρότερη αναλογία ελληνικών διαλόγων. Αυτή η προσέγγιση επιδιώκει να συγχρονίσει τις ενσωματώσεις (embeddings) λεκτικών μονάδων (tokens) και από τις δύο γλώσσες, συντονίζοντας κυρίως τα μοντέλα χρησιμοποιώντας τα εκτεταμένα και καλά προετοιμασμένα αγγλικά δεδομένα.

Αντί να διεξάγουμε δύο ξεχωριστές διαδικασίες μικρορύθμισης όπως στη μάθηση με λίγα παραδείγματα, συγχωνεύσαμε τα δείγματα του ελληνικού συνόλου δεδομένων με το αγγλικό σύνολο δεδομένων για να σχηματίσουμε ένα νέο, κυρίως αγγλικό, κοινό σύνολο δεδομένων. Η διαδικασία εκπαίδευσης περιλαμβάνει (fine-tuning) όλων των παραμέτρων του μοντέλου. Τα μοντέλα εκπαιδεύονται έτσι να χειρίζονται την εργασία στα αγγλικά ενώ παράλληλα προσπαθούν να κατακτήσουν την πιο απαιτητική, με λιγότερα δεδομένα, ελληνική εργασία, αξιοποιώντας τη γνώση που αποκτείται από τα αγγλικά δεδομένα.

Prompt based learning

Τα πειράματά μας με τη διαγλωσσική μεταφορά μάθησης και την πολυεργασιακή μάθηση αποκάλυψαν μια τάση των μοντέλων να ξεχνάνε πρότερη γνώση (catastrophic forgetting), ένα φαινόμενο που έχει επίσης παρατηρηθεί από άλλους ερευνητές [42]. Για την αντιμετώπιση αυτού του ζητήματος, υιοθετήσαμε μια στρατηγική που περιλαμβάνει προκαθορισμένα (hard prompts) που είναι συνεπή και στις δύο γλώσσες, παρόμοια με το [48]. Αυτά τα prompts βοηθούν στην κατεύθυνση της ροής πληροφοριών, βοηθώντας το μοντέλο να κατανοήσει βασικά στοιχεία διαλόγου που είναι κοινά μεταξύ των γλωσσών, ενισχύοντας έτσι τη μεταφορά γνώσης από τα αγγλικά στα ελληνικά. Για την εφαρμογή αυτής της στρατηγικής, προσθέσαμε στην αρχή κάθε εισόδου τη φράση 'Dialog history¨. Επιπλέον, προσθέσαμε τις φράσεις 'User:' πριν από την είσοδο του χρήστη και 'System:' πριν από την είσοδο του μοντέλου μας. Επανεφαρμόσαμε δύο ρυθμίσεις εκπαίδευσης από τα πειράματα διαγλωσσικής μεταφοράς μάθησης και πολυεργασιακής μάθησης, ορίζοντας τον αριθμό των παραδειγμάτων σε 128, χρησιμοποιώντας τα prompts όπως φαίνεται παρακάτω:

$$Dialoghistory :< context > \ User :< user\_input > \ System :< model's\_output >$$

0.4.4 Λεπτομέρειες εκπαίδευσης

Σε αυτή την ενότητα, συζητάμε τις λεπτομέρειες όλων των διαφορετικών προσεγγίσεων εκπαίδευσης που αναλύσαμε στην προηγούμενη Ενότητα 0.4.3, και τον τρόπο με τον οποίο προσαρμόσαμε κάθε προσέγγιση στα μοντέλα που συζητήσαμε στην Ενότητα 0.4.2.

Εκπαίδευση GPT2-Greek

Εκπαιδεύσαμε το μοντέλο στο μεταφρασμένο σύνολο δεδομένων DailyDialog. Κάθε περίπτωση εκπαίδευσης αποτελούνταν από έναν ολόκληρο διάλογο από αυτούς στο σύνολο εκπαίδευσης με ένα ειδικό σύμβολο που εισάγεται μεταξύ κάθε εκφώνησης του διαλόγου. Κατά τη διάρκεια της εκπαίδευσης, θέλαμε να βελτιστοποιήσουμε τον αντικειμενικό στόχο μοντελοποίησης γλώσσας απόκρισης, προσπαθώντας να προβλέψουμε την επόμενη λέξη και υπολογίζοντας την απώλεια του μοντέλου γλώσσας χρησιμοποιώντας cross-entropy ως συνάρτηση κόστους. Η απώλεια υπολογίζεται σε ολόκληρο τον διάλογο και όχι μόνο στη χρυσή απάντηση της τελευταίας πρότασης. Με αυτόν τον τρόπο, το μοντέλο μαθαίνει τα μοτίβα μεταξύ όλων των εκφωνήσεων του διαλόγου και δεν μαθαίνει μόνο να παράγει την τελική απάντηση ανάλογα με το ιστορικό του διαλόγου.

Εκπαίδευση GREEK-BERT2GREEK-BERT

Το μοντέλο GREEK-BERT2GREEK-BERT εκπαιδεύτηκε επίσης χρησιμοποιώντας το μεταφρασμένο σύνολο δεδομένων DailyDialog. Ωστόσο, χρησιμοποιήθηκε μια διαφορετική στρατηγική για τη δημιουργία των παραδειγμάτων που δόθηκαν ως είσοδο στο μοντέλο. Αρχικά προσπαθήσαμε να έχουμε τον ίδιο αριθμό παραδειγμάτων εκπαίδευσης όπως με το προηγούμενο μοντέλο, δίνοντας ως είσοδο στον κωδικοποιητή τις πρώτες $i-1$ προτάσεις του διαλόγου, όπου i είναι ο συνολικός αριθμός των προτάσεων στον διάλογο, και υπολογίζοντας το λάθος μεταξύ της εξόδου του αποκωδικοποιητή και της τελευταίας πρότασης του διαλόγου χρησιμοποιώντας cross-entropy ως συνάρτηση κόστους. Αυτή η μέθοδος οδηγεί σε χαμηλή γενίκευση, λόγω του μικρού αριθμού διαλόγων στο σύνολο δεδομένων, καθώς το μοντέλο μαθαίνει να παράγει μόνο την τελευταία πρόταση κάθε διαλόγου δεδομένων όλων των προηγούμενων και δεν μαθαίνει να παράγει κάθε γύρο του διαλόγου. Για αυτόν τον λόγο, χωρίσαμε κάθε διάλογο σε μικρότερους διαλόγους που σχημάτισαν τις διαφορετικές περιπτώσεις εκπαίδευσης, όπως φαίνεται παρακάτω. Από έναν παραδειγματικό διάλογο τεσσάρων φράσεων, μετά την επεξεργασία, προκύπτουν 3 νέοι διάλογοι.

Εκπαίδευση mT5

Το μοντέλο mT5 χρησιμοποίησε την ίδια δομή για τις περιπτώσεις εκπαίδευσης με το μοντέλο GREEK-BERT2GREEK-BERT και πειραματιστήκαμε και με τις τέσσερις προσεγγίσεις εκπαίδευσης που αναφέρθηκαν προηγουμένως. Τα μοντέλα που προέκυψαν από αυτές χαρακτηρίστηκαν ως mT5-native (mT-NV), mT5-cross-lingual (mT5-CL), mT5-multitask (mT5-MTL, και mT5-prompt (mT5-P) αντίστοιχα. Στόχος ήταν βελτιστοποίηση της γλωσσικής μοντελοποίησης υπλογίζοντας κάθε φορά την επόμενη λέξη, και χρησιμοποιώντας όπως και προηγουμένως cross-entropy ως συνάρτηση κόστους.

Εκπαίδευση XGLM

Το μοντέλο XGLM ακολούθησε την ίδια ρύθμιση με το μοντέλο GPT2-Greek για τις περιπτώσεις εκπαίδευσης. Ως πολύγλωσσο μοντέλο, υποβλήθηκε σε τέσσερις διαφορετικές τεχνικές εκπαίδευσης: XGLM-native (XGLM-NV μικρορυθμισμένο αποκλειστικά σε ελληνικούς διαλόγους), XGLM-cross-lingual (XGLM-CL εκπαιδευμένο διαδοχικά σε αγγλικά και στη συνέχεια σε ελληνικά σύνολα δεδομένων χρησιμοποιώντας k παραδείγματα), XGLM-multitask (XGLM-MTL μικρορυθμισμένο ταυτόχρονα και στα δύο σύνολα δεδομένων χρησιμοποιώντας k παραδείγματα), όπου και στις δύο περιπτώσεις $k = 32, 64, 128, 512, 1024$, και XGLM-prompt όπου εκπαιδεύτηκε χρησιμοποιώντας μάθηση με λίγα παραδείγματα (XGLM-P-CL) και παράλληλη μάθηση (XGLM-P-MTL).

0.4.5 Αξιολόγηση

Για την αξιολόγηση της απόδοσης των γλωσσικών μοντέλων, χρησιμοποιήσαμε διάφορες μετρικές όπως η διαπλεκτικότητα (perplexity), το SacreBLEU, το Distinct-N, και το BertScore. Καθένα από αυτά τα εργαλεία μας βοηθά να κατανοήσουμε διαφορετικές πτυχές της απόδοσης του μοντέλου, από τη ρευστότητα και την ποικιλομορφία μέχρι την ακρίβεια σε σχέση με μια χρυσή απάντηση αναφοράς και την ευθυγράμμιση με το πλαίσιο.

Perplexity: Αξιολογεί την αβεβαιότητα του μοντέλου στην πρόβλεψη της επόμενης λεκτικής μονάδας. Μια χαμηλότερη βαθμολογία υποδεικνύει ένα πιο σίγουρο μοντέλο, αντανακλώντας καλύτερη κατανόηση της γλώσσας.

SacreBLEU, είναι μια βελτιωμένη έκδοση της βαθμολογίας BLEU, η οποία χρησιμοποιείται ευρέως για την αξιολόγηση της ποιότητας του κειμένου που παράγει ή μεταφράζει ένα μοντέλο. Περιλαμβάνει διάφορες μετρικές:

- **BLEU-1**: Μετρά την αντιστοίχιση μεμονωμένων λεκτικών μονάδων (μονογράμματα) μεταξύ της εξόδου του μοντέλου και της αναφορικής απάντησης. Αξιολογεί την ακρίβεια στο πιο βασικό επίπεδο παραγωγής κειμένου.

- **BLEU-2**: Αξιολογεί τη συν-εμφάνιση δύο διαδοχικών λεκτικών μονάδων (διγράμματα) στην έξοδο του μοντέλου σε σύγκριση με την αναφορική απάντηση. Αυτό μετρά πόσο καλά το μοντέλο συλλαμβάνει φράσεις δύο λέξεων, αντανακλώντας περισσότερο τις συντακτικές δομές από το BLEU-1.

- Συνολική βαθμολογία **BLEU**: Αυτή η βαθμολογία συγκεντρώνει την απόδοση του μοντέλου σε διαφορετικά μήκη n-gram (έως 4), σταθμισμένα με γεωμετρικό μέσο. Παρέχει μια ολοκληρωμένη εικόνα του πόσο καλά η έξοδος του μοντέλου ευθυγραμμίζεται με το αναφορικό κείμενο σε διαφορετικά επίπεδα λεπτομέρειας.

Distinct-N: Οι μετρικές Distinct-N, συμπεριλαμβανομένων των Distinct-1 και Distinct-2, μετρούν την ποικιλομορφία του παραγόμενου κειμένου μετρώντας τα μοναδικά N-grams κανονικοποιημένα με τον συνολικό αριθμό των λέξεων. Υψηλότερες τιμές υποδεικνύουν πλουσιότερο και πιο ποικίλο λεξιλόγιο.

BertScore: Το BertScore ελέγχει τη σημασιολογική ομοιότητα μεταξύ της εξόδου του μοντέλου και του αναφορικού κειμένου χρησιμοποιώντας embeddings βασισμένα στο BERT. Υψηλές τιμές BertScore υποδηλώνουν ισχυρή σημασιολογική ευθυγράμμιση, υποδεικνύοντας αποτελεσματική κατανόηση του πλαισίου και συνάφεια των αποκρίσεων του μοντέλου.

0.4.6 Αποτελέσματα

Αυτή η ενότητα παρουσιάζει την ολοκληρωμένη αξιολόγηση διαφόρων μοντέλων που εκπαιδεύτηκαν χρησιμοποιώντας διαφορετικές μεθοδολογίες: native training, cross-lingual transfer learning, multitask learning, και prompt-learning εκπαίδευση. Στον Πίνακα 1 βρίσκεται η κύρια περίληψη των αποτελεσμάτων. Οι Πίνακες 2 και 3 δείχνουν μια μελέτη σχετικά με τον αριθμό των παραδειγμάτων k στις ρυθμίσεις cross-lingual transfer και multitask learning για τα 2 πολύγλωσσα μοντέλα.

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	score	Distinct-1	Distinct-2	Precision	Recall	F-1
Native training									
GREEK-BERT2GREEK-BERT-NV	14.16	23.82	8.43	5.66	16.72	42.23	70.58	69.77	70.01
GPT2-Greek-NV	12.47	25.93	11.07	6.93	23.13	51.28	71.53	71.47	71.37
mT5-NV	6.52	25.37	9.64	5.74	19.51	43.36	70.11	69.02	69.56
XGLM-NV	9.95	27.58	13.01	6.29	19.34	43.02	70.68	68.32	69.33
cross-lingual transfer training									
mT5-CL (k=128)	19.39	13.54	5.99	3.53	18.71	37.62	64.12	63.24	63.52
XGLM-CL (k=128)	15.74	23.61	10.03	4.95	18.75	41.91	69.60	68.32	68.99
Multitask training									
mT5-MTL (k=128)	12.27	18.9	6.83	3.93	21.12	46.35	68.35	67.96	68.04
XGLM-MTL (k=128)	16.53	23.25	9.89	4.75	18.24	40.39	69.53	68.25	68.75
Prompt-learning training									
mT5-P-CL (k = 128)	16.12	18.64	8.26	4.41	18.52	38.22	66.31	64.36	65.12
mT5-P-MTL (k = 128)	13.45	13.83	4,99	3.12	18.52	43.27	64.48	55.59	65.12
XGLM-P-CL (k=128)	10.47	25.01	12.04	4.52	18.31	40.16	69.75	68.14	68.84
XGLM-P-MTL (k=128)	11.31	25.07	12.10	5.00	16.91	38.31	69.89	68.50	69.12

NV: Native training, CL: Cross-Lingual transfer learning, MTL: Multitask learning, P: prompt
k: number of examples

Πίνακας 1: Αποτελέσματα για όλες τις διαφορετικές προσεγγίσεις εκπαίδευσης και τα μοντέλα.

Συνολικά, τα μοντέλα που εκπαιδεύτηκαν σε ολόκληρο το μεταφρασμένο σύνολο δεδομένων (native training) επιτυγχάνουν καλύτερη απόδοση σε όλες τις μετρικές. Τα αποτελέσματα δείχνουν ότι μεταξύ των μοντέλων που εκπαιδεύτηκαν εγγενώς, το mT5-NV πέτυχε το χαμηλότερο perplexity 6,52, υποδεικνύοντας ανώτερη προγνωστική απόδοση σε σύγκριση με άλλα μοντέλα σε αυτή την κατηγορία. Όσον αφορά τις βαθμολογίες SacreBLEU, που μετρούν την ομοιότητα της παραγόμενης απάντησης σε σύγκριση με μια ανθρώπινη χρυσή απάντηση, όσον αφορά τη χρήση των ίδιων ν-γραμμάτων, το XGLM-NV ξεπέρασε τα άλλα με βαθμολογίες 27,58 για B-1, 13,01 για B-2, και συνολική βαθμολογία 6,29. Αυτό υποδηλώνει ότι το XGLM-NV παράγει απαντήσεις πιο κοντά στις πραγματικές των διαλόγων, χωρίς αυτές να εξασφαλίζουν συνοχή. Ωστόσο, η μέση βαθμολογία υπολείπεται λίγο του GPT2-Greek-NV υποδεικνύοντας ότι το τελευταίο είχε καλύτερες βαθμολογίες σε πιο περίπλοκες βαθμολογίες B-3 και B-4.

Το μοντέλο GPT2-Greek-NV επέδειξε τις υψηλότερες βαθμολογίες Distinct-1 (23,13) και Distinct-2 (51,28), υποδεικνύοντας ότι παρήγαγε πιο ποικίλες εξόδους κειμένου. Αυτό είναι κρίσιμο για εφαρμογές που απαιτούν πλούσια και ποικίλη παραγωγή γλώσσας. Στο πλαίσιο του Bertscore, το οποίο αξιολογεί την ομοιότητα μεταξύ παραγόμενων και κειμένων αναφοράς χρησιμοποιώντας ενσωματώσεις BERT, το GPT2-Greek-NV πέτυχε τις υψηλότερες βαθμολογίες με ακρίβεια 71,53, ανάκληση 71,47, και βαθμολογία F-1 71,37. Η απόδοση αυτού του μοντέλου υποδηλώνει ότι είναι εξαιρετικά αποτελεσματικό στην παραγωγή κειμένου που ταιριάζει στενά με τα κείμενα αναφοράς στο νόημα και την ποιότητα.

Γενικά, το GPT-Greek-NV ξεπερνά τα άλλα μοντέλα και τεχνικές εκπαίδευσης. Αυτή η ανώτερη απόδοση πιθανώς οφείλεται στην προηγούμενη εκπαίδευσή του σε ελληνικά δεδομένα, η οποία παρείχε μια ισχυρή βάση. Επιπλέον, η χρήση ενός ελληνικού (tokenizer) συνέβαλε σημαντικά στην ικανότητά του να παράγει πιο ποικίλες αποκρίσεις. Από την άλλη πλευρά, η απόδοση του XGLM είναι κατώτερη του αναμενόμενου λαμβάνοντας υπόψη το μεγαλύτερο μέγεθός του σε σύγκριση με τα άλλα μοντέλα.

Ακόμη, τα αποτελέσματα από τον Πίνακα 1 αποκαλύπτουν ότι τα μοντέλα XGLM που εκπαιδεύτηκαν με prompts—τόσο σε σενάρια cross-lingual transfer learning (XGLM-P-CL) όσο και σε multitask learning (XGLM-P-MTL)—επιδεικνύουν ανώτερη απόδοση σε βασικές μετρικές σε σύγκριση με τα αντίστοιχά τους χωρίς ενσωμάτωση prompts. Συγκεκριμένα, το XGLM-P-CL πέτυχε βαθμολογίες SacreBLEU 25,01 για B-1 και 12,04 για B-2, οι οποίες είναι βελτιώσεις σε σχέση με το μοντέλο XGLM-CL χωρίς prompts. Αυτό υποδηλώνει ότι η συμπερίληψη prompts οδηγεί σε πιο ακριβείς και συναφείς με το πλαίσιο εξόδους διαλόγου. Επιπλέον, οι βαθμολογίες BertScore για τα XGLM-P-CL και XGLM-P-MTL (βαθμολογίες F-1 68,84 και 69,12, αντίστοιχα) είναι υψηλότερες σε σύγκριση με τα αντίστοιχά τους χωρίς prompts, υποδεικνύοντας μια στενότερη σημασιολογική ομοιότητα με αποκρίσεις διαλόγου ανθρώπινου τύπου. Ωστόσο, δεν βλέπουμε τις ίδιες βελτιώσεις για το mT5, καθώς η μάθηση με prompts δεν φαίνεται να ωφελεί το μοντέλο.

Αυτά τα ευρήματα δείχνουν ότι η εκπαίδευση βασισμένη σε prompts όχι μόνο ενισχύει τη γλωσσική ακρίβεια και συνάφεια του παραγόμενου κειμένου αλλά διασφαλίζει επίσης ότι ο διάλογος διατηρεί ένα υψηλό επίπεδο ποικιλομορφίας και πολυπλοκότητας. Αυτό είναι κρίσιμο στα συστήματα διαλόγου όπου η ικανότητα παραγωγής συνεκτικών, συναφών με το πλαίσιο και ποικίλων αποκρίσεων μπορεί να επηρεάσει σημαντικά την ικανοποίηση και τη δέσμευση του χρήστη. Επομένως, η ενσωμάτωση εκπαίδευσης βασισμένης σε prompts στο μοντέλο XGLM αξιοποιεί τα αρχιτεκτονικά του πλεονεκτήματα, επιτρέποντας πιο αποτελεσματική μάθηση από λιγότερα παραδείγματα, κάτι που είναι ιδιαίτερα επωφελές σε σενάρια με περιορισμένα δεδομένα εκπαίδευσης.

Επιπλέον, η εφαρμογή prompts στη διαδικασία εκπαίδευσης συμβάλλει σημαντικά στον μετριασμό του ζητήματος της καταστροφικής λήθης καθώς το μοντέλο μεταβαίνει από αγγλικά σε ελληνικά σύνολα δεδομένων. Η καταστροφική λήθη συμβαίνει όταν ένα νευρωνικό δίκτυο χάνει τις πληροφορίες που είχε μάθει προηγουμένως κατά την εκμάθηση νέων πληροφοριών, το οποίο είναι μια κοινή πρόκληση κατά την προσαρμογή μοντέλων σε νέες γλώσσες ή σύνολα δεδομένων. Με την ενσωμάτωση prompts, το μοντέλο XGLM είναι καλύτερα εξοπλισμένο για να διατηρήσει σχετικά χαρακτηριστικά από τα δεδομένα εκπαίδευσης στα αγγλικά ενώ αποκτά αποτελεσματικά νέα γλωσσικά μοτίβα από τα ελληνικά δεδομένα. Τα prompts λειτουργούν ως άγκυρες ή οδηγοί που βοηθούν να διατηρηθεί η εστίαση του μοντέλου σε κρίσιμες πτυχές του διαλόγου, διασφαλίζοντας ότι η μετάβαση μεταξύ γλωσσών δεν αφαιρεί προηγουμένως εδραιωμένες ικανότητες.

Παράλληλα, διεξήγαμε και μια μελέτη για τα μοντέλα mT5 και XGLM σχετικά με τεχνικές cross-lingual transfer και multitask learning, χρησιμοποιώντας $k = 32, 64, 128, 512, 1024$ τυχαία παραδείγματα από το ελληνικό σύνολο δεδομένων (Πίνακες 2 και 3).

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	average-score	Distinct-1 (%)	Distinct-2 (%)	Precision (%)	Recall (%)	F-1 (%)
mT5-MTL (k=32)	14.78	13.96	5.98	3.05	20.51	45.52	66.84	65.36	65.91
mT5-CL (k=32)	22.46	13.39	6.24	3.55	14.92	28.79	64.29	62.23	63.15
mT5-MTL (k=64)	13.06	17.32	6.64	3.77	22.82	51.27	68.23	67.56	67.82
mT5-CL (k=64)	20.19	13.37	6.06	3.49	18.13	36.14	64.11	62.97	63.44
mT5-MTL (k=128)	12.27	18.9	6.83	3.93	21.12	46.35	68.35	67.96	68.04
mT5-CL (k=128)	19.39	13.54	5.99	3.53	18.71	37.62	64.12	63.24	63.52
mT5-MTL (k=512)	9.84	21.75	7.75	4.49	21.23	47.16	68.94	68.25	68.57
mT5-CL (k=512)	14.56	17.33	6.67	3.97	20.03	42.38	65.51	65.13	65.22
mT5-MTL (k=1024)	8.92	22.98	8.25	4.85	18.65	39.66	69.37	68.60	68.85
mT5-CL (k=1024)	12.37	20.24	7.35	4.18	20.95	44.59	68.83	67.72	68.82

CL: Cross-Lingual transfer learning, MTL: Multitask learning, k: number of examples

Πίνακας 2: Απόδοση του μοντέλου mT5 στις διάφορες τεχνικές που χρησιμοποιήθηκαν για διαφορετικό αριθμό κ, ελληνικών παραδειγμάτων στα δεδομένα εκπαίδευσης.

Για το mT5-MTL, η αύξηση του k βελτίωσε την απόδοση: μειώθηκε το perplexity (8,92 στο $k = 1024$) και αυξήθηκαν οι βαθμολογίες SacreBLEU (22,98 B-1, 8,25 B-2, και μέσο σκορ με 4,85). Οι μετρικές (Distinct-1, Distinct-2) ήταν υψηλότερες στο $k = 64$ (22,82%, 51,27%), ενώ το Bertscore F-1 έφτασε 68,85% στο $k = 1024$. Αξιοσημείωτα, η ποικιλομορφία μειώθηκε με περισσότερα δείγματα καθώς το μοντέλο εστίασε σε πιο συγκεκριμένα παραδείγματα και γλωσσικά παραδείγματα ώστε να παράγει απαντήσεις.

Το mT5-CL έδειξε βελτιώσεις με αύξηση του k, αλλά υπολειπόταν του MTL. Στο $k = 1024$, πέτυχε perplexity score 12,37, SacreBLEU 20,24 (B-1), 7,35 (B-2), μέση 4,18. Σημαντική διαφορά παρατηρήθηκε στο Distinct-N: στο $k = 512$, το MTL πέτυχε 21,23% Distinct-1 και 47,16% Distinct-2, έναντι 20,03% και 42,38% του CL.

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	average-score	Distinct-1 (%)	Distinct-2 (%)	Precision (%)	Recall (%)	F-1 (%)
XGLM-MTL (k=32)	18.06	19.75	8.85	4.06	17.06	36.21	69.26	67.99	68.41
XGLM-CL (k=32)	20.33	19.98	9.38	3.29	22.54	48.68	68.86	67.97	68.36
XGLM-MTL (k=64)	17.34	20.93	9.29	4.36	17.64	38.46	69.34	67.88	68.52
XGLM-CL (k=64)	17.61	20.41	8.99	4.21	19.78	43.04	69.23	67.98	68.51
XGLM-MTL (k=128)	16.53	23.25	9.89	4.75	18.24	40.39	69.53	68.25	68.75
XGLM-CL (k=128)	15.74	23.61	10.03	4.95	18.75	41.91	69.60	68.32	68.99
XGLM-MTL (k=512)	14.38	25.35	11.12	5.51	18.85	41.17	69.41	67.95	68.57
XGLM-CL (k=512)	14.16	25.44	11.31	5.35	19.17	43.45	69.92	68.60	69.10
XGLM-MTL (k=1024)	13.53	26.49	11.66	5.78	19.21	41.97	69.51	67.96	68.60
XGLM-CL (k=1024)	13.31	26.13	11.42	5.91	19.11	42.88	69.26	68.60	69.23

CL: Cross-Lingual transfer learning, MTL: Multitask learning, k: number of examples

Πίνακας 3: Απόδοση του μοντέλου XGLM στις διάφορες τεχνικές που χρησιμοποιήθηκαν για διαφορετικό αριθμό κ, ελληνικών παραδειγμάτων στα δεδομένα εκπαίδευσης.

Το XGLM έδειξε παρόμοια τάση. Το XGLM-MTL ($k = 1024$) είχε perplexity 13,53 και υψηλότερες βαθμολογίες SacreBLEU: 26,49 (B-1), 11,66 (B-2), με μέση βαθμολογία 5,78 - υψηλότερες από του mT5. Οι βαθμολογίες Distinct-N βελτιώθηκαν με περισσότερα παραδείγματα, αλλά η ποικιλομορφία σταθεροποιήθηκε σε υψηλότερα k. Το BERTScore F-1 έφτασε 69,23% για το XGLM-CL ($k = 1024$), ελαφρώς υψηλότερο από το mT5.

Συμπερασματικά, η πολυεργασιακή μάθηση υπερτερεί της μάθησης με λίγα παραδείγματα για τα mT5 και XGLM σε όλες τις μετρικές (perplexity, SacreBLEU, BERTScore, και Distinct-N). Παράλληλα, αύξηση των παραδειγμάτων βελτιώνει την απόδοση, υπογραμμίζοντας τη σημασία περισσότερων δεδομένων εκπαίδευσης.

0.4.7 Ανθρώπινη Αξιολόγηση

Οι αυτόματες μετρικές, παρότι χρήσιμες για ποσοτική ανάλυση, συχνά αποτυγχάνουν να αξιολογήσουν πλήρως τις πραγματικές δυνατότητες του μοντέλου. Για μια πιο ολοκληρωμένη κατανόηση της απόδοσης των μοντέλων, διεξήγαμε μια ανθρώπινη αξιολόγηση μέσω διαδικτυακής έρευνας με 40 συμμετέχοντες, επιτρέποντας την αξιολόγηση ποιοτικών πτυχών που αυτόματες μετρικές από μόνες τους δεν μπορούν να μετρήσουν αποτελεσματικά. Βασιζόμενοι στα ευρήματα της αυτόματης αξιολόγησης από την προηγούμενη ενότητα, επιλέξαμε τα μοντέλα που επέδειξαν τα πιο υποσχόμενα αποτελέσματα για ανθρώπινη αξιολόγηση: GPT-Greek-NV, XGLM-NV και XGLM-P-MTL. Επιπλέον, συμπεριλάβαμε το Meltemi, ένα σημαντικά μεγαλύτερο μοντέλο (7B παραμέτρους σε σύγκριση με τα 550M των άλλων), για να συγκρίνουμε τα μοντέλα μας με μια πιο ισχυρή αρχιτεκτονική και να κατανοήσουμε τη διαφορά απόδοσης μεταξύ διαφορετικών κλιμάκων μοντέλων.

Στην έρευνα, στους συμμετέχοντες παρουσιάστηκαν πανομοιότυπα ιστορικά διαλόγων και τους ζητήθηκε να αξιολογήσουν τις απαντήσεις που παρήγαγαν καθένα από τα τέσσερα μοντέλα με βάση πολλαπλά κριτήρια. Αυτή η προσέγγιση μας επέτρεψε να συγκεντρώσουμε λεπτομερείς ανθρώπινες κρίσεις σχετικά με την ποιότητα των απαντήσεων που συμπληρώνουν τα αποτελέσματα των αυτόματων μετρικών.Κάθε συμμετέχων αξιολόγησε 5 διαλόγους, με κριτήρια την ευφράδεια (fluency), η οποία δείχνει τη συντακτική ορθότητα της απάντησης, και τη συνοχή (coherence) η οποία μας δείχνει πόσο συναφή με το ιστορικό του διαλόγου είναι η απάντηση, χρησιμοποιώντας κλίμακα Likert 1-5. Συνολικά συγκεντρώθηκαν 180 αξιολογήσεις ανά μοντέλο.

Το Meltemi επέδειξε την καλύτερη απόδοση με στατιστικά σημαντικές βαθμολογίες 4,01 στην ευφράδεια και 3,97 στη συνοχή (p῾0,05), επιβεβαιώνοντας την ανωτερότητά του που μπορεί να αποδοθεί στο μεγαλύτερο μέγεθος και την εκτενέστερη εκπαίδευσή του. Το XGLM-P-MTL παρουσίασε σημαντική βελτίωση έναντι του XGLM-NV τόσο στην ευφράδεια (3,46 έναντι 3,13) όσο και στη συνοχή (2,98 έναντι 2,62), αποδεικνύοντας την αποτελεσματικότητα της προσέγγισης εκπαίδευσης πολλαπλών εργασιών με αγγλικά prompts. Το GPT-Greek-NV απέδωσε καλά στην ευφράδεια (3,42) αλλά υστέρησε στη συνοχή (2,90), υποδηλώνοντας δυσκολία στη διατήρηση συνάφειας σε εκτενείς απαντήσεις.

Model	Fluency	Coherence
GPT-Greek-NV	3,42	2,90
XGLM-NV	3,13	2,62
XGLM-P-MTL	3,46*	2,98*
Meltemi	4,01*	3,97*

Πίνακας 4: Σύγκριση μοντέλων ως προς την ευφράδεια και τη συνοχή. Τα αποτελέσματα με * είναι στατιστικά σημαντικά με π῾0,05 χρησιμοποιώντας το τεστ MannWitney U.

0.5 Συνεισφορές και μελλοντικές προεκτάσεις

0.5.1 Συνεισφορές

Η έρευνά μας περιλάμβανε μια ολοκληρωμένη σειρά πειραμάτων χρησιμοποιώντας μια ποικιλία μονογλωσσικών και πολυγλωσσικών μοντέλων βασισμένων σε transformers. Αυτά περιλάμβαναν τα GREEK-BERT, GPT-2 Greek, mT5 και XGLM. Διερευνήσαμε διαφορετικές προσεγγίσεις εκπαίδευσης για την αποτελεσματική αξιοποίηση των περιορισμένων πόρων. Αυτές οι προσεγγίσεις περιλάμβαναν διαγλωσσική μεταφορά γνώσης zero-shot, few-shot και full-shot, καθώς και εγγενή εκπαίδευση. Επιπλέον, διερευνήσαμε τη χρήση τεχνικής εκμάθησης με προτροπές (prompt learning) για τη βελτίωση της απόδοσης των πολυγλωσσικών μοντέλων μας, αποδεικνύοντας την αποτελεσματικότητά της στη βελτίωση της παραγωγής διαλόγου.

Αξιολογήσαμε τα μοντέλα χρησιμοποιώντας διάφορες αυτόματες μετρικές: Perplexity, BLEU, BertScore και Distinct-n. Αυτές οι μετρικές βοήθησαν στην αξιολόγηση της ποιότητας, της ποικιλομορφίας και της συνάφειας των παραγόμενων απαντήσεων. Η αξιολόγησή μας αποκάλυψε ότι η εγγενής εκπαίδευση γενικά ξεπέρασε άλλες τεχνικές, με το XGLM-P-MTL να είναι το μόνο συγκρίσιμο μοντέλο. Αυτό το μοντέλο εκπαιδεύτηκε ταυτόχρονα σε αγγλικά δεδομένα διαλόγου και ένα μικρό μέρος ελληνικών δεδομένων χρησιμοποιώντας συνεπείς προτροπές σε όλες τις γλώσσες.

Για περαιτέρω αξιολόγηση της απόδοσης και σύγκριση με μεγαλύτερα μοντέλα, διεξήγαμε μια έρευνα ανθρώπινης αξιολόγησης συγκρίνοντας τα τρία καλύτερα μοντέλα σύμφωνα με τις αυτόματες μετρικές (GPT-Greek-NV, XGLM-NV και XGLM-P-MTL) μαζί με ένα πιο προηγμένο αυτοπαλινδρομικό μοντέλο, το Μελτέμι (Meltemi). Τα αποτελέσματα της έρευνας έδειξαν ότι το Μελτέμι ξεπέρασε όλα τα άλλα μοντέλα, ακολουθούμενο από το XGLM-P-MTL, το οποίο επέδειξε στατιστικά σημαντική βελτίωση σε σχέση με το μοντέλο XGLM που εκπαιδεύτηκε αποκλειστικά σε μεταφρασμένα ελληνικά δεδομένα.

Τα ευρήματα των πειραμάτων μας προσφέρουν πολύτιμες γνώσεις σχετικά με τις πολυπλοκότητες της παραγωγής διαλόγου σε γλώσσες με περιορισμένους πόρους. Τα αποτελέσματά μας υπογραμμίζουν τις δυνατότητες της διαγλωσσικής μεταφοράς μάθησης ως βιώσιμη στρατηγική για τέτοια σενάρια όταν συνδυάζεται με κάποια εκμάθηση προτροπών, παρέχοντας ένα μονοπάτι για μελλοντική έρευνα και ανάπτυξη.

0.5.2 Μελλοντικές Προεκτάσεις

Αυτή η διπλωματική εργασία ανοίγει πολλές οδούς για μελλοντική έρευνα. Πιθανές επεκτάσεις και προσαρμογές για περαιτέρω διερεύνηση περιλαμβάνουν:

- Εφαρμογή προηγμένων τεχνικών επαύξησης δεδομένων πέρα από τη μηχανική μετάφραση, όπως παράφραση και συνθετική παραγωγή διαλόγων, για τον εμπλουτισμό των συνόλων δεδομένων εκπαίδευσης για γλώσσες με περιορισμένους πόρους.

- Ανάπτυξη και δοκιμή εξειδικευμένων προσεγγίσεων μηχανικής προτροπών (prompt engineering) προσαρμοσμένων σε συστήματα διαλόγου, ιδιαίτερα διερευνώντας παραδείγματα μάθησης few-shot που αξιοποιούν τη διαγλωσσική μεταφορά γνώσης.

- Επέκταση της αξιολόγησης μοντέλων σε διάφορους τομείς και πλαίσια διαλόγου για την αξιολόγηση της ευρωστίας και της προσαρμοστικότητας σε διαφορετικά σενάρια συνομιλίας.

- Σχεδιασμός ολοκληρωμένων πλαισίων ανθρώπινης αξιολόγησης που συλλαμβάνουν λεπτές πτυχές της ποιότητας του διαλόγου, συμπεριλαμβανομένης της συνοχής, της δέσμευσης και της πολιτισμικής καταλληλότητας.

- Επέκταση των μεθοδολογιών που αναπτύχθηκαν σε αυτή τη διπλωματική εργασία σε άλλες γλώσσες με περιορισμένους πόρους, ιδιαίτερα σε εκείνες με περιορισμένη εκπροσώπηση στην έρευνα Επεξεργασίας Φυσικής Γλώσσας (NLP).

- Διερεύνηση αποτελεσματικών τεχνικών λεπτομερούς συντονισμού (fine-tuning) που ελαχιστοποιούν τους υπολογιστικούς πόρους, μεγιστοποιώντας παράλληλα τα οφέλη απόδοσης για συστήματα διαλόγου.

Η έρευνα που παρουσιάζεται σε αυτή τη διπλωματική εργασία αποτελεί θεμέλιο για την αντιμετώπιση των προκλήσεων της δημιουργίας αποτελεσματικών συστημάτων διαλόγου για γλωσσικά υποεκπροσωπούμενες κοινότητες.

Chapter 1

Introduction

1.1 Motivation

The rapid advancement of natural language processing (NLP) technologies has revolutionized various aspects of human-computer interaction. Virtual assistants like Siri and Alexa, alongside sophisticated customer service chatbots, have become integral parts of our daily lives. These dialogue response generation systems help us with tasks across multiple domains—such as booking flights, making restaurant reservations, and shopping online—and also entertain us.

Advances in Machine Learning (ML), particularly through Deep Learning (DL), have spurred the development of diverse dialogue systems. Notably, open-domain chatbots are a significant area of research. These systems aim to emulate human-like interactions by engaging in free-flowing conversations on any topic, much like humans. Modern open-domain conversational models are usually trained on extensive datasets and enhanced through massively pre-trained language generation models, which are fine-tuned to perform specific dialogue tasks [91], [63].

However, despite the wealth of resources for building these models [89], [37], [62], most are predominantly in English, presenting significant challenges in developing similar technologies for other languages. While limited research has addressed low-resource dialogue generation—such as template-based approaches for Chinese [87] and simple fine-tuning on auto-translated Arabic data [53], [54]—these efforts have focused on single training methodologies without systematic exploration of cross-lingual transfer strategies or modern multilingual model capabilities.

Greek, with its rich historical and cultural heritage, is considered a low-resource language in the context of NLP due to the scarcity of large-scale annotated datasets and linguistic resources. To the best of our knowledge, no prior work has addressed open-domain dialogue generation for Greek, creating a significant gap in conversational AI accessibility for Greek speakers.

This thesis addresses these limitations through several key innovations. Initially, we present the first systematic comparison of multiple training strategies (native, cross-lingual transfer, multitask, and prompt-based learning) for low-resource dialogue generation, moving beyond the single-approach focus of previous work. Subsequently, through this comprehensive evaluation, we identify and develop a novel prompt-based cross-lingual transfer methodology that uses shared linguistic structures to facilitate knowledge transfer from English to Greek dialogues—an approach that emerges as superior and has not been previously explored in open-domain dialogue generation scenarios. Furthermore, we demonstrate how modern multilingual models can be effectively leveraged through strategic combination of source language data, limited target language data, and consistent prompt structures—establishing a new framework for efficient dialogue system development in low-resource settings.

1.2 Thesis Contributions

This diploma thesis contributes to the field of open-domain dialogue systems for low-resource languages, specifically addressing the challenges of developing such systems for the Greek language.

The principal contributions of this work are fourfold. We begin by addressing the absence of Greek dialogue data through the creation of a Greek version of the Daily Dialog dataset via machine translation, enabling experimental research in Greek dialogue generation that was previously impossible.

Subsequently, we conduct the first comprehensive systematic evaluation of multiple training methodologies for low-resource dialogue generation, comparing native training, cross-lingual transfer learning, multitask learning, and prompt-based approaches across both monolingual and multilingual transformer architectures. This systematic comparison provides crucial insights into the relative effectiveness of different strategies and represents the first such comparative study in this domain.

Through this comprehensive evaluation, we identify and develop a novel prompt-based cross-lingual transfer learning approach for dialogue generation that employs shared prompt structures across languages to facilitate knowledge transfer from high-resource to low-resource languages. This methodological innovation, which emerges as the most effective strategy from our systematic comparison, significantly enhances multilingual model performance in few-shot scenarios and represents the first application of such techniques to open-domain dialogue generation.

Most significantly, we establish an optimal training framework for low-resource dialogue generation that combines multilingual models with both source language data and small amounts of target language data, enhanced by consistent prompt structures across languages. This approach achieves performance comparable to or better than models trained solely on target language data, while requiring significantly less target language resources—providing a practical solution for developing dialogue systems in resource-constrained settings.

1.3 Thesis outline

Chapter 2: **From Machine Learning to Deep Learning**, provides background knowledge to set the stage for the subsequent chapters. First, we provide an overview of technical information that is relevant to understand the contents of this thesis. Next, we introduce the reader to machine learning together with its most elementary methods. We subsequently delve into the original deep learning models and the basic architectures. Then, transfer learning methods that are currently used to train natural language processing models are explained.

Chapter 3: **Natural Language Processing**, presents the natural language processing background needed to understand this thesis. After briefly presenting popular natural language processing tasks, language modeling is presented, initially in the form of an n-gram model based on the Markov assumption and then as a recurrent neural network. The chapter then transitions to large-scale pre-trained transformer models, highlighting the most significant architectures and models. It concludes with an analysis of various prompt learning techniques that enable the efficient training of larger models with smaller datasets.

Chapter 4: **Open-Domain Dialogue Generation for Low-Resource Languages** surveys the approaches researchers have taken to address open-domain dialogue generation in languages with limited or no available data. It starts by describing the problem and then examines different training strategies for Transformer-based models, focusing on their success with varying data quantities. The chapter also reviews the evaluation metrics used to assess such tasks and discusses the insights each metric provides.

Chapter 5: **Dialogue Generation - Greek Case** details the development of a Greek open-domain dialogue generation system using various models and techniques. It covers a range of models, from encoder-decoder to decoder-only, and from monolingual to multilingual approaches. The chapter discusses the use of different subsets of datasets in combination with different training techniques and their impact on performance. It concludes with a comparative analysis of all the models and techniques, presenting the best results achieved.

Chapter 6: **Conclusions**, contains our conclusion, summarizing our findings and providing an outlook into the future work.

Chapter 2

Machine Learning and Deep Learning

2.1 Introduction

Machine learning and deep learning are two interconnected subfields of artificial intelligence that have revolutionized the way computers learn from data and make decisions. Machine Learning is at the connection between computer science and statistics. Its goal is the algorithmic use of data, both structured and unstructured, to mimic how humans learn and to perform tasks without being explicitly programmed. Deep learning is a subdomain of machine learning and has gained significant attention for its ability to learn representations from complex data. Deep learning involves neural networks with numerous layers of interconnected artificial neurons, which are inspired by the structure of the human brain.

Machine learning and Deep learning have found applications in various fields, such as computer vision, natural language processing, healthcare, and finance, transforming the way we approach various tasks and challenges. With their ability to uncover hidden patterns and handle massive amounts of data, machine learning, and deep learning have become essential tools for solving complex problems in the field of artificial intelligence.

In this chapter, we provide an in-depth review of the technical components that form the foundations of machine learning and deep learning.

2.2 Types of learning

Machine Learning algorithms can be classified depending on the way of learning. The most common categories are supervised learning, unsupervised learning, semi-supervised learning, and self-supervised learning.

2.2.1 Supervised Learning

In supervised learning, the model can make predictions with the help of a labeled dataset. The aim is to build an algorithm that learns the mapping function f between input variables X and target variables Y.

$$Y = f(X) \tag{2.1}$$

The labeled dataset contains the corresponding target variable Y for every variable X. During training, both X and their corresponding label Y are provided. At inference, we expect the mapping function to predict the output for every new input sample provided from the same distribution as the training samples. Supervised learning can be further divided into two types:

1. Classification
2. Regression

In classification, the model learns to predict the class of an input sample. For example, in image recognition, the model classifies images of animals into categories such as 'dog', 'cat', or 'bird'. Another example is email spam detection, where the model classifies emails into 'spam' or 'not spam'. In regression, the model predicts a numerical value. For instance, predicting house prices based on features like size, location, and number of bedrooms, or predicting stock prices based on historical data.

2.2.2 Unsupervised Learning

In contrast to supervised learning, unsupervised learning is used for problems where the output variable Y is not available, thus the data are unlabeled. The algorithm's goal is to identify patterns and commonalities among its input data.

The main tasks of unsupervised learning are clustering, generative modeling, and dimensionality reduction. The process of clustering involves grouping the population or data points into a number of groups so that the data points within each group are more similar to one another than the data points within other groups. Simply said, the aim is to segregate groups with similar traits and assign them into clusters. For example, customer segmentation in marketing, where customers are grouped based on purchasing behavior. Generative models are the ones that mimic the method used to produce training data. A good generative model should produce new data that is somewhat similar to the training data. For instance, Generative Adversarial Networks (GANs) can generate realistic images of faces that do not exist in reality. Since the process of generating the data cannot be directly observed, this type of learning is regarded as unsupervised [17].

2.2.3 Semi-Supervised Learning

Semi-supervi- sed learning lies between supervised and unsupervised learning. Semi-supervised learning algorithms operate on datasets that are partially labeled. The labeled samples are initially used to train a model, which will be used to label the whole dataset. After that, we combine the given labels with pseudo-labels we generated to create the fully labeled dataset. It is used for tasks, for which it is infeasible to label every sample, like web content classification or text document classification. For example, in a large-scale text classification task, a small subset of the documents might be manually labeled by experts, and the rest of the documents are labeled using the model trained on this subset.

2.2.4 Self-Supervised Learning

Self-supervised learning is a type of learning where the model generates its own labels from the input data, effectively creating a supervised learning problem from an unlabeled dataset. This approach leverages the structure within the data itself to create pseudo-labels.

In self-supervised learning, the algorithm typically uses part of the input data to predict another part. This can be done through tasks such as predicting the next word in a sentence, completing masked parts of an image, or predicting future frames in a video sequence. For example, BERT (Bidirectional Encoder Representations from Transformers) uses self-supervised learning by masking out words in a sentence and training the model to predict these masked words [14]. Large Language Models (LLMs), such as GPT (Generative Pre-trained Transformer), also leverage self-supervised learning. These models are trained on vast amounts of text data by predicting the next word in a sentence, which inherently creates a supervised learning task from the text [8].

2.3 Basic concepts in machine learning

2.3.1 Loss function

Any Supervised Learning algorithm's goal is to return a mapping function $f()$, that maps the input instances to the corresponding labels. In order to quantify the error (loss) of the model, we introduce the loss function $L(y, \hat{y})$, where $\hat{y}$ is the predicted output and y is the true label. The Loss function $L(y, \hat{y})$ assigns a numerical score (a scalar), and the lower the numerical score, the better the prediction made. During the training phase, the parameters θ of the mapping function are determined by minimizing the loss L. Given a train set ($x_{1:n}$,$y_{1:n}$), and the function $f(x; \theta)$ the total loss over the training set is defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(x_i;\theta), y_i) \tag{2.2}$$

The model's optimal parameters θ are determined by minimizing the total loss L.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\,\mathcal{L}(\theta) = \underset{\theta}{\operatorname{argmin}} -\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(x_i;\theta), y_i) \tag{2.3}$$

For different tasks, different loss functions maybe should be selected. Classification (binary or multi-label) and regression tasks use different cost functions. Also, for the same task and dataset, a different loss function can give better results than another one. The most common cost functions are described below:

Mean Squared Error(MSE): The mean squared error prediction is mainly used for regression models and is described as:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.4}$$

Where (y_i) is the true value, $(\hat{y}_i)$ is the predicted value and θ is the parameter vector of the network.

Cross-entropy loss: Cross-entropy loss is mainly used for classification problems. On this kind of task, the model predicts an output with probability $p \in [0, 1]$. For binary classification, the output $\hat{y}$ of the model is interpreted as the conditional probability $\hat{y} = P(y = 1 \mid x)$. We want to maximize the log conditional probability $P(y = 1 \mid x)$, or equivalently minimize the cross-entropy loss. Let $\mathbf{y} = (y_1, y_2, \dots, y_m)$ be a vector representing the true multinomial distribution over the labels $1, \dots, m$, and let $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$ be the linear classifier's output, the loss function is defined as:

$$\mathcal{J} = -\sum_{i=1}^{2} y_i \log(\hat{y}_i) \tag{2.5}$$

For multi-label classification problems, we use the negative log-likelihood loss, also known as categorical cross-entropy loss. The loss is calculated among two probability distributions, we call p the predicted probability distribution and q the ground truth distribution. The goal is to minimize the cross-entropy between the model's distribution and the distribution of the given data. The formula, for M classes classification, is given as:

$$\mathcal{J}(p, q) = -\sum_{k=1}^{M} q_k \log p_k \tag{2.6}$$

The negative log-likelihood loss is important and widely used in training language generative models, such as the models we use.

2.3.2 Optimization

Machine learning optimization is the process of updating the parameters, during the training phase, to minimize the loss. The most common category of optimization algorithms is gradient-based. These methods minimize the loss function $J(\theta)$ by updating the parameters θ of the model in the opposite direction of the gradient $\nabla_\theta J(\theta)$.

Gradient Descent algorithm computes the gradient of the cost function with respect to the parameters θ for the entire training dataset at each iteration $n+1$, and is defined as:

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta J(\theta_n) \tag{2.7}$$

The learning rate η determines the size of the steps the algorithm takes to reach a minimum. However, this algorithm is possible to be stuck at a local minimum and not reach a global one. Therefore, it is important to carefully adjust the learning rate because a small value results in slow convergence while a large value may cause the cost function to fluctuate around a minimum. Additionally, calculating the loss over the entire dataset at each iteration may be computationally expensive, if we operate on a large dataset, as it recomputes gradients for similar examples before each parameter update.

Stochastic Gradient Descent [7], in contrast, computes the gradient of the cost function and performs a parameter update over a subset of the sample. For each training example, x_i and its corresponding label y_i the parameter update at step $n+1$, is defined as:

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta J(\theta_n; x_i, y_i) \tag{2.8}$$

SGD algorithm is much faster than gradient descent as it performs one update at a time and avoids redundant computations.

2.3.3 Backpropagation

Backpropagation is a standard method for neural network training. The term is a short form for "backward propagation of errors". During the training process, to find the optimal parameters (weights) of the model, we minimize the loss function. This method helps calculate the gradients of the loss function with respect to the weights of the network[32]. Backpropagation computes the gradients of a complex expression using the chain rule, one layer at a time. The algorithm starts from the last layer of the network and iterates backward while caching the intermediate terms. With this method we update the weights of the network after each computation of the cost function L, using the partial derivatives $\frac{\partial L}{\partial w}$ with respect to any learnable weight w. In this way, we fine-tune the weights, which leads to a further decrease in the model's loss and improves the performance.

2.3.4 Generalization: Underfitting and Overfitting

The main goal of training in machine learning is to develop the model's ability to generalize successfully, thus performing well in previously unseen data. Generalization is an important concept in machine learning and examines how well a model can digest new data and make correct predictions after getting trained on a fixed training set.

During the training phase, the model computes an error (training error) based on predictions on the training set, which it tries to minimize through the backpropagation process. After the training, the model is tested in a different, called the testing dataset which probably includes previous unseen input samples. The error that is computed on the testing dataset is called generalization or testing error. The success of a machine learning algorithm can be determined by how the model handles data seen during the training process and the way it adapts to unseen data. Simply, the goal is to make the training error small and at the same time make the gap between training and generalization error small, as shown in Fig 2.1.

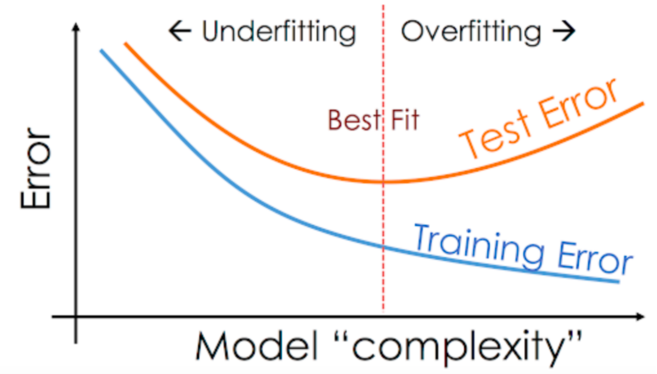


Figure 2.1: At the left end of the graph, where training and test error are both high is the underfitting regime. As we increase capacity, training error decreases, but the gap between training and generalization error increases. Eventually, the size of this gap outweighs the decrease in training error, and we enter the overfitting regime, where capacity is too large, above the optimal capacity. Source: vitalflux

The two factors mentioned above represent two important machine learning concepts: underfitting and overfitting. Underfitting occurs when our machine learning model is not able to capture the underlying pattern of the training data, thus the training error is not low enough. Overfitting occurs when the model learns the training data really well. As a result, the model starts modeling noise and inaccurate values present in the dataset. This has an effect on the model's performance in unseen data, resulting in a large gap between training and testing error. So, it's critical to find a good trade-off between training error and the gap between training and test error during the training phase.

2.3.5 Tackling overfitting

Overfitting is one of the most common issues encountered when training a machine learning model. There are several approaches to overcoming overfitting and improving generalization. Regularization and dropout are the two most common.

Regularization is the most common technique that is used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting. The regularization term in the adjusted loss function now penalizes complex hypotheses that we believe are unlikely to generalize well and favor simple ones. With the new regularization term, our cost function described in equation 2.2 is now defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(x_i;\theta), y_i) + \lambda R(\theta) \tag{2.9}$$

The regularization term(R) considers the parameters θ and scores their complexity. The training algorithm is now encouraged to find a compromise between the fit of the training data and the norms of the weights. The two most common regularization norms are L2 and L1.

- **L1 Regularization:**

  The L1 norm penalty, also known as Lasso regression, uses the L1 norm (also called Manhattan distance), and is thus defined as:

  $$R(\mathbf{w}) = \sum_{i=1}^{K} |w_i|$$

  where $w_i, i = 1, 2, \ldots, K$ are the model's weights.

- **L2 Regularization:**

  The L2 norm penalty, also known as weight decay or ridge regression, uses the L2 or Euclidean norm, and is thus defined as:

  $$R(\mathbf{w}) = \sum_{i=1}^{K} w_i^2$$

  where $w_i, i = 1, 2, \ldots, K$ are the model's weights.

Comparing the two regulation methods, we can conclude the following: Due to squaring, L2 regularization penalizes parameters with bigger values much more strongly, while it only affects smaller values lightly. It can therefore reduce overfitting by decreasing model complexity, but, since it does not lead any parameters to become equal to zero and only decreases them, it does not reduce the total parameter number. On the other hand, L1 regularization affects all values equally, decreasing all non zero parameters and leading some of them to assume an optimal value of zero. For this reason, L1 regularization often leads to more sparse models than L2 regularization

Dropout is a form of stochastic regularization. By injecting some stochasticity into the computations, we can sometimes prevent certain pathological behaviors and make it hard for the network to overfit. Dropout is intended to prevent the network from relying on specific weights. The algorithm itself is simple: we drop out each individual unit(hidden or visible) with some probability p by setting its activation to zero, as shown in Fig 2.2. Dropout technique has become part of the standard toolbox for neural network training and can give a significant performance boost. [71]

2.3.6 Activation Functions

An activation function is the last part of a layer in a neural network and decides if a neuron should be activated or not. This means that the activation function will decide how important is the neuron's input to the network. Basically, is a mathematical function that transforms the weighted sum of a neuron's input into an output. If f is the activation function of a neuron with n inputs $x_1, x_2, \ldots, x_n$, and bias b then the output y is defined as:

$$y = f(w_1x_1 + \ldots + w_nx_n + b) \tag{2.10}$$

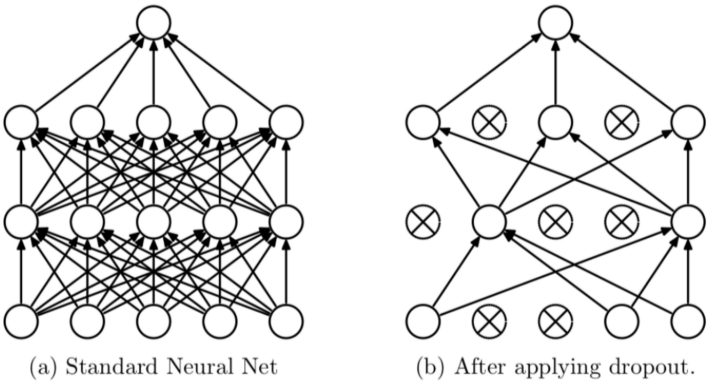


Figure 2.2: . Dropout Neural Net Model. Left: A standard neural net with 2 hidden layers. Right: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped. Source: [71]

The simplest form of an activation function can be defined as a binary function that activates the neuron based on the input and the output is either 0 or 1. Activation functions can be linear or non-linear. Linear functions, also known as no-activation functions, do not change the weighted sum and the activation is proportional to the input. So, we usually use non-linear functions in order to add non-linearity to the network, learn non-linear states, and create complex mappings between the network's inputs and outputs. The most common non-linear functions are presented below.

Sigmoid Function

The sigmoid function is commonly used for models we have to predict the probability as an output, as it squishes the value between the range of 0 and 1. The mathematical form of the function is:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.11}$$

The larger the input, the closer the output value will be to 1, whereas the smaller the input (more negative) the closer the value will be to 0. However, large changes in the inputs of the sigmoid will cause a small change in the output because of the compression we mentioned before. Thus, the derivatives become small and after stacking many layers with sigmoid as the activation function, during the backpropagation algorithm, all these derivatives are multiplied together and lead to the vanishing of the gradients.

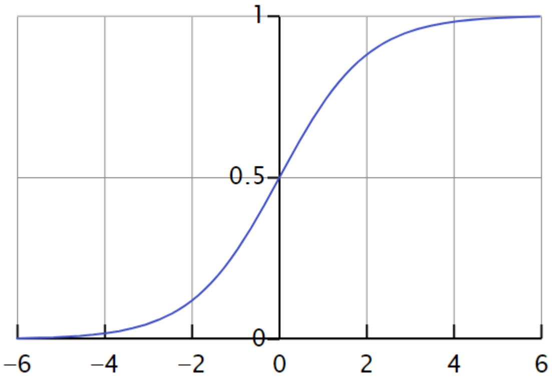


Figure 2.3: The sigmoid function

Tanh Function

The hyperbolic tangent (tanh) function is mathematically represented as:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.12}$$

Tanh function is similar to the sigmoid with the difference that is zero-centered and the output is in the range $[-1, 1]$. It has a steeper gradient compared to the sigmoid function which has the advantage of faster learning, but Tanh also faces the problem of vanishing gradients.

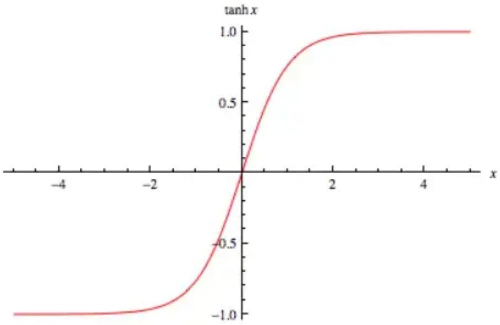


Figure 2.4: The tanh function

ReLU Function

Rectified Linear Unit (ReLU) is probably the most commonly used activation function in neural networks and is defined as:

$$f(x) = \max(x, 0) \tag{2.13}$$

The function returns the value of the input or 0 if the input is negative. It seems like a linear function but has a derivative function and allows for backpropagation. Also, the neurons with negative output (those with negative activation function input) will be deactivated. Since only a certain number of neurons are activated, the ReLU function is far more computationally efficient when compared to the sigmoid and tanh functions. Moreover, ReLU due to its linear, non-saturating property can accelerate the convergence towards the global minimum of the loss function.

GELU Function

Gaussian Error Linear Unit (GELU) [22] is the result of combining the dropout technique (see Section 2.3.5) and the ReLU activation function. Both ReLU and dropout yield a neuron's output. The first one does it deterministically, while the second one stochastically by randomly multiplying a few activation functions by 0 at certain nodes in layers. GELU function merges these functionalities by multiplying the input by either zero or one which is stochastically determined and is dependent upon the input. Mathematically it can be represented as:

$$GELU(x) = xP(X < x) = x\Phi(x) = 0.5(1 + \tanh[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)]) \tag{2.14}$$

Where $\Phi(x)$ is the standard Gaussian cumulative distribution function. GELU's nonlinearity is better than ReLU's and is the most commonly used activation function in the top NLP models like BERT [14], GPT [59], and RoBERTa [46].

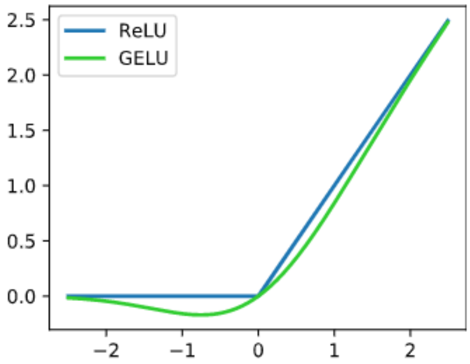


Figure 2.5: Comparison between ReLu and GELU function

2.4 Traditional Machine Learning Models

2.4.1 Linear Regression

Linear regression is a supervised algorithm that learns to model a dependent variable, y, as a function of some independent variables, x_i, by finding a line (or surface) that best "fits" the data. For an input vector $\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$, we define the output value $\hat{y}$ as:

$$\hat{y} = \mathbf{w}^T \mathbf{x} \tag{2.15}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the parameter vector.

The error function we try to minimize in order to find the optimal fit of the model to training is the mean squared error (loss) function, which is defined in Eq. 2.4. Minimizing this function with respect to the parameters $\mathbf{w}$ we find the optimal parameters. The solution is also called the least squares solution. We can calculate the optimal solution using the gradient descent algorithm we described earlier. Even though the family of linear models, like linear regression, has considerable limitations, they serve as the foundation for more complicated models, such as neural networks, which we shall examine later.

2.4.2 Support Vector Machine

Support Vector Machine (SVM) is one of the most performant off-the-shelf supervised machine learning algorithms [12]. Suppose given some data points in the N-dimensional space belonging to one of two different classes. If the data are linearly separable, an SVM algorithm will calculate the optimal hyperplane:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = 0 \tag{2.16}$$

that separates the data of each class, where $\phi(\mathbf{x})$ denotes a fixed feature-space transformation and b is a bias parameter. Finding the optimal hyperplane means that the SVM algorithm finds the decision plane that has the maximum margin between the samples of each category, thus the maximum distance from the nearest data point on each side. We just use a fraction of the training data points to discover the optimal hyperplane location (those which are closest to the decision plane). These data points are referred to as support vectors.

In the case of nonlinearly separable data, SVM models can still find a solution performing non-linear classification by mapping the input vectors to a higher-dimensional space, where they are more likely to be linearly separable. For this mapping SVM models employ a kernel function. The most common kernel functions are:

- Polynomial kernel: $k(x_i, x_j) = (x_i \cdot x_j + 1)^d$
- Gaussian radial basis kernel: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ for $\gamma > 0$

2.5 Deep Learning

Deep learning is a subset of a larger class of machine learning approaches that combine artificial neural networks with representation learning. This section will discuss deep-learning models and their ability to handle big amounts of data to extract high-level features. Deep learning models are composed of stacked neural network levels, with each level serving as a function that learns to transform the input data into a representation. Deep learning has been applied to numerous fields of study, including computer vision, speech recognition, natural language processing, bioinformatics, and medical image analysis.

2.5.1 Feedforward Neural Networks

A Feed Forward Neural Network is an artificial neural network (ANN) in which the connections between nodes do not form a loop. They were the first ANNs proposed and are mainly used in supervised learning problems with non-sequential input data. Feedforward neural networks are so named because all information flows in a forward manner only; from the input layer to the hidden layers and then to the output layer. Regardless of whether the data passes through multiple hidden nodes, it always travels in one direction and never backward.

A Feed Forward Neural Network is commonly seen in its simplest form as a single-layer perceptron. Perceptron consists of one node that computes the weighted sum of its inputs and passes the result through a non-linear activation function. An extension of the single-layer perceptron is the multi-layer perceptron(MLP). It consists of an input layer, one or multiple stacked hidden layers, and an output layer and each layer includes multiple perceptron nodes. Because of their numerous layers and non-linear activation function, MLPs can differentiate non-linearly separable data. MLPs with one hidden layer are considered a simple Neural Network while those with more hidden layers constitute a Deep Neural Network as shown in Fig 2.6.

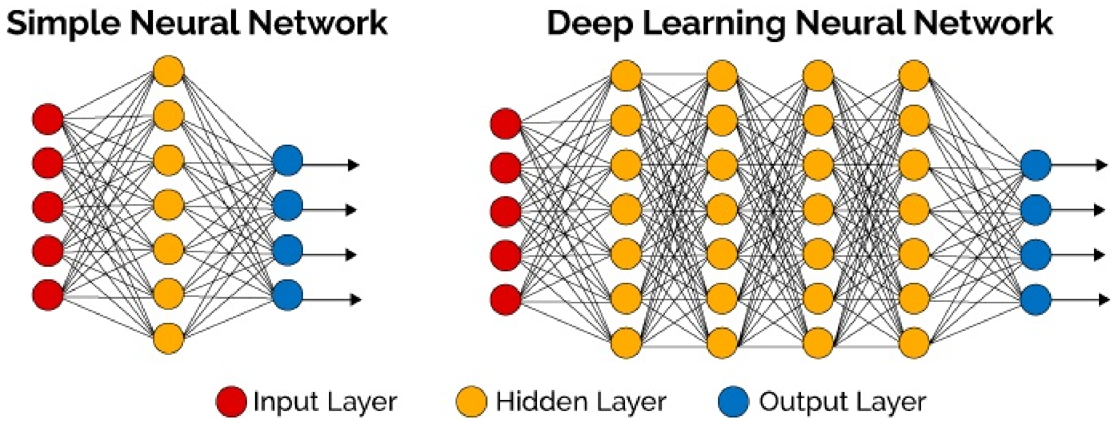


Figure 2.6: An illustration of a simple (left) and a deep (right) neural network. Source: electronicdesign

In general, feed-forward neural networks perform well in the case that the input features are independent.

2.5.2 Recurrent Neural Networks

Many times we encounter sequential data, such as time series, speech signals, and language texts, where past inputs are important for the next. Recurrent Neural Networks are a common solution for modeling this type of data. A recurrent neural network (RNN) is a type of artificial neural network proposed in the 1980s where connections between units form a directed cycle. It is recurrent as the output of every step is copied and sent back into the recurrent network and thus it is fed as input to the next step. This creates an internal network state, allowing it to exhibit dynamic behavior, and maintain information about what has been computed so far. The mechanism makes RNNs applicable to tasks that require remembering the history of previous inputs and outputs, such as natural language generation and speech recognition.

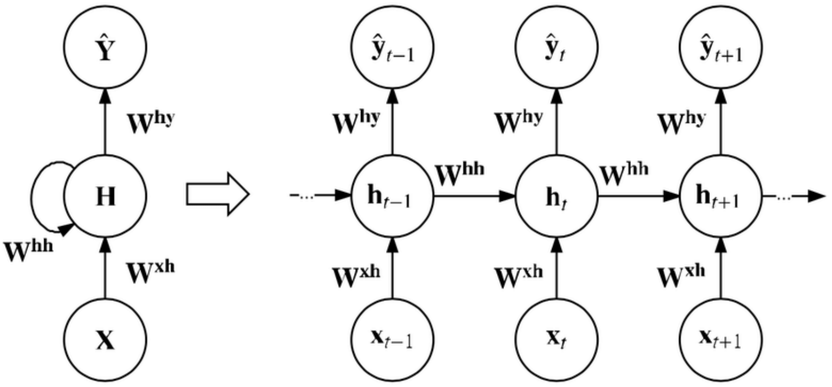


Figure 2.7: An unfolded recurrent neural network. Source: [28]

Unrolling the feed-back loop of an RNN through time, we can think of RNNs as multiple copies of the same network, one for each input point, each passing a message to a successor, in sequential order. We can see both the rolled and unrolled versions of the RNN architecture in Figure 2.7, where X_t is the input at time t, A is the internal part of an RNN cell (different for vanilla RNNs, LSTM, and GRU) and h_t is its hidden state. For vanilla RNNs the hidden state at each timestep and the output are calculated as follows:

$$\begin{aligned} \mathbf{h}_t &= f_h(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{hx}\mathbf{x}_t + \mathbf{b}_h) \\ \hat{\mathbf{y}}_t &= f_y(\mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y) \end{aligned} \tag{2.17}$$

where h_t, x_t, $\hat{y}_t$ is the hidden state, the input vector, and the output vector at time step t respectively, b_h is the bias for h, b_y is the bias for $\hat{y}$ and f_y, f_h are the activation functions for x and h respectively. W parameters are three separate matrices of weights. where W_{hx} are the input to hidden weights, W_{hh} are the hidden to hidden weights and W_{hy} are the hidden to output weights. Also, it is worth mentioning that we could construct stacks of RNN layers on top of each other by simply connecting each cell's activation,h_t at time step t, as an input to the next RNN layer at the same time step.

Long-term dependencies : While RNNs showed great promise in handling sequential data, they fall short in handling "long-term dependencies". For example, consider a language model trying to predict the next masked word based on a small sequence of words, such as *"It is a sunny summer day and the temperature is [MASK]"*. The RNNs can efficiently predict the masked word in a small sequence, but as the sequence gets longer and the number of

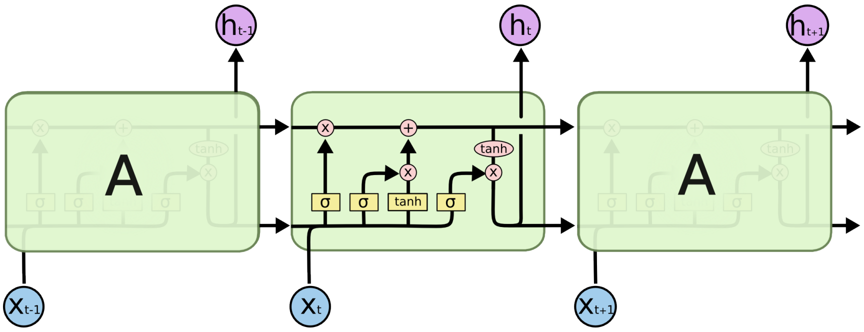


Figure 2.8: The LSTM cell. Source: colah.github.io

words after the most important one increases, it becomes almost impossible for the RNNs to remember all the previous context. The reason for this failure is the vanishing gradient problem. To overcome this problem, the Long Short-Term Memory Networks (LSTMs) were proposed by Sepp Hochreiter [23].

Long Short-Term Memory (LSTM)

Long Short-Term Memory Networks are a subcategory of Recurrent Neural Networks that overcome the problem addressed above by preserving long-term dependencies using the cell state. Adding the LSTM to the network is analogous to adding a memory unit within the network that can recall context from the start of the input. The internal architecture of an LSTM is depicted in Figure 2.8. It is composed of a cell state, an input gate, a forget gate, and an output gate. These components organize the flow of information through the cell. Given a sequence $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}, \ldots, \mathbf{x_n}$ of vectors of an input sequence of length n, for a vector $\mathbf{x_t}$, with inputs $\mathbf{h_{t-1}}$ and $\mathbf{c_{t-1}}$, the hidden-state $\mathbf{h_t}$ and cell state with $\mathbf{c_t}$ for time-step t are computed as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \\
\mathbf{u}_t &= \tanh(\mathbf{W}_u\mathbf{x}_t + \mathbf{U}_u\mathbf{h}_{t-1} + \mathbf{b}_u) \\
\mathbf{c}_t &= \mathbf{f}t \odot \mathbf{c}t - 1 + \mathbf{i}_t \odot \mathbf{u}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{2.18}$$

Forget gate ($\mathbf{f}t$) determines which information should be preserved or discarded. Information from the previous hidden state $\mathbf{h}_{t-1}$ together with information from the current input $\mathbf{x_t}$ is passed through a sigmoid activation function, which squeezes the values between 0 and 1. A value closer to 0 means to forget, while closer to 1 means to keep.

Input gate ($\mathbf{i}_t$), the previous hidden state together with the current input is passed into a sigmoid function, to squeeze the values between 0 and 1 and determine which values will be updated (0 means unimportant and 1 means important). The hidden state and current input are also passed to the tanh function to squish values between -1 and 1 ($\mathbf{u}_t$). Finally, the tanh output is multiplied with the sigmoid output ($\mathbf{i}_t \odot \mathbf{u}_t$), so that the latter will filter the important information of the former.

Cell gate ($\mathbf{c}_t$), To compute the next cell state, firstly the current cell state $\mathbf{c}_t$ gets pointwise multiplied by the result of the forget gate. This results in dropping the information from the cell state that is not that important. Then, a pointwise addition is applied between the previous result and the output from the input gate, which updates the cell state to new values that the neural network finds relevant.

Output gate ($\mathbf{o}_t$). The output gate decides what the next hidden state should be. As the hidden state contains information on previous inputs, it is also used for predictions. First, the previous hidden state and the current input are passed into a sigmoid function. Then, the newly modified cell state is passed to the tanh function. We multiply the tanh output with the sigmoid output ($\mathbf{o}_t \odot \tanh(\mathbf{c}_t)$) to decide what information the hidden state should carry. This hidden state is the output of the LSTM at each moment. The new cell state and the new hidden are then carried over to the next time step.

2.5.3 Sequence to Sequence Models

Sequence to Sequence (often abbreviated to seq2seq) models is a special class of neural network architectures that is typically used to solve NLP tasks like Machine Translation, Question Answering, creating chatbots, etc. A sequence-to-sequence model consists of an encoder-decoder architecture proposed by Cho et al. [9] as shown in Figure 2.10(a). The encoder is an RNN that takes an input sequence of tokens $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ where n is the length of the input sequence, and encodes it into fixed length hidden vectors $\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_n}$. The decoder is also an RNN which then takes a single fixed length vector $\mathbf{x_n}$ (the last hidden state of the encoder) as its input and generates an output sequence $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_{n'}}$ token by token, where n' is the length of the output sequence.

2.5.4 Attention Mechanism

In applications, recurrent neural networks often only use the final hidden state to model a sequence, to be used in a successive network. Compressing all the input information into a single fixed-length vector creates a bottleneck in the information an encoder can pass to the decoder. Because of the way the last hidden state is aggregated, the system pays more attention to the last parts of the sequence and the decoder lacks any mechanism to selectively focus on relevant input tokens while generating each output token. The attention mechanism that mitigates this problem was introduced by Bahdanau et al. in [4], initially for the task of neural machine translation, in order to deal with the need for an effective translation of very long sentences. Attention has grown in popularity in the Artificial Intelligence community as a critical component of neural networks for numerous applications in Natural Language Processing, Speech Recognition, and Computer Vision. Especially in the field of natural language processing, the attention mechanism gave rise to the transformer architecture [74], which enabled researchers to attain state-of-the-art performance in many tasks and is the foundation of today's large pre-trained language models.

The key idea of attention mechanisms is to apply attention weights α over the input sequence to prioritize the positions where relevant information is essential for generating the next output token, in a sequence-to-sequence model. The encoder-decoder architecture with the attention mechanism is shown in Figure 2.9. Using attention, we obtain a context vector c_j, which is passed as an input to the decoder. At each decoding position j, the context vector c_j is a weighted sum of all hidden states of the encoder and their corresponding attention weights α_j and contains information about the encoder's hidden state, decoder's hidden state, and the alignment between the input sequence and the target.

$$c_j = \sum_{i=1}^{n} a_{ji} h_i \tag{2.19}$$

where h_i is the encoder's hidden state for time step i and a_{ji} are the attention weights, that determine the importance of each of the encoder's outputs through time, for the calculation of the decoder's input context vector. The attention weights are calculated as:

$$a_{ji} = softmax(f(s_{j-1}, h_j)) \tag{2.20}$$

where f is an alignment function, often learned by a trainable model, such as a feed-forward neural network and scores how important is the encoder hidden state h_j for the decoder hidden state s_{j-1}. Some examples of alignment functions are shown in Table 2.1.

Name	Function
Similarity	$f(s_t, h_j) = sim(s_t, h_j)$ [18]
Additive	$f(s_t, h_j) = u_\alpha^T \tanh(W_\alpha[s_t; h_j])$ [4]
Dot-Product	$f(s_t, h_j) = s_t^T h_j$ [47]
Scaled Dot-Product	$f(s_t, h_j) = \frac{s_t^T h_j}{\sqrt{n}}$ [74]
General	$f(s_t, h_j) = s_t^T W_\alpha h_j$ [47]

Table 2.1: Summary of the most common Alignment functions

The most notable attention mechanisms that have been developed and that we will describe in more detail are the Scaled Dot-Product Attention, the Multi-Head Attention and the Self-Attention. In all of these mechanisms, the attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as the weighted sum of the values, where the weight assigned to each value is the result of the alignment function between the keys and a query and shows the keys which are relevant for the main task with respect to the query.

2.5.5 Transformers

Transformers represent a revolutionary architectural concept in the field of deep learning, leveraging attention mechanisms as their core building blocks. Introduced by Vaswani et al. in [74], transformers have been adopted in various natural language processing, computer vision and sequence modeling tasks. Transformers are designed to handle sequential data, without the need of processing the data in order, and allow for much more parallelization than RNNs during training. This enabled training on larger datasets and the development of large pre-trained transformer-based models that achieved state-of-the-art results in tasks like Machine Translation, Question Answering and Dialogue Generation.

The Transformer is a sequence-to-sequence model and consists of an encoder and a decoder, each of which is a stack of N identical layers, with different weights. The authors proposed a scaled dot-product alignment function for a self-attention mechanism. Each encoder block consists of a multi-head self-attention module and a fully connected feed-forward network, followed by a residual connection [20] and layer normalization [3]. The output of the top encoder layer is then transformed into a set of Key and Value vectors and is fed through a Cross-Attention module to the self-attention module of the decoder. This assists the decoder in focusing on the right locations in the input sequence. Furthermore, the self-attention modules in the decoder are adapted to prevent each position from attending to subsequent positions. The overall architecture of the Transformer is shown in Figure 2.9.

Scaled Dot-Product Attention

For each input vector **x** the self-attention mechanism creates a Query vector(Q), a Key vector(V) of dimension d_K, and a Value vector(V). The vectors are obtained by multiplying

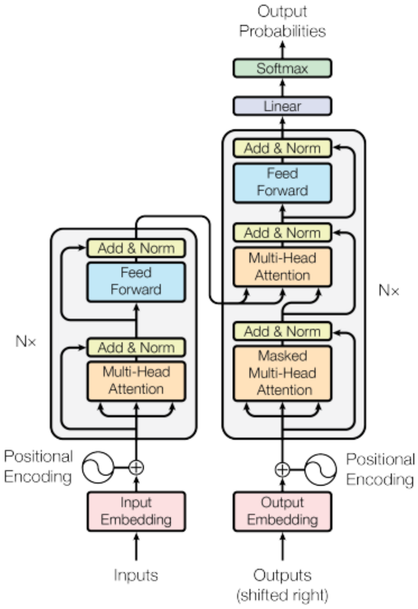


Figure 2.9: Overview of the simple Transformer architecture. Source: [74]

the input embedding with the 3 learnable matrices W_Q, W_K, and W_V. As we show in Table, the alignment function is calculated by scaling the dot-product of the K and Q vectors with the factor $\sqrt{d_K}$ and passing the result through a softmax function. To get the result of the self-attention module, the softmax function is then multiplied by the Value vector V.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \tag{2.21}$$

An illustration of scaled-dot-product self-attention is shown in Figure 2.10.

Multi-Head Attention

The multi-head attention extends the classic scaled-dot-product attention mechanism, performing attention in parallel multiple times and concatenating the output representations. The multi-head attention mechanism linearly projects the queries, keys and values N times, where N is the number of heads, with different learned linear projections and each head performs a Scaled Dot-Product Attention individually. With this procedure, separate sections of the Embedding can learn different aspects of the meanings of each word and allows the Transformer to capture richer interpretations of the sequence. Multi-Head Attention can be denoted as

$$MultiHead = Concat(head_1, \dots, head_n)W^O \tag{2.22}$$

$$head_i = Attention(QW_I^Q, KW_i^K, Vw_i^V) \tag{2.23}$$

where $W_i^Q \in \mathbb{R}^{d \times d_q}$, $W_i^K \in \mathbb{R}^{d \times d_k}$, $W_i^V \in \mathbb{R}^{d \times d_v}$ and $W_i^O \in \mathbb{R}^{d \times d_o}$ are the projection matrices for the attention head i, d is the dimension of the models hidden layer and d_q, d_k, d_v are learnable linear projections. An illustration of Multi-Head Attention is shown in the Figure 2.10.

Positional Embeddings

Since an RNN implements a loop where each word is input sequentially, it implicitly knows the position of each word. As transformers rely completely on attention mechanisms and do

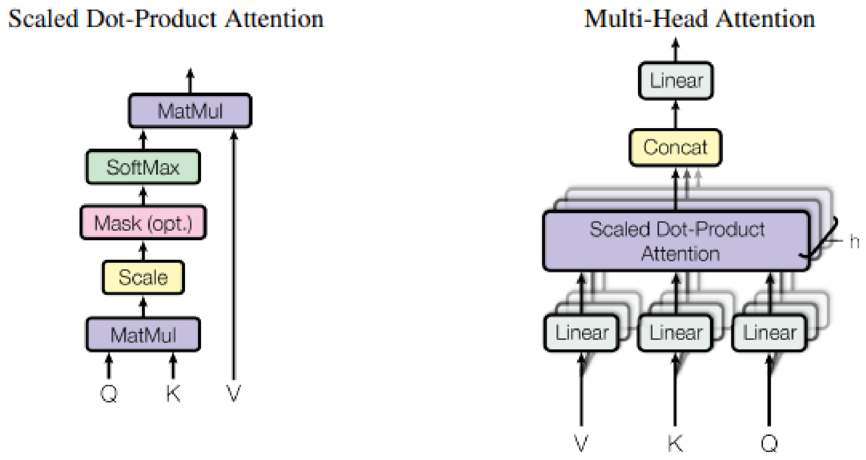


Figure 2.10: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. Source: [74]

not have inherent notions of order or position, positional embeddings provide a mechanism for encoding sequential information into the model. To achieve that, it uses positional encoding, each position in the input sequence is assigned a unique vector representation based on sine and cosine functions. The frequency and phase of the sine and cosine functions are used to encode different positions, ensuring that each position has a distinct representation. For a position pos and dimension i, the sine and cosine functions are defined as:

$$PE_{pos,2i} = sin(pos/10000^{(2i/d_{model})}) \tag{2.24}$$

$$PE_{pos,2i+1} = cos(pos/10000^{(2i/d_{model})}) \tag{2.25}$$

In summary, transformers provide numerous benefits compared to conventional recurrent or convolutional architectures. By exploiting self-attention, transformers excel at capturing long-range dependencies in sequences and generating context-aware representations by attending to relevant parts of the input sequence. Moreover, the self-attention mechanism in transformers allows for parallel processing of the input sequence, leading to significant speed improvements during training and inference. This capability facilitates the utilization of large models and extensive datasets for training purposes. With ongoing advancements and research, transformers are likely to continue pushing the boundaries of deep learning and revolutionize other domains beyond natural language processing.

2.6 Transfer Learning

Transfer learning is a machine learning technique that involves leveraging knowledge gained from solving one problem (domain task) and applying it to a different one(target task), usually relevant. An illustrated version of the transfer learning technique can be shown in Figure 2.11. In many deep learning problems, to build a model that solves a complex task, a vast amount of labeled data is required. However, collecting adequate training data can be costly, time-consuming, or even impractical in numerous situations. Instead of starting the learning process from scratch for a new task, transfer learning takes advantage of pre-existing knowledge or models trained on large datasets. [99]

Transfer learning typically involves two stages: pre-training and fine-tuning. In the pre-training stage, a model is trained on a large amount of data. This initial training helps the model learn generic features and patterns that can be useful for many different tasks. In the fine-tuning stage, the pre-trained model is further trained on a smaller dataset that is specific to the target task. Fine-tuning involves training the model on the new dataset while maintaining the initial weights fixed or modifying them according to the new task. However, it is worth noting that transferred knowledge does not always favor the new task, as it may be unsuccessful if there is little in common between the source and the target domain.

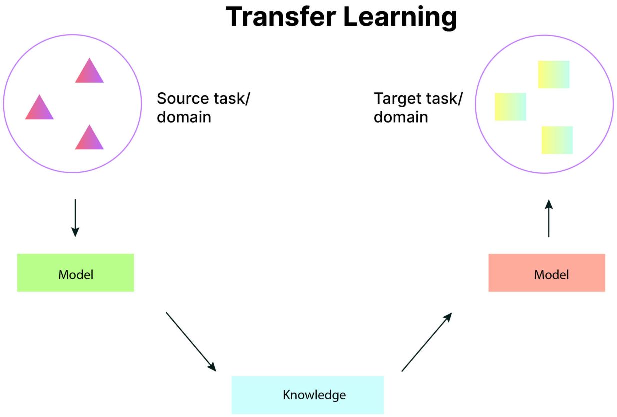


Figure 2.11: An illustration of the transfer learning technique Source: nimblebox.ai

In special cases, the domain task involves unsupervised learning. This is really common in Natural Language Processing, where we pre-train large language models with large amounts of unlabeled data. These models learn to predict the next word in a sentence and manage to capture rich contextual information. Later, these language models can be fine-tuned on downstream tasks, such as machine translation, dialogue generation and question-answering.

2.7 Multi-task learning

Multi-task learning (MTL) is a machine learning approach that involves jointly training a model on multiple related tasks. Instead of training different models for each task, Multi-task learning makes use of shared representations and knowledge across tasks with the goal of improving performance on each individual task. The key idea behind MTL is that learning from multiple tasks simultaneously can provide benefits to each individual task. There are various approaches to implementing MTL. One approach is to share lower-level layers of the model across tasks while retaining task-specific layers on top. This strategy allows the model to learn both shared representations and task-specific characteristics.

Alternatively, a single shared model can be utilized, employing multiple output heads where each head predicts the corresponding task. During training, the shared layers and the task-specific heads are optimized jointly. The shared layers are updated based on the gradients from all tasks, allowing the model to capture the common features across tasks. The task-specific heads are updated based on the gradients specific to each task, enabling them to specialize in making accurate predictions for their respective tasks. This approach reduces the model's overall complexity and memory requirements compared to training separate models for each task and often leads to better generalization as the model needs to find a common representation that improves performance across all the individual tasks. An illustration of this multitask learning approach is shown in Figure 2.12.

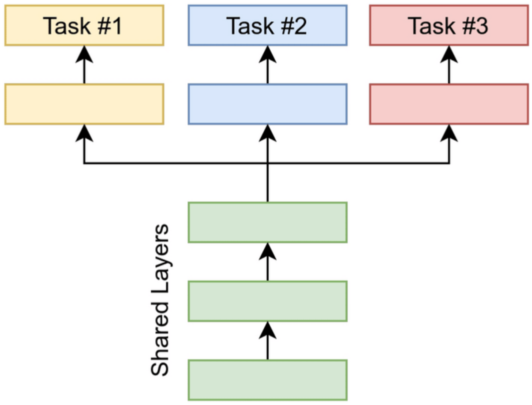


Figure 2.12: An illustration of a multi-task learning technique Source: [67]

2.8 Summary

In this chapter, we introduce the fundamental concepts and theories of machine learning and deep neural networks. These ideas are essential for this diploma thesis, as the models discussed in later sections build on these principles. Understanding recurrent neural networks, attention mechanisms, transfer learning, and multi-task learning is crucial for grasping the concepts and experiments that follow. The next chapter will cover the basics of Natural Language Processing (NLP), an important area of study for understanding dialogue systems.

Chapter 3

Natural Language Processing

3.1 Introduction

Natural language processing (NLP) is the branch of computer science—specifically, the branch of artificial intelligence or AI—concerning giving computers the ability to comprehend and understand text and spoken words in the same manner that humans do. It includes the development of algorithms and models that enable computers to understand, interpret, and generate natural language. With the increasing amount of textual data available, because of the internet and social media platforms, NLP plays a critical role in extracting important information, enabling effective communication, and automating various language-related tasks.

One of the fundamental challenges in NLP is the complexity and ambiguity of human language. Natural language is full of context-depended interpretations, idiomatic expressions and syntactic structures that pose challenges for computational systems. To tackle these difficulties NLP researchers use a variety of methods, including statistical models, machine learning algorithms, deep neural networks. These techniques help in generating coherent responses, extracting valuable information from text and achieving high levels of language understanding.

The advancements in NLP have been greatly influenced by large-scale datasets, powerful computing infrastructure, and transformer-based models like BERT [14] and GPT[59]. Transformer architectures have revolutionized various NLP tasks as they manage to capture contextual dependencies, produce language representations and generate natural language.

In this chapter, we first present the most common applications of NLP. We then explore the most common word representation methods, discuss language modeling and the creation of the most significant pre-trained language models. Finally, we will examine methods for adapting and transferring the knowledge of pre-trained models to specific NLP tasks.

3.2 Applications

Natural Language Processing (NLP) is a highly popular subject within the realm of data science. NLP finds applicability in any domain that involves working with text data. Here are some of the common applications of NLP:

1. Information Retrieval: The science of searching for documents, extracting information from within them, and retrieving metadata associated with the documents.

2. Information Extraction: This involves identifying, tagging, and extracting specific key elements from large text collections and representing them in a structured format.

3. Text Summarization: Text summarization involves generating a shorter version of one or more documents while retaining the main meaning and important information.

4. Machine Translation (MT): Machine translation utilizes computer software to translate text from one natural language to another.

5. Question Answering (QA): Question Answering is the task of extracting or generating the answer to a question from a given text.

6. Natural Language Understanding (NLU): NLU aims to understand human language and generate computer-based representations for effective analysis and interpretation.

7. Natural Language Generation (NLG): The task of generating natural language from computer-based representation, enabling machines to communicate using human-like language.

8. Dialogue Systems or Conversational Agents (CA): The task of designing computer systems that engage in conversations with humans, providing interactive and conversational experiences.

3.3 Language Modeling

Language modeling involves estimating the probability distribution over sequences of words. The objective is to construct models that assign higher probabilities to sequences that are more grammatically correct or more likely to occur. Mathematically, the probability of any given sequence of n words can be denoted as:

$$P(w_1, w_2, \ldots, w_n) \tag{3.1}$$

This probability can be calculated using the following formulation:

$$P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_{i-1}, w_{i-2}, \ldots, w_1) \tag{3.2}$$

3.3.1 Traditional Language Models

Early language models primarily relied on statistical techniques, such as n-gram models. An n-gram refers to any sequence consisting of n consecutive words. In n-gram language modeling, the word sequence is split, and one word is predicted at a time. This process can be described using the chain rule with the following equation:

$$P(w_1, \ldots, w_n) = P(w_1)P(w_2|w_1) \ldots P(w_n|w_1, \ldots, w_{n-1}) \tag{3.3}$$

To calculate the probability of a word w_i being the next word in a sequence, given a corpus C, the formula is as follows:

$$P(w_i|w_1, \ldots, w_{i-1}) = \frac{\text{count}(w_1 \ldots w_i)}{\sum_{w \in C} \text{count}(w_1 \ldots w_{i-1})} \tag{3.4}$$

The above formula is computationally expensive, and therefore, certain assumptions are made to efficiently train a language model. Instead of requiring the entire history to compute the probability $P(w_1, w_2, \ldots, w_n)$, the Markov condition is employed, assuming that the probability of a word depends only on its $n-1$ previous words. Mathematically, this can be described as:

$$P(w_M|w_1, w_2, \ldots, w_{M-1}) \approx P(w_M|w_{M-n}, \ldots, w_{M-2}, w_{M-1}) \tag{3.5}$$

While sentences can exhibit arbitrarily long dependencies, the Markov assumption is applicable for relatively small values of n and has been the dominant approach to language modeling for many decades. For the commonly used bigram and trigram models, the estimation of probabilities $P(w_2|w_1)$ and $P(w_3|w_2, w_1)$ is computed as follows:

$$P(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)} \tag{3.6}$$

$$P(w_3|w_2, w_1) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)} \tag{3.7}$$

In simpler terms, for the bigram model, the count of how often the word w_1 is followed by the word w_2 is compared to the count of other words in the training corpus. Similarly, for the trigram model, the count of how often the word sequence w_1, w_2 is followed by the word w_3 is compared to the count of other words.

The specific number of words considered in the history depends on the amount of available training data. Trigram language models are commonly used, which require a two-word history to predict the third word. Language models can also be estimated using bigrams, unigrams, or any other order of n-grams, depending on the requirements and available resources.

3.3.2 Neural Language Models

Neural language models are language models based on neural networks. Bengio proposed a neural probabilistic language model in [6]. Non-linear neural network models allow conditioning on large context sizes with only a linear increase in the number of parameters, which makes them computationally affordable. Furthermore, these models can effectively learn dense word representations, which proves helpful in addressing the curse of dimensionality.

The model tries to simultaneously learn a word vector representation space and the probability distribution for word sequences. The model takes as input vector representations, also known as word embeddings. These word embeddings, denoted as $C(w) \in \mathbb{R}^{d_w}$, represent a window of n preceding words. The embeddings are concatenated together and passed through a hidden layer. The resulting output is then fed to a softmax layer, as illustrated in Figure 3.1.

More formally, this process can be described by the following equations:

$$\begin{aligned}
&\mathbf{x} = [\mathbf{C}(w_1); \mathbf{C}(w_2); \ldots; \mathbf{C}(w_n)] \\
&\hat{y} = P(w_i|w_{1:k}) = softmax(\mathbf{h}\mathbf{W}_2 + \mathbf{b}_2) \\
&\mathbf{h} = g(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)
\end{aligned} \tag{3.8}$$

where V is the vocabulary, $w_i \in V, W_1 \in \mathbb{R}^{n \cdot d_w \times d_{hid}}, b_1 \in \mathbb{R}^{d_{hid}}, W_2 \in \mathbb{R}^{d_{hid} \times |V|}, b_2 \in \mathbb{R}^{|V|}$, and d_{hid} and d_w are the dimensions of the hidden layer and the word embedding correspondingly.

The vocabulary size $|V|$, ranges between 1,000 - 1,000,000 words, with the common size being around 70,000 unique words. In recent times, there has been a shift from employing feed-forward neural networks to adopting recurrent neural networks (RNNs) and long short-term memory (LSTM) networks for language modeling, as discussed in Section 2.5.2. Additionally, the use of transformer architectures mentioned in Section 2.5.5, has prevailed over the previous ones and is proving to be the go-to method for language modeling.

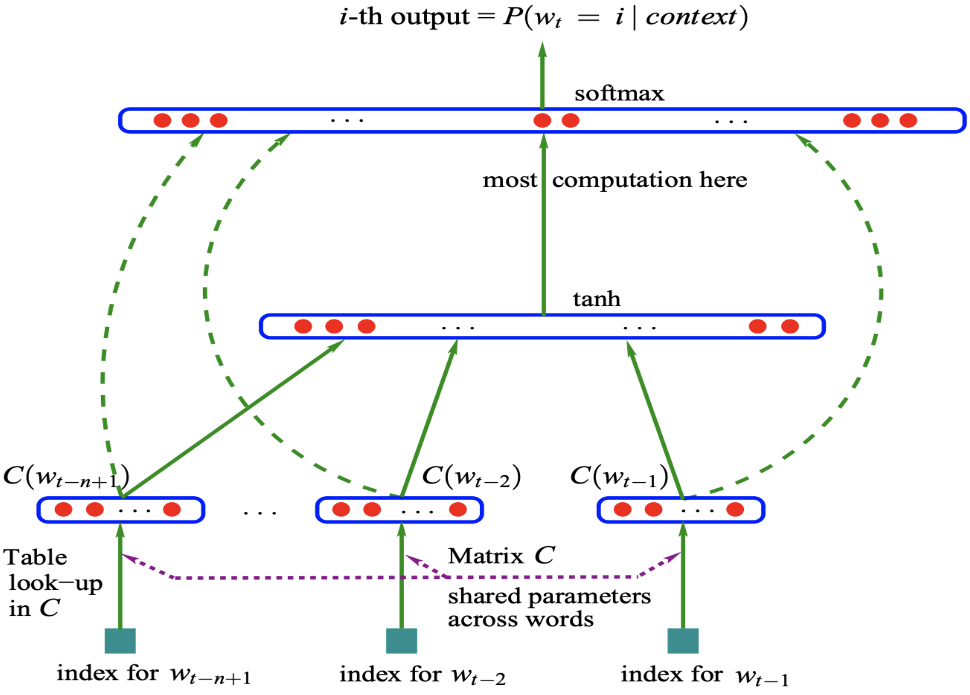


Figure 3.1: A feed-forward neural network language model. Source: [6]

3.4 Large-scale pre-trained language models

Large-scale pre-trained language models, such as GPT and BERT, have revolutionized the field of natural language processing (NLP). With the rise of deep learning, transformer architecture, and the increase of computational power, deeper model architectures with a large number of trainable parameters have emerged. These models are pre-trained on massive amounts of unlabeled corpora, allowing them to learn meaningful language patterns and relationships. Through pre-training on extensive corpora, these models learn to predict missing words, understand context, and capture semantic relationships.

Furthermore, large-scale pre-trained language models offer transfer learning capabilities, where knowledge gained during pre-training can be effectively utilized for various NLP applications. This transferability significantly reduces the need for task-specific training data and computational resources, as extensive labeled datasets are difficult to collect due to the expensive annotation costs. In this section, we will discuss the architecture and the pre-training techniques of the most important Large Language models.

3.4.1 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT), proposed by Google AI [14], is a pre-trained model of deep bidirectional transformers for language understanding, which is then fine-tuned to be used in a wide variety of language tasks, such as classification, question-answering, etc. BERT consists of a stack of transformer encoders. Each encoder block comprises multiple self-attention layers and feed-forward neural networks, augmented with residual connections and layer normalization, similar to the original transformer architecture described in 2.5.5. Two different versions of BERT were introduced by the authors. BERTbase which has 12 transformer encoder layers, 12 attention heads and a hidden dimension of 768, and BERTlarge which has 24 transformer encoder layers, 16 attention heads, and a hidden dimension of 1024. The maximum length of the input sequence is restricted to 512 tokens, and the extra tokens in the sequence are ignored. The BERTbase model contains 110 million trainable parameters, while bert-large contains 340 million.

Traditionally, training a unidirectional encoder, either as a left-right or a right-left model, was the only possible option. However, BERT is bidirectional, being able to consider both the left and right contexts of a word during the training process, which allows it to build a deep understanding of the context and generate contextually rich word representations.

This is achieved by setting a bidirectional LM task, instead of the classic Language Modeling, called **Masked Language Modeling(MLM)**. The model was trained to predict only the words that had been masked while being able to see the entire sequence. The authors randomly masked 15% of tokens in each input sequence and replaced the original token with the special token [MASK]. The model is then asked to predict only the correct words that were masked and not the whole sequence, resulting in an output size equal to 15% of the input size. However, instead of constantly replacing the selected words with a [MASK] token, it was decided the masked word to be:

- Replaced with a [MASK] token 80% of the time.
- Replaced with another random word 10% of the time.
- Left unchanged 10% of the time.

The second unsupervised task that the authors trained their model on, was the task of **Next Sentence Prediction(NSP)**. The objective of this task is to perform binary classification on whether one sentence is the next sentence of another. The dataset used for training had a balanced 50/50 distribution. Specifically, for a sentence pair (A,B) from the dataset, sentence B follows sentence A 50% of the time, and 50% of the time sentence B is a random sentence from the corpus. The input sequence for this pair classification task is created as:

$$[CLS] \ < SentenceA > \ [SEP] \ < SentenceB > \ [SEP]$$

where [CLS] token is the first token used to obtain a fixed vector representation that is consequently used for classification, and [SEP] is used to separate the two input sequences. For the 2 pre-training tasks the authors used the BooksCorpus(800M words) [98] and English Wikipedia texts(2500M words).

In addition, BERT introduces certain changes in the way the input word embeddings are created. BERT uses WordPiece embeddings, dividing the words into smaller sub-word units in order to handle words with common root or rare words more effectively, and has a total vocabulary of 30.000 tokens. Moreover, to distinguish tokens belonging to different input segments, BERT incorporates a learned embedding known as "segment embedding" into each word embedding, that denotes on which sentence a token belongs. Finally, the positional embeddings are learned rather than hard-coded as it was in the vanilla Transformer architecture.

After pre-training, BERT model can be fine-tuned in a number of downstream tasks. Fine-tuning involves training BERT on labeled data for tasks such as text classification, named entity recognition, or question-answering. During fine-tuning, the text input is transformed, in order to match the input template of BERT and the model's parameters are adjusted to adapt its learned representations to the target task. BERT's fine-tuning can be performed effectively using much smaller datasets than those used in the pre-training stage and it demands less computation resources. The way model is finetuned on these tasks is shown in Figure 3.3.

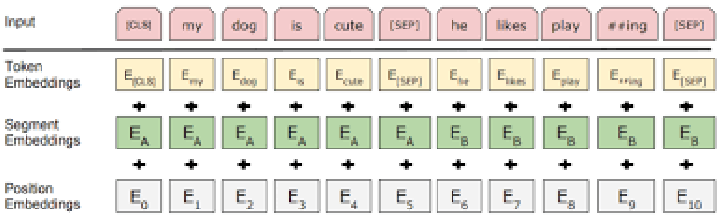
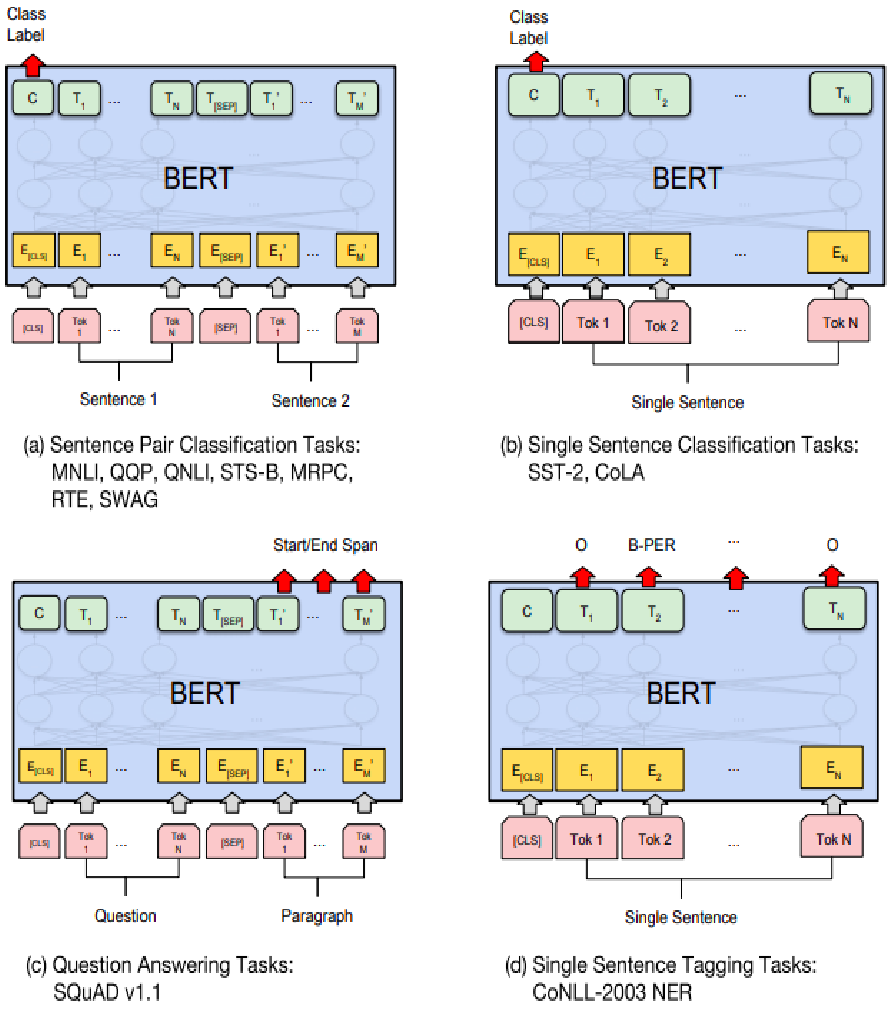

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
Segment Embeddings
Position Embeddings
E[CLS] E_my E_dog E_is E_cute E[SEP] E_he E_likes E_play E_##ing E[SEP]
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_10

Figure 3.2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings. Source: [14]



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Figure 3.3: Illustrations of Fine-tuning BERT on Different Tasks. Source: [14]

At the time of its release, BERT found significant success in multiple NLP tasks and achieved state-of-the-art results. Subsequently, a family of language models were developed by adapting the BERT to different languages, or modifying its architecture. The most known are, mBERT [14] which is a multilingual model that covers 102 languages, trained on the multilingual version of the Wikipedia dataset, RoBERTa [46] which is a retraining of BERT with optimized training methodology and much more data, and GREEK-BERT which we will discuss in the next subsection.

GREEK-BERT

GREEK-BERT [30] is a variant of the BERT model specifically designed for the Greek language. The authors used the BERTbase's architecture and pre-trained the model on the tasks of Masked Language Modeling and Next-Sentence Prediction. For the pre-training a total of 29GB of Greek text used from the following corpora:

- the Greek part of Wikipedia.
- the Greek part of the European Parliament Proceedings Parallel Corpus.
- the Greek part of OSCAR, a clean version of Common Crawl.

The fine-tuning process adapted the model to perform well on 3 Greek language tasks, Part-of-Speech tagging, Named Entity Recognition, and Natural Language Inference. On NER and NLI tasks, GREEK-BERT outperformed the multilingual models mBERT and XLM-R [11], whereas on the PoS tagging task, GREEK-BERT had similar results to XLM-R.

3.4.2 Generative Pretrained Transformer 2 (GPT-2)

Developed by OpenAI, GPT-2 [60] is a language model that has gathered attention and recognition from many researchers and has been used in a wide variety of tasks, achieving state-of-the-art results. The model's architecture is simple and it is very similar to the transformer's decoder architecture. GPT-2 is a decoder-only model and is built by stacking decoder blocks. GPT-2 is an auto-regressive model meaning that it outputs one token at a time and adds that token to the sequence of inputs. In the next step, that new sequence is fed as input into the model.

The GPT-2 model is available in 4 different versions, each varying in dimensionality and number of decoder layers: small, medium, large, and extra-large. The different dimensions of these versions are 768, 1024, 1280, and 1600, respectively. Additionally, the number of decoder layers in the decoder stack varies across the different versions, with 12, 24, 36, and 48 decoder layers in the respective versions. For the pre-training of the model, a large-scale, unsupervised strategy was adopted. Through the pre-training, GPT-2 learns to predict the next word in a sentence given the preceding context. The model was trained using the WebText dataset, which contains slightly more than 8 million documents totaling 40 GB of text. This extensive pre-training allows GPT-2 to develop a deep understanding of language structures, grammar, and semantic relationships.

Now, we will focus on the decoder layer of the GPT-2 model where most of the work is done. The key difference between BERT and GPT-2 is that the latter uses masked multi-head self-attention. Masked self-attention attention works by blocking information from tokens that are to the right of the position being calculated. An illustration of the simple self-attention, used in BERT, and the masked self-attention is shown in Figure 3.4. Moreover, each attention head in GPT-2 is responsible for attending to a different part of the input text. For example, one head might focus on the syntax of the sentence, while another might focus on the semantics. This way, the model can focus on different aspects of the input text, and it is able to generate more accurate predictions and better understand the context of the sentence.

Initially, OpenAI limited GPT-2 model's availability; however, they later made smaller versions accessible to the research community. This decision enabled researchers to conduct experiments across a variety of NLP applications, including text classification, summarization, and question-answering. Notably, GPT-2 demonstrated remarkable adaptability to these

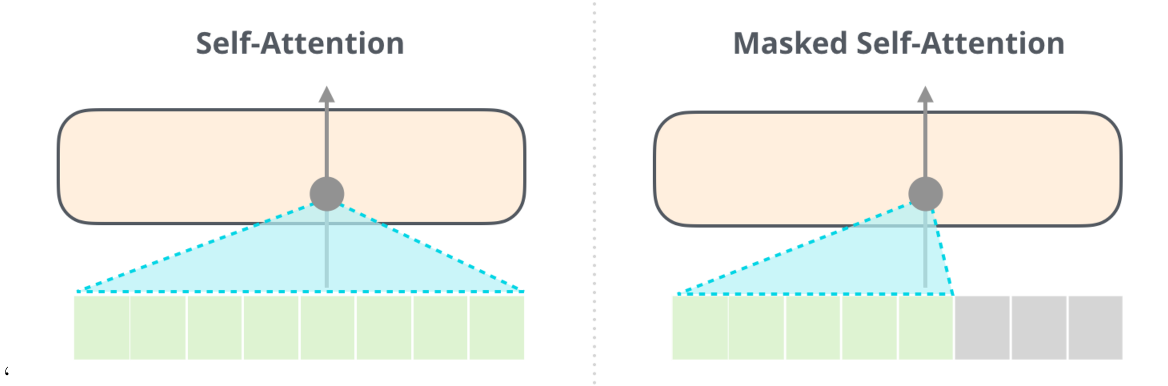


Figure 3.4: Left: Bert's self-attention mechanism. Right: GPT's masked-attention mechanism. Source: The Illustrated GPT-2

downstream tasks following successful fine-tuning. Fine-tuning involves training GPT-2 on a labeled dataset that is tailored to the specific task the model is intended to perform. Through this process, GPT-2 can acquire task-specific knowledge and improve its performance accordingly.

GPT-2 Greek

The open-source versions of GPT-2 made possible the creation of many family models like DialoGPT[91] and GPT-2 Greek [38], which is an adaptation of the GPT-2 language model specifically designed for the Greek language. GPT-2 Greek model is built by fine-tuning the original GPT-2 small model with gradual layer unfreezing. This is a more efficient and sustainable alternative compared to training from scratch, especially for low-resource languages, as it optimizes the performance of the GPT-2 Greek model, allowing it to better comprehend and generate Greek text. To create the GPT-2 Greek model, the authors utilized a 23.4GB sample from a consolidated Greek corpus from CC100, Wikimatrix, Tatoeba, Books, SETIMES, and GlobalVoices containing long sentences.

The availability of GPT-2 Greek opens up new opportunities for researchers, developers, and organizations working with the Greek language. It enables them to use the power of a language model that understands and generates Greek text and helps them create a variety of NLP applications for the Greek language.

3.4.3 Text-To-Text Transfer Transformer (T5)

Text-to-Text Transfer Transformer (T5) was introduced in 2019 by researchers at Google [61] and represents a significant advancement in transfer learning for NLP tasks. The authors' main idea is to approach every text-processing problem as a text-to-text problem. By adopting this text-to-text framework, it becomes possible to directly apply the same model, objective, training procedure, and decoding process to a wide range of NLP tasks.

At its core, T5 is built upon the encoder-decoder framework of the original transformer architecture, described in 2.5.5, which has proven highly effective in capturing contextual information and relationships in sequences. In the T5-base model, both the encoder and decoder stacks are composed of 12 layers. Each layer includes feed-forward networks that consist of a dense layer with an output dimensionality of 3072, followed by a ReLU non-linear activation function and another dense layer. All attention mechanisms in T5-base utilize 12 heads, while the sub-layers and embeddings have a dimensionality of 768. The T5-base model has a total of around 220 million parameters.

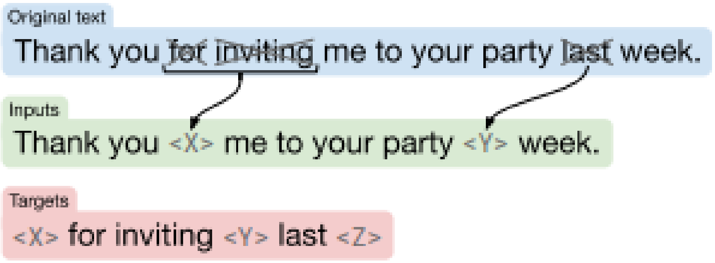

Original text
Thank you ~~for~~ ~~inviting~~ me to your party ~~last~~ week.
Inputs
Thank you <X> me to your party <Y> week.
Targets
<X> for inviting <Y> last <Z>

Figure 3.5: The words "for", "inviting" and "last" (marked with an ×) are randomly chosen for corruption. Each consecutive span of corrupted tokens is replaced by a sentinel token (shown as <X> and <Y>) that is unique over the example. Since "for" and "inviting" occur consecutively, they are replaced by a single sentinel <X>. The output sequence then consists of the dropped-out spans, delimited by the sentinel tokens used to replace them in the input plus a final sentinel token <Z>. Source: [61]

For effective transfer learning, the model needs to be trained on a massive high-quality dataset, during the pre-training phase. To satisfy these requirements the authors developed the Colossal Clean Crawled Corpus (C4), a cleaned version of Common Crawl that is 2 times bigger in size than the Wikipedia. T5 utilizes a modified version of the masked language modeling task, for the pretraining process, known as corruption span. As opposed to BERT, which uses a mask token for each word, T5 replaces many consecutive tokens with a single mask keyword. Since the final goal is to have trained a model that inputs text and outputs text, the targets were designed to produce a sequence as shown in Fig 3.5.

The authors tested 3 different corruption strategies:

- Masking a random word, like BERT
- Masking more than one consecutive word (a span)
- Dropping a word from the input

and corrupting the span was the method that worked best for them. Furthermore, the researchers conducted experiments involving various lengths of corruption spans. They discovered that as the span length increased, the model's performance decreased. This finding aligns with expectations since if the span length were equal to the length of the sentence, the model would essentially be generating text from an empty input, allowing for a high level of variability.

Furthermore, during pretraining the researchers trained the model on some supervised tasks, using multitask learning. One crucial aspect in this stage of training was the insertion of a task-specific prefix-text in the input sequence before encoding. This prefix helped specify the desired task for the model to perform. For instance, to ask the model to perform the task of translation from English to German, a **translate English to German:** prefix-text was added. By incorporating these task-specific prefixes, the model could adjust its weights to focus on the specific task. This narrowing of the generation scope ensured that the model would only produce the expected output for the designated task, enhancing its performance and task specificity.

In conclusion, the pretraining phase serves as a crucial step in equipping T5 with the foundational knowledge necessary for subsequent fine-tuning on specific tasks, enabling it to perform at state-of-the-art levels in a wide range of NLP applications. After its success, Google released some follow-up works like T5.1.1 which is an improved version of T5 with some architectural tweaks and is pre-trained on C4 only without mixing in the supervised tasks, mT5 which is a multilingual T5 model, UL2 which is a T5 like model pre-trained on various denoising objectives, and Flan-T5 which used pretraining methods based on prompting.

Multilingual Text-To-Text Transfer Transformer (mT5)

The model architecture and training procedure that the authors used for mT5 [85] closely follow that of the T5 recipe. Similar to T5, mT5 also introduced five different variants, namely small, base, large, xl, and xxl, with varying numbers of parameters ranging from 300 million to 13 billion. However, there is a difference in the pretraining process of mT5 compared to the original T5 model. The mT5 model was only pre-trained on mC4 excluding any supervised training. Therefore, this model has to be fine-tuned before it is usable on a downstream task, unlike the original T5 model. Since mT5 was pre-trained unsupervisedly, there's no real advantage to using a task prefix during single-task fine-tuning, and only if you are doing multi-task fine-tuning, a prefix is needed. The mT5 model has achieved state-of-the-art performance on various multilingual NLP benchmarks and tasks. Its ability to handle multiple languages within a unified model, leverage cross-lingual transfer learning, and facilitate fine-tuning for specific languages and tasks makes it a valuable tool for multilingual text understanding and generation.

3.4.4 XGLM

Large-scale autoregressive language models such as GPT-3 can be adapted, via few- and zero-shot learning, to a wide range of tasks with significantly less cost than full fine-tuning. While these models are known to be able to jointly represent many different languages, their training data is dominated by English, potentially limiting their cross-lingual generalization. To address this, Meta AI introduces the XGLM model in [39], a multilingual autoregressive language model inspired by GPT-3 [8]. XGLM is trained on a balanced corpus covering a diverse set of languages and aims to explore the multilingual few- and zero-shot learning capabilities in a wide range of tasks. It was one of the first multilingual autoregressive models that were created and has an architecture similar to GPT-3 and GPT-2, mentioned in 3.4.2. Four different variants were created with 564M, 1.7B, 2.9B, and 7.5B parameters respectively. Due to the fact that the authors processed all languages using a combined vocabulary of 250k tokens, XGLM's variants have more parameters than the comparable variants of the GPT-3 model.

To collect their pretraining data, the researchers extend the pipeline used for mining the CC100 corpus, to generate the CC100-XL, a multilingual dataset spanning 68 monthly snapshots of the Common Crawl. In total, pretraining data includes 30 languages covering 16 language families. However, due to an imbalanced distribution with English tokens being six times more abundant than the second-largest language, the authors up-sampled the medium and low-resource languages to create a more balanced language distribution. During the pretraining phase, the model focused on the unsupervised task of language modeling, aiming to predict the next word.

The authors tested the model on a variety of downstream tasks, mainly using few-shot learning. In these low-resource scenarios, the performance of the model heavily depends on the prompt construction. This problem is further complicated in the multilingual setting, where it is necessary to find the optimal prompts for examples in different languages. Three different approaches were explored for obtaining prompts in non-English tasks. The first approach involved native speakers of the target language crafting the prompts. The second approach entailed translating from English prompts, leveraging the ease of constructing high-quality prompts in English. The final approach was cross-lingual prompting, where prompts in English (or another high-resource language) were directly applied to non-English examples, capitalizing on the model's cross-lingual capabilities after being trained on a diverse language set. While handcrafted and translated prompts achieved better results across various tasks, cross-lingual prompting demonstrated competitive performance, particularly in low-resource languages where creating effective prompts was more challenging.

In conclusion, XGLM model achieved state-of-the-art results for few-shot learning in more than 20 languages (including mid- and low-resource languages) on commonsense reasoning, NLI, and machine translation tasks. A key factor for XGLM's performance was its robust cross-lingual capabilities. The model showcased a deep understanding of multiple languages, allowing it to transfer knowledge and effectively learn from examples in non-English languages.

3.5 Prompt-Based Learning

Prompt-based learning has emerged as a promising methodology for adapting an LLM to a specific downstream task [43]. Prompt-based learning addresses the problems of the fine-tuning method, which requires the training of millions or billions of parameters and often needs a large amount of good-quality data to properly train the model in a downstream task. Instead of altering the parameters of the pre-trained model, prompt-based learning introduces an additional set of parameters known as a prompt into the model's input[43], [33]. These prompts can be included in one of two ways: as a series of prompt tokens within the text input or as prompt embeddings directly in the embedding space. Using prompts, a set of parameters θ_p is added to the parameters θ of the model. Thus, the model now calculates the probability:

$$P_{\theta;\theta_p}(Y|X;P) \tag{3.9}$$

where P are the prompt tokens, parameterized by θ_p, which are passed into the model as additional input.

3.5.1 Creating prompts

As mentioned, prompts are constructed from a collection of prompt tokens. There are two approaches to determining these tokens: manual selection and automatic learning [8]. In manual determination, predefined prompts are used, typically designed to align with human understanding of the task. Alternatively, automatic learning methods can be employed to determine the tokens, utilizing various techniques to identify the most suitable prompts.

Discrete prompts

In the context of prompt-based learning, a discrete prompt refers to a fixed set of predefined instructions or task descriptions provided to a language model. Discrete or hard prompts consist of tokens that are directly mapped to an existing word in the model's vocabulary [92],[26]. In that way, the prompt parameters θ_p are a subset of the pre-trained language model's word embedding parameters θ_{emb}, and no extra parameters are added to the model $\theta_p \subseteq \theta_{emb}$. Authors in [13], proposed an easy way to optimize discrete prompts through reinforcement learning to find the suitable prompts for your task.

The advantage of using discrete prompts is the level of control they provide over the AI system's behavior. By explicitly defining the instructions, researchers can shape the generated responses to meet their specific requirements. Discrete prompts have been utilized in various natural language tasks. For instance, in classification tasks, researchers have created prompts that define the desired classification task and guide the model to produce the appropriate class labels. Discrete prompts have also been employed in text generation tasks, where researchers aim to generate coherent and contextually appropriate responses. By providing explicit prompts, researchers guide the system's response generation, ensuring the production of text that is both coherent and of high quality.

It is worth noting that despite the advantages discrete prompts can offer, they may have some limitations. Designing discrete prompts requires an understanding of the model's inner workings. Additionally, prompts that may seem reasonable to humans may not necessarily be effective for language models. The choice of prompts can significantly impact the performance of pre-trained language models, as they are sensitive to this selection process.

Soft prompts

Soft prompts, also known as continuous prompts, offer a distinct approach compared to discrete prompts. Rather than providing explicit instructions, soft prompts are characterized by continuous parameters that can be optimized through back-propagation in the embedding space [33],[45]. This makes them easily trainable and adaptable to specific tasks and language models. Unlike hard prompts, the embeddings of soft prompts do not correspond to specific words in the model's vocabulary.

One advantage of soft prompts is their ability to handle more complex and ambiguous tasks. Unlike discrete prompts, which provide rigid instructions, soft prompts allow the AI system to consider a broader range of information and adapt its response generation accordingly. This flexibility makes soft prompts suitable for tasks that require more context-aware or context-dependent responses. his flexibility makes soft prompts well-suited for tasks that require context-aware or context-dependent responses. However, it is important to note that while soft prompts provide flexibility, they do not replace the importance of effective prompt engineering

The initialization of soft prompts can be performed in different ways: The simplest approach is a random initialization of the prompt parameters. However, over the years, other methods have been proposed as well. For instance, soft prompts can be initialized with embeddings of random words from the language model's vocabulary or with other pre-trained embeddings.

Mixture of hard and soft prompts

Rather than choosing exclusively between discrete prompts and soft prompts, researchers have proposed a hybrid approach that combines the benefits of both types [45],[19]. This approach involves utilizing tunable soft prompts, which can have their parameters optimized to improve performance, while also incorporating hard prompt tokens that directly correspond to words in the model's vocabulary. These hard prompt tokens are specifically determined for each task, aiming to further improve performance and provide more explicit guidance to the language model. In that way, we combine the flexibility and adaptability of soft prompts, with the precision and specificity of hard prompts.

3.5.2 Training and prompt-based learning

Prompt-based learning is often utilized as a more lightweight option compared to fine-tuning. However, prompts can be utilized alongside fine-tuning to enhance performance. In the following paragraphs, we present these 2 alternatives.

Alternative to finetuning

As previously stated, in prompt-based learning, the pre-trained model parameters are typically kept frozen, especially for large pre-trained language models and only the prompt parameters θ_p are optimized. The optimization process can involve back-propagation, particularly when working with soft prompts, or other manual and automatic methods such as reinforcement learning for discrete prompts. By training solely the prompt parameters, prompt-based learning offers a more efficient alternative to fine-tuning in terms of computational resources and storage requirements.

Prompt-based learning proves to be highly advantageous in scenarios where the available data for a given task are limited. This is because the pre-trained model's parameters remain unchanged, preserving the language understanding capabilities acquired during the pre-training phase. Consequently, prompt learning solely guides the model towards the specific task without impacting its underlying comprehension and generation abilities.

Supplementary to finetuning

While it is common practice to keep the parameters of the pre-trained language model frozen, especially when dealing with extremely large pre-trained models, this is not always the case. Certain researchers use prompts as supplemental information to improve performance while also fine-tuning some or all of the model's parameters [45], [5]. This is especially common when working with smaller models because then fine-tuning requires fewer resources and space. The decision to freeze or adjust particular parameters is ultimately determined by a number of criteria, including the specific task, the availability of data, and the available resources.

3.5.3 Previous work using prompts

In the field of prompt-based learning using soft prompts, many different variations have been proposed in recent years. Lester et al. [33], use a sequence of prompts that lie directly in the embedding space of the model and are concatenated with the word embeddings that are produced from the text input that is given to the model. From their results, they observe that their method can be very effective for billion parameter models, but lacks in performance in comparison to finetuning, when the language model is smaller (i.e. 100 million parameters). For this reason, transfer learning was later proposed by Vu et al. [78]: Prompts were first trained on different tasks similar to the downstream task or tasks that involve high-level reasoning about semantic relationships among sentences. The pre-trained prompts were then used to initialize the prompts for the target task. As the authors observe, this method can allow prompt-based learning to be effective even for smaller-scale models.

Other researchers have explored a modified approach to prompt-based learning by incorporating prompts not only at the input layer but also deeper within the model. For instance, Li and Liang [36] used a prefix-tuning method. They employed prefix activations added to every layer in the encoder of the model, including the input layer, particularly for language generation tasks. They discovered that initializing with task-relevant words improves generation performance. In a similar vein, Liu and colleagues [44] integrated prompts at various layers of the pre-trained model, placing them as prefixes. They expanded upon Li and Liang's method for tasks related to natural language understanding (NLU). Furthermore, they found that, in situations with complete data, both a language modeling head and a randomly initialized classification head can be used for predicting the final classification labels in prompt-based learning.

As previously discussed, some researchers have advocated for a hybrid approach that combines both hard and soft prompts, striving to achieve optimal performance. One notable example of this is presented by Liu and colleagues in their work on "P-tuning" [45]. In this approach, continuous prompts are employed, which are generated by a prompt encoder that is trainable. The key function of the prompt encoder is to model the relationship between prompt embeddings and mitigate the risk of getting stuck in local minima. This encoder is structured with a bidirectional LSTM, followed by a RELU-activated MLP (Multi-Layer Perceptron). Additionally, within the prompt template, Liu and his team incorporated task-specific anchor tokens. These tokens, such as "?" for tasks like Recognizing Textual Entailment (RTE), help tailor the prompts to the particular task, enhancing the model's ability to provide precise and relevant responses. This combination of techniques demonstrates an innovative approach to prompt-based learning, effectively leveraging both hard and soft prompts to achieve improved performance across various tasks.

3.6 Summary

In this chapter, we studied the basic principles of the Natural Language Processing (NLP) research field, emphasizing the computational techniques that enable machines to comprehend and generate human language. The discussion begins with an introduction to the diverse applications of NLP, ranging from text summarization to conversational systems, which underscores the field's vast scope and potential. We then explore language modeling, detailing the evolution from traditional n-gram models to sophisticated neural network approaches that allow for efficient management of larger contexts. This sets the stage for an in-depth look at transformative pre-trained language models like BERT and GPT-2, highlighting their architectures, training methods, and pivotal role in advancing NLP tasks through contextual understanding. The chapter culminates with an examination of prompt-based learning, a novel approach that leverages the capabilities of these large models to perform specific tasks efficiently, without the need for extensive retraining. This progression from basic concepts to cutting-edge technologies not only provides a foundational understanding of NLP but also prepares the ground for addressing the challenges associated with dialogue generation in underrepresented languages such as Greek.

Chapter 4

Open-Domain Dialogue Generation for Low-Resource Languages

4.1 Introduction

Natural language holds immense significance in human civilization as it evolved to facilitate coexistence, communication, and social evolution. Dialogue, being an important part of language, connects humans through conversations. Whether with family, friends, or in business settings, we all utilize this form of language in our daily lives. In the realm of artificial intelligence, dialogue systems have emerged as a challenging field that enables communication between conversational agents and humans through natural language. These automated conversational agents are designed to imitate human behavior during conversations.

Conversational agents fall into two main categories: task-oriented and non-task-oriented agents. Task-oriented agents are designed for specific tasks and engage in short conversations with users. Their primary objective is to assist users in completing particular tasks by gathering information and providing relevant responses. We frequently encounter task-oriented dialogue systems in various daily life services, such as booking, traveling, shopping, or food ordering applications.

On the other hand, open-domain dialogue systems represent a fascinating variation. Unlike task-oriented systems, their focus is on engaging in free-flowing, unrestricted conversations with users. These chatbots, also known as open-domain dialogue systems, possess the capability to understand natural language inputs and generate human-like responses. They aim to emulate human-like conversation by incorporating techniques like language understanding, context retention, and context-aware generation. Generative systems can produce flexible and dialogue context-related responses while sometimes they lack coherence and tend to make dull responses. Retrieval-based systems select responses from human response sets and thus are able to achieve better coherence in surface-level language. However, retrieval systems are restricted by the finiteness of the response sets and sometimes the responses retrieved show a weak correlation with the dialogue context.

The earliest examples of chatbots include ELIZA [79], a system based only on simple text parsing rules that managed to convincingly mimic a Rogerian psychotherapist by persistently rephrasing statements or asking questions, and PARRY [10], which managed to mimic the pathological behavior of a paranoid patient to the extent that clinicians could not distinguish it from real patients. These models relied on rules and patterns rather than data for learning. However, more recent methods have leveraged data-driven approaches, enabling chatbots to learn from vast amounts of conversations between humans, such as those on chat platforms, Twitter, or in movie dialogues. These methods can be broadly categorized as retrieval-based or generation-based. Retrieval-based chatbots are trained to select the most suitable response from a predefined database of responses [82]. In contrast, generation-based systems aim to generate responses word-by-word, drawing from a probability distribution over the vocabulary used [68]. This allows them to create more diverse and contextually relevant responses.

Dialogue systems can exist for both voice and text modalities. While we discuss dialogue systems for the text modality in this thesis, spoken dialogue systems are equally popular in industries and are a very active research area in academia. By incorporating additional audio features such as acoustic cues, spoken dialogue systems might be extremely useful in situations where the user might be visually challenged or have difficulty writing.

Existing powerful dialogue models have been typically pre-trained on a significant number of English dialogue sessions extracted from either social media (e.g. Reddit and Twitter) or web documents. However, dialogue systems for many other languages have long been underexplored. In the following sections, we analyze some of the most important generation-based approaches for creating an open-domain dialogue generation system and later present techniques for dialogue generation on a low-resource language.

4.2 Dialogue Generation

Before we study dialogue systems conversing with humans, it is crucial to understand dialogues and their properties, to understand better how humans converse with each other. Typically, a dialogue is a sequence of utterances often regarded as turns between two or more parties in the literature. Each utterance (turn) is a single contribution from one speaker to the dialogue. At the very least, a dialogue should involve more than one person, so usually, when someone asks a question, they expect a response. Essentially a dialogue system seeks to comprehend the user's utterance (question, request, statement, etc.) and attempts to generate suitable and coherent responses while trying to use its memory and reasoning over the context.

Dialogue generation models are typically built on the sequence-to-sequence (seq2seq) model, an encoder-decoder architecture where both the encoder and decoder can be either recurrent neural networks (RNNs) or Transformers with self-attention blocks, while lately autoregressive transformer, decoder only, models have found significant success as the go-to method of building a dialogue generation system. Let the input sequence be $X = (x_1, x_2, \ldots, x_T)$ termed context and the output sequence be $Y = (y_1, y_2, \ldots, y_{T'})$ termed response, the learning objective of the task is to maximize the generation probability of response conditioned on context:

$$p(Y|X) = \prod_{t'=1}^{T'} p(y_{t'}|y_1, y_2, \ldots, y_{t'-1}, X) \tag{4.1}$$

where $p(y_{t'}|y_1, y_2, \ldots, y_{t'-1}, X)$ denotes the conditional probability of $y_{t'}$ given context X and its prior words in response Y.

The basic idea behind generation-based methods is to synthesize a new sentence word by word as a response to the user's request. Traditionally conversational agents are built using the sequence-to-sequence (seq2seq) architecture [76] and were inspired by the work in machine translation. However, the task of response generation in dialogue settings is a bit different from machine translation, as in machine translation words or phrases in the source and target sentences tend to align well with each other, but in conversation, a user utterance may share no words or phrases with a coherent response. The sequence-to-sequence architecture consists of an encoder model that encodes the user input and represents it as a vector, and a decoder model that decodes the vector (representation of encoded input) and generates a sentence word by word. The first conversational agents created that way used recurrent neural network (RNN) as encoder and decoder models respectively. Engaging with these agents, however, leads to short conversations [75] as the responses produced are dull and generic as the generated response is based on the previous turn while the huge amount of information derived from previous turns of the dialogue is partially ignored. To overcome this problem and to incorporate dialogue history in response generation the hierarchical recurrent encoder-decoder (HRED) architecture was adopted, allowing the model to summarize information over multiple prior turns [70], [65]. Subsequently, [66] proposed a Latent Variable Hierarchical Recurrent Encoder–Decoder (VHRED) to model complex dependencies between sequences. Based on HRED, VHRED combined a latent variable into the decoder and turned the decoding process into a two-step generation process: sampling a latent variable at the first step and then generating the response conditionally. VHRED was trained with a variational lower bound on the log-likelihood and exhibited promising improvement in diversity, length, and quality of generated responses. The introduction of memory networks allowed the researchers to create models able to condition responses on both dialogue history and external knowledge. However, these architectures could not produce continuous and coherent responses across multiple turns. Also, some researchers adopted reinforcement learning in a try to train a model to generate more natural responses [64].

Subsequently, the transformative architecture of Transformers, as discussed in the previous section, emerged and revolutionized the field of Natural Language Processing. This innovation led to enhancements in answer quality while reducing computational costs, enabling the development of larger models capable of capturing more extended dependencies. The introduction of self-attention architecture, coupled with the "Pretrain, then fine-tune" paradigm, has significantly influenced dialogue generation. Many recent open-domain dialogue systems now rely on extensive transformer structures and pre-trained datasets.

There have been some noteworthy developments in relation to knowledge-aware systems. The authors in [93] built a knowledge-grounded dialogue system in a synthesized fashion. Authors used both BERT and GPT-2 to perform knowledge selection and response generation jointly, where BERT was for knowledge selection and GPT-2 generated responses based on dialogue context and the selected knowledge. Researchers in [16] addressed the issue of factually inaccurate responses and hallucination using a generate-then-refine strategy, where generated responses are corrected using a knowledge graph.

Significant research efforts have been dedicated to exploring emotion [84], [81] and empathy within dialogue systems. One notable example is Know-EDG [35], which features a knowledge-enhanced context encoder and an emotion identifier linear layer integrated into a transformer model. The emotion identifier allows the model to adapt its responses based on the emotions expressed by its dialogue partner. Beyond providing engaging responses, the ability to comprehend the situation and generate appropriate emotional responses is also a desirable trait.

In the realm of open-domain chatbots, with a broader focus on dialogue characteristics, Meena [1] stands out. Meena is a transformer-based seq2seq model trained on substantial volumes of real chat data, designed to engage in natural and open-ended conversations with users, aspiring to pass the Turing Test. Training data for Meena comprises context-response pairs, where the context encompasses the last few turns, up to a maximum of 7. Meena analyzes previous turns to predict responses, similar to how BERT learns by comparing actual and predicted words. Subsequently, Facebook researchers introduced BlenderBot [63], highlighting that chatbot performance relies on more than just parameter scalability. Effective conversation necessitates a blend of skills, including offering engaging talking points, active listening, demonstrating knowledge, empathy, and appropriate personality expression, all while maintaining a consistent persona. BlenderBot achieved state-of-the-art results in terms of engagement and human-like qualities, as evaluated by human judges.

Furthermore, notable progress has been observed in recent years based on the pre-trained GPT-2 language model ([50], [86], [91],[94]). In 2020, researchers in [91] introduced DialoGPT, which treats multi-turn dialogues as long text and frames the generation task as language modeling using a vast dataset sourced from Reddit. DialoGPT is specifically designed to generate human-like text in a conversational context, making it well-suited for chatbot applications. It achieved state-of-the-art performance across various conversational tasks and was one of the first models capable of handling multi-turn dialogues effectively.

Subsequently, Google unveiled LaMDA [72], an improved iteration of Meena, based on seq2seq architecture. Researchers demonstrated that scaling models through pretraining indeed enhanced their capabilities. However, to address metrics such as bias and groundedness, and target more real-world subjects fine-tuning was necessary on domain-specific datasets. They also noted that combining fine-tuning with prompting further improved all metrics, ultimately achieving state-of-the-art results.

In recent years, large language models (LLMs), which function as open-domain dialogue systems, have gained significant prominence. These models primarily utilize a transformer-decoder architecture. A notable milestone in this evolution was the introduction of the ChatGPT model [55], which was soon followed by various other models that cater to diverse languages and specialized domains such as coding and healthcare. The defining features of these newer models compared to earlier versions is their extensive scaling in terms of both parameters and the data volume used for training, as well as advancements in instruction tuning after the large scale pretraining.

Efforts are underway to adapt these LLMs to low-resource languages, with the Meltemi model [77] serving as a prominent example for the Greek language. This model builds on the foundation of the Mistral-7B model [25], originally trained primarily on English texts. To better accommodate Greek, the developers of Meltemi expanded the original tokenizer's vocabulary by adding a substantial number of Greek tokens. Mistral-7B's pretraining was extended to include Greek, utilizing a corpus of approximately 40 billion tokens—28.5 billion of which are in Greek. This corpus is supplemented by an additional 10.5 billion tokens of English texts and a 600 million token bilingual Greek-English dataset, which aids in maintaining the model's bilingual capabilities and mitigating catastrophic forgetting.

However, most of the techniques mentioned above are focused on the English language and require a vast amount of English data. In the next section, we will discuss techniques that can be used to create open-domain chatbots in languages where there's hardly any conversation data available.

4.3 Techniques for dialogue generation in low resource languages

Open-domain dialogue datasets in languages other than English and Chinese are scarce. Authors in [29] (2021) addressed this gap by crafting a Korean dataset through the translation of the English Wizard of Wikipedia dataset [15]. As far as our knowledge extends, one notable multilingual dataset is XPersona [40]. This dataset is an expansion of the English PersonaChat dataset [89], encompassing languages such as Chinese, French, Indonesian, Italian, Korean, and Japanese. The creation process involves an initial phase of automatic translation for the training, development, and test data. Subsequently, the latter two partitions undergo manual correction, while the training set undergoes a semi-manual cleaning process. Researchers employ this dataset for the evaluation of methods that rely on multilingual models and automatic translation. Additionally, the MDIA [88] dataset offers another valuable resource as the first large-scale multilingual benchmark for dialogue generation, encompassing low- to high-resource languages. This extensive collection includes real-life conversations in 46 languages, spanning 19 language families. However, it is worth noting that the MDIA dataset primarily consists of single-turn dialogues and may not exhibit the highest quality in terms of dialogue data.

In addition to the challenges posed by the scarcity of open-domain dialogue datasets in languages other than English and Chinese, the utilization of Pretrained Language Models (PLMs) in low-resource languages introduces another layer of complexity. PLMs have demonstrated remarkable capabilities in English and, to some extent, Chinese, owing to the vast amounts of training data available. However, for low-resource languages with limited digital content, the performance of these models can be suboptimal. Adapting PLMs to such languages requires innovative strategies, such as cross-lingual transfer learning and data augmentation techniques. Researchers actively explore these approaches to tap into PLMs potential for building dialogue systems in historically underrepresented languages, or languages with a few amount of available data. The development of multilingual PLMs has partly bridged the gap between English and other systems, enabling the utilization of techniques that don't demand extensive data for training dialogue systems in low-resource languages.

4.3.1 Translation and native training

To address the problem of data unavailability, some prior work has focused on training a model on translations of known English datasets. Strategies that leverage neural machine translation models to create training datasets, by converting existing high-resource language datasets into the target low-resource language, can enrich the available data pool for training and introduce linguistic diversity into the model.

The term native training means that we train the model only on the data of the targeted low-resource language (either translated original datasets in that language). This can be done using specific-language pretrained language models, or multilingual models when monolingual ones are not available. In [53], the authors proposed a transformer-based encoder-decoder (BERT2BERT) initialized with AraBERT[2] parameters. They showed this approach facilitates knowledge transfer, significantly improving performance in response generation tasks. To enhance their model with empathy, they trained it using the ArabicEmpatheticDialogues dataset [2], a translated version of the EmpatheticDialogues dataset [62]. Their model achieved superior performance compared to previous state-of-the-art models, as evidenced by the lower perplexity value and higher BLEU scores.

Given the constraints in data and computational resources, which limit the availability of Pre-trained Language Models (PLMs) for many languages, an alternative approach has emerged. Researchers have begun fine-tuning multilingual PLMs with these translated datasets, adapting them to the linguistic specifics of the target languages. However, another investigation highlighted the limitations of translating high-resource language datasets into low-resource languages [69]. Through experiments involving English and Chinese dialog data, the authors examined different cross-lingual transfer methods. They found that directly training multilingual models using English dialog corpora without translation can be more effective than using translated versions. This is attributed to the cultural specificity of dialog, where direct translations may not capture the nuances of the target language, leading to unnatural dialog generations. The study suggests focusing on utilizing untranslated high-resource language data for cross-lingual transfer, providing insights into the limitations of MT in dialog generation tasks.

4.3.2 Cross-lingual transfer learning

Few-shot cross-lingual transfer learning, supported by multilingual pre-trained language models offers a promising solution for various natural language processing (NLP) tasks, particularly for languages with limited resources. This approach typically involves two key phases, and :

- Source-training: The mPLM is initially fine-tuned using the comprehensive training dataset available in a source language, such as English.
- Target-adapting: Subsequently, the model that has been trained on the source language is fine-tuned further using a small number of example data points (i.e., few-shot examples) from the target language.

While using Multilingual models has shown promise in enabling cross-lingual transfer for other generation tasks, zero-shot cross-lingual transfer with mPLM suffers much from catastrophic forgetting, where mPLM that has been fine-tuned on the source language is unable to generate fluent sentences in the target language when being evaluated on it. Vinit et al. [31] showed that while these multilingual transformers show impressive transfer capabilities, their performance significantly drops for languages that are linguistically distant or have smaller training corpora. In their study, however, they stated that inexpensive few-shot transfer (i.e., additional fine-tuning on a few target-language instances) can be surprisingly effective across the board.

The study conducted by Otegi et al. [56] came to examine the previous techniques, within the framework of conversational question-answering (CQA) tasks, specifically targeting the Basque language. They examined the performance of CQA systems using native training data only, curating a small dataset through crowdsourcing, zero-shot transfer learning (using English training data only), and low-resource transfer (a combination of native and English training data). For the native training, they fine-tuned both monolingual and multilingual models. The results demonstrated that by fine-tuning multilingual PLMs with English data and doing few-shot learning using a small amount of native data, it is possible to achieve performance comparable to English-centric systems. Interestingly, the monolingual model, fine-tuned with native data, performed nearly as well as the multilingual model pre-trained on a larger English dataset before undergoing few-shot learning, and much better than the multilingual that was trained on the native Basque data.

4.3.3 Mutlitask learning

Multitask learning (MTL) is an effective inductive transfer approach that improves generalization by jointly learning one or more auxiliary tasks together with the target task. In this work, we focus on pairwise MTL, where there is only one auxiliary task trained together with the target task, as it works better when the target dataset is smaller than the auxiliary dataset [43]. In our case, we have auxiliary language(s) and target language as auxiliary task(s) and target task, respectively.

The authors in [80], investigated the differences of Sequential Transfer Learning with Intermediate Tasks in comparison to Pairwise Multi-Task Learning. In our case, the sequential transfer learning with intermediate tasks is identical to the cross-lingual transfer learning methods, where we first finetune a model on a source language and then we train the model on the target language. In their research, they found out Pairwise MTL tends to outperform STILTs (Supplementary Training on Intermediate Labeled-data Tasks) when the target task has fewer instances than the supporting task. Conversely, STILTs is preferable when the target task has more instances than the supporting task.

Multitask learning (MTL) has shown significant promise in enhancing performance across various natural language processing tasks by leveraging shared representations, however little to no work has to do with tasks around dialogue generation on a low resource language.

Recent studies have explored different facets of MTL, applying it to both high-resource and low-resource languages. For instance, Zhu et al. [97] developed a framework combining natural language generation (NLG) with an unconditioned language model. This approach not only addressed the semantic correctness of the responses but also their naturalness, a crucial aspect often missing in low-resource settings. They reported that this dual-task model significantly outperformed traditional single-task models across various datasets, demonstrating the effectiveness of MTL in generating more diverse and contextually appropriate responses.

Similarly, Ide and Kawahara [24] focused on integrating emotion detection with response generation, enhancing the emotional intelligence of dialogue systems. By training a model to recognize and generate emotional responses simultaneously, their system could engage more naturally with users, an approach that could be particularly effective for culturally nuanced languages like Greek.

In the context of Greek, where data scarcity poses a significant challenge, MTL offers a pathway to enrich dialogue systems by borrowing strength from related tasks. Magooda et al. [49] explored MTL for abstractive summarization in low-resource languages and found that tasks like paraphrase detection and concept detection could enhance summarization quality. Applying similar principles to dialogue response generation, where generating paraphrases and detecting relevant concepts are equally vital, could similarly improve performance.

4.3.4 Prompt learning

Prompt learning represents a strategic adaptation of pre-trained language models (PLMs) to specialized tasks using tailored input modifications, which guide the models' generative capabilities without the need for extensive retraining. This method is especially relevant in the context of dialogue systems where generating contextually appropriate responses is critical but often hindered by the scarcity of training data in low-resource languages.

Madotto et al. [48] introduced prompt-based few-shot learning for dialogue systems, demonstrating that significant performance can be achieved by embedding task-specific prompts into the model's input. This approach leverages the intrinsic capabilities of large language models trained on diverse datasets to perform tasks with only a few instructive examples. The efficacy of this method provides a framework for applying large models to dialogue generation tasks where training data may be limited.

Kasahara et al. [27] further explored the utility of prompt tuning in dialogue systems by optimizing only the prompt's embedding vectors while keeping the rest of the model parameters frozen. Their findings suggest that such an approach not only maintains the generative quality of responses but also reduces the computational overhead associated with traditional fine-tuning methods. This methodology is particularly advantageous for low-resource scenarios, where computational resources and domain-specific data are often limited.

Brown et al. [8] investigated the application of zero-shot and few-shot learning paradigms using manually crafted prompts that encapsulate task-specific directions and examples. Their research highlights the potential of using prompts to adapt pre-trained models to new tasks efficiently, without the need for large annotated datasets. Such strategies are pivotal for languages and dialects that lack extensive computational resources and linguistic data.

These studies collectively underline the transformative potential of prompt learning in making advanced NLP technologies accessible for low-resource languages. By employing minimal prompt engineering and leveraging pre-existing large models, researchers and practitioners can develop robust dialogue systems capable of handling a variety of interactions with reduced resource investment. This body of work not only contributes to the theoretical understanding of prompt-based learning but also provides a practical blueprint for its application across diverse linguistic landscapes.

4.4 Evaluation Metrics

In this section, we look into the most commonly used metrics for evaluating the generated responses of an open-domain conversational agent. These metrics are divided into automatic and human-based metrics.

4.4.1 Automatic Metrics

Although there is no well-established method for automatic evaluation of the response quality, there are some automatic metrics for reference.

- **Word Perplexity:** This metric, designed to evaluate probabilistic language models [6], is widely used in end-to-end dialogue systems assessment. It calculates the likelihood of the model predicting the next word in a conversation accurately. A lower perplexity indicates a better model. Modifications to this metric exclude stop-words and punctuation to emphasize the semantic content [52]. Despite its popularity, its effectiveness is limited in dialogue systems due to the diverse range of potential responses and the numerous ways a sentence can be constructed while retaining the same meaning.

- **BLEU:** Originating from machine translation evaluation, BLEU [57] scores a response based on how well it matches n-gram sequences found in a reference response. The formula is given by:

$$BLEU = BP \cdot \exp\,(\sum_{n=1}^{N} w_n \log p_n) \tag{4.2}$$

  where BP is the brevity penalty on the length of the utterance, p_n represents the probability that the n-grams in a generated response occur in the real response, N is the max number of grams, and w_n is the weight for each n-gram (normally set as $1/N$). BLEU's output is always a number between 0 and 1. A score close to 1 indicates a high similarity to the reference, suggesting better model performance. However, its correlation with human judgment on dialogue quality is weak [41].

- **SacreBLEU:** SacreBLEU [58] provides a standardized methodology for computing the BLEU score, including consistent preprocessing and tokenization. While primarily designed for machine translation, sacreBLEU can be applied to evaluating conversational agents by providing a standardized measure for comparing the generated responses to reference responses. However, like BLEU, sacreBLEU may not fully capture the conversational context or the appropriateness of responses in a dialogue setting.

- **BERTScore:** BERTScore leverages the contextual embeddings from BERT and computes the cosine similarity between the embeddings of words in the generated and reference texts [90]. It is particularly useful for evaluating conversational agents where semantic accuracy and the ability to produce contextually relevant responses are crucial. BERTScore can capture the subtleties of meaning that traditional metrics might miss, offering a deeper insight into model performance.

- **Response Diversity:** Distinct-1 and Distinct-2 measure the number of distinct unigrams and bigrams of the generated responses [34], trying to measure the diversity of the generated responses.

4.4.2 Human-based Metrics

Currently, human evaluation is still the most convincing method for judging the response quality and is widely applied in chatbot evaluation. The most common human-based metrics are:

- Pair-wise comparison to let humans choose which of the two responses is more suitable, more appropriate, and more helpful, etc. [68].
- Evaluating relevance: Humans grade the generated responses according to whether they seem relevant to the conversation and on-topic. [62].
- Evaluating fluency/coherency: Humans grade the responses according to whether they seem understandable, logically, and syntactically correct. [59].

In conclusion, evaluating open-domain conversational agents requires an all-around approach, combining a variety of metrics. Using a combination of metrics such as Word Perplexity, BLEU, Distinct-N, BERTScore, along with sacreBLEU for standardized comparisons, offers valuable insights into various aspects of conversational quality, including fluency, relevance, and semantic accuracy. However, the limitations of these metrics underscore the importance of human-based evaluations for capturing the nuances of natural conversation, including context, coherence, and user satisfaction.

4.5 Summary

In this chapter, we provided a theoretical background knowledge of open-domain dialogue generation for languages with limited resources, starting with a foundational discussion on the evolution of natural language, which has significantly influenced human interaction and societal development. We explored the roles of task-oriented and non-task-oriented dialogue systems, emphasizing their functionality in everyday applications and their broader implications in AI-driven communications. In Section 4.2 we broke down dialogue generation mechanisms, transitioning from traditional rule-based systems, like ELIZA [79], to modern data-driven approaches that use vast corpora to train more nuanced conversational systems, like ChatGPT [55]. We considered various methodologies like retrieval-based and generative systems, each with unique strengths in coherence and contextual relevance, and look into the previous advances in the field of open domain dialogue.

In Section 4.3, the narrative shifted towards the challenges faced by low-resource languages in developing dialogue systems. We examined strategies such as translating high-resource datasets and native training, which adapt existing content to enrich linguistic databases for underrepresented languages. Cross-lingual transfer learning was highlighted as a key technique for using robust multilingual models to extend the benefits of advanced NLP technologies to these languages. Multitask learning and prompt-based learning were discussed as methods that enhance model performance by integrating auxiliary tasks or fine-tuning models with minimal examples, respectively.

Finally, the chapter concluded with an analysis of the usual metrics, automatic and human-based, that are used to evaluate open domain dialogue systems. These assessments are crucial for understanding the effectiveness of various models in producing relevant, coherent, and contextually appropriate responses, thereby pushing the boundaries of what automated systems can achieve in real-world interactions.

Chapter 5

Dialogue Generation - Greek Case

5.1 Introduction

Open-domain conversational models aim to seamlessly blend knowledge and intelligence while satisfying users' need for communication and social belonging. A long-standing goal of Artificial Intelligence (AI) has been to build intelligent open-domain conversational models that can understand the semantics of input utterances and provide coherent and relevant responses. Early attempts in building open-domain models relied on developing Natural Language Processing (NLP) modules for utterance understanding and generation, and a dialog manager to switch between modules [95]. However, such approaches remained limited in their capabilities and failed to generalize beyond a specific set of domains. With the advances in AI techniques and computation power, researchers recently showed that open-domain conversational models developed using end-to-end neural network-based approaches, such as Sequence-to-Sequence (Seq2Seq) models and autoregressive transformers, generalize well to unforeseen domains without needing complicated setups or predefined modules and hand-crafted rules [63]. These approaches, however, require training on large corpora of open-domain conversational data [1]. The availability of massively pre-trained language generation models also helps address the issue of data scarcity whereby less task-specific labeled data would be needed to reach reasonable performance.

However, when it comes to languages that aren't widely supported, like Greek, the challenge is finding enough conversational data, and the pre-trained models available aren't as advanced as those for English. This gap led us to our research goal: finding the best way to create a chat model for open-domain conversations in Greek, where resources are scarce.

The rest of this chapter is organized as follows: Section 2 overviews recent work on open-domain response generation models, in addition to works targeting low-resource languages. Section 3 presents the dataset we used for our experiments. The proposed architectures are explained in Section 4. In section 5, we discuss the different approaches used, the training implementations, the conducted experiments, and the results achieved.

5.2 Related Work

With the progress in neural conversational AI, there has been a resurgent interest in developing open-domain conversational models. These models are typically trained using large amounts of open-domain conversational datasets and also benefit from massively pre-trained language generation models that can be fine-tuned on the target open-domain response generation task. The wide availability of such resources has contributed to the development of high-performing open-domain conversational models [91], [63]. Despite the existence of valuable resources to build open-domain conversational models [89], [37], [62], most of them are in English, making it challenging to produce similar models for other languages.

The low-resource challenge has been previously studied in the literature for task-oriented conversational models [83], machine translation [51], question-answering [56], and other NLP applications [21], [31], as we extendedly discussed in Section 4.3. However, very little work targeted the issue of low resources in open-domain conversational models.

Yang et al. [87] studied the problem of low-resource response generation with 360K utterance response pairs in Chinese. The authors proposed estimating templates from large-scale unlabeled samples to aid an encoder-decoder model in response generation. Naous et al. [53] achieved high performance in open-domain response generation in Arabic by fine-tuning a transformer model on 36K utterance-response samples that were automatically translated from English [54]. In our work though, we tackle the problem of open-domain response generation in Greek using a chit-chat dataset of 11k dialogues. To the best of our knowledge, this is the first attempt to tackle open-domain response generation in Greek.

5.3 Data

In this section, we present and analyze the DailyDialog dataset that we used for our experiments.

5.3.1 DailyDialog dataset

The DailyDialog [37] dataset is a publicly available collection of multi-turn dialogue conversations that span across a diverse range of topics. The dataset consists of over 13,000 conversations between two or more speakers, with each conversation consisting of up to 15 turns. Each conversation has on average 7.9 utterances and the average utterance length is about 15 words long. The basic statistics are presented in Table 5.1

Total Dialogues	13,118
Average Speaker Turns Per Dialogue	7.9
Average Tokens Per Dialogue	114.7
Average Tokens Per Utterance	14.6

Table 5.1: Caption

The conversations were collected from various websites that serve for English learners to practice English dialog in daily life. DailyDialog datasets are written to reflect our daily conversations, so the main purpose of the dialogues is to exchange information and enhance social bonding. The dialogues in the developed dataset cover a wide range of daily scenarios: chit-chats about holidays and tourism, service-dialog in shops and restaurants, and so on. The 3 most common topic categories of a conversation are Relationship (33.33%), Ordinary Life (28.26%), and Work (14.49%). Finally, the dataset was split into training, validation, and test sets. A conversation from the training set is provided below.

Turn 1: **A:** I'm worried about something.

Turn 2: **B:** What's that?

Turn 3: **A:** Well, I have to drive to school for a meeting this morning, and I'm going to end up getting stuck in rush-hour traffic.

Turn 4: **B:** That's annoying, but nothing to worry about. Just breathe deeply when you feel yourself getting upset.

Turn 5: **A:** Ok, I'll try that. *[A contemplates B's advice]*

Turn 6: **B:** Is there anything else bothering you?

Turn 7: **A:** Just one more thing. A school called me this morning to see if I could teach a few classes this weekend, and I don't know what to do.

Turn 8: **B:** Do you have any other plans this weekend?

Turn 9: **A:** I'm supposed to work on a paper that's due on Monday.

Turn 10: **B:** Try not to take on more than you can handle.

Turn 11: **A:** You're right. I probably should just work on my paper. Thanks!

5.3.2 Greek DailyDialog dataset

To create the Greek version of the DailyDialog dataset(el) we translated the original English DailyDialog(en) dataset using neural machine translation. The neural machine translation model from the OPUS collection [73] serves as the primary tool for this translation task, offering state-of-the-art performance in translating between English and Greek. A conversation from the projected dataset is provided below.

Turn 1: **A:** ανησυχώ για κάτι.

Turn 2: **B:** τι είναι αυτό·

Turn 3: **A:** λοιπόν, πρέπει να πάω στο σχολείο για μια συνάντηση σήμερα το πρωί, και θα καταλήξω να κολλήσω στην κίνηση ρας.

Turn 4: **B:** αυτό είναι ενοχλητικό, αλλά τίποτα να ανησυχείς. απλά αναπνέεις βαθιά όταν νιώθεις τον εαυτό σου να αναστατώνεται.

Turn 5: **A:** εντάξει, θα το δοκιμάσω.

Turn 6: **B:** σ ένοχλεί κάτι άλλο·

Turn 7: **A:** ένα ακόμα πράγμα. ένα σχολείο μου τηλεφώνησε σήμερα το πρωί για να δω αν μπορώ να διδάξω μερικά μαθήματα αυτό το σαββατοκύριακο, και δεν ξέρω τι να κάνω.

Turn 8: **B:** έχεις άλλα σχέδια αυτό το σαββατοκύριακο·

Turn 9: **A:** υποτίθεται ότι πρέπει να δουλέψω σε ένα χαρτί που οφείλεται τη δευτέρα.

Turn 10: **B:** προσπάθησε να μην αντέξεις περισσότερα από όσα μπορείς να χειριστείς.

Turn 11: **A:** έχεις δίκιο. μάλλον πρέπει να δουλέψω πάνω στο χαρτί μου. ευχαριστώ!

5.4 Proposed architectures

In this section, we describe the generative models that we used for our task and compare the results of all models in Section 5.5. We tried 4 different generative models, a monolingual autoregressive decoder model, a monolingual encoder-decoder model, a multilingual autoregressive decoder model, and a multilingual encoder-decoder model. The models we can use for our task are limited as there are few monolingual and multilingual models trained on a Greek corpus. So, there is a comparable difference in the number of parameters between our 4 proposed models, as shown in the details below.

Model 1 - GPT2-Greek: This model used the GPT-2 architecture described in Section 3.4.2. GPT2-Greek is a monolingual model and has 117M parameters (12-layer, 768-hidden, 12-heads). It was developed by finetuning the English version with gradual layer unfreezing. We initialize the model using the 'lighteternal/gpt2-finetuned-greek' [38] checkpoint from the HuggingFace library.

Model 2 - GREEK-BERT2GREEK-BERT: This monolingual model uses the seq2seq architecture, with encoder and decoder both composed from transformer models. More specifically, the encoder and the decoder are initialized with the weights of the GREEK-BERT, whereas the language model head at the top of the decoder and the cross-attention layers are randomly initialized. The GREEK-BERT2GREEK-BERT model has 224M parameters as the GREEK-BERT encoder and decoder models are similar to the English BERT model and have 110M parameters (12-layer, 768-hidden, 12-heads).

Model 3 - mT5: This multilingual model follows the original T5 recipe described in section 3.4.3. We use the small model which has 300M parameters. The bigger vocabulary employed in mT5 results in a higher parameter count when compared to the corresponding T5 small model.

Model 4 - XGLM: XGLM is a multilingual decoder-only generative model with transformer architecture similar to GPT-3 described in section 3.4.4. We use the smallest XGLM model with 564M parameters which has a similar architecture to GPT-3 Medium (24-layer, 1024-hidden, 16-heads).

5.5 Experiments and Results

In this section, we first outline the different training approaches utilized, delve into the training specifics of the models highlighted in Section 5.4, detail the experiments conducted, and evaluate the outcomes of the proposed models.

5.5.1 Training approaches

For our experiments, we implemented four primary training approaches, based on the ideas from previous work in this domain as described in Section 4.3: native training, cross-lingual transfer learning, multitask learning, and prompt based learning.

Native training

Following the methodology proposed by the authors in [54], our initial approach involved training on a Greek dialogue generation dataset. Given the absence of such datasets, we utilized the Greek segment of the Daily Dialog dataset (see Section 5.3.2), which we had translated. We used a full model fine-tuning training approach, which was conducted solely with Greek data, allowing all four models, both monolingual and multilingual, to be trained using the same language modeling loss. The training processes for the decoder-only models differed slightly from those for the seq2seq models, as will be further explained in Section 5.5.2.

Cross-lingual transfer learning

As outlined in Section 4.3.2, cross-lingual transfer learning is an effective strategy for transferring knowledge from a source language, in this case, English, to a target language, Greek. To facilitate this, both the original and the translated versions of the DailyDialog dataset are necessary. Consequently, as this stage involves training on both English and Greek data, only models 3 and 4, which are multilingual were trained under these conditions. The typical approach involved first fine-tuning the model on the English dataset to allow the model to learn dialogue generation with abundant high-quality data. Subsequently, the model was fine-tuned again using various subsets of the Greek dataset, with $k = 32, 64, 128, 512, 1024$ examples. This method aims to transfer the learned representations from the initial training stage to the target language, where less data is available. Both fine-tuning stages involved training the complete set of model parameters, rather than freezing any layers or using selective parameter updates.

Multitask learning

The concept and settings for multitask learning are akin to those of few-shot learning. Here, the objective is to learn the same task across different languages simultaneously. A substantial portion of English data is used to adjust the models' parameters, while simultaneously incorporating a smaller proportion of Greek dialogues. This approach seeks to synchronize the token embeddings from both languages, primarily tuning the models using the extensive and well-prepared English data.

Instead of conducting two separate fine-tuning processes as in cross-lingual transfer learning, we merged the Greek dataset samples with the English dataset to form a new, predominantly English, joint dataset. The number of Greek samples used to create the joint dataset varies, similarly to the Cross-Lingual transfer learning method. For training, we used a full-parameter fine-tuning approach. The models are thus trained to handle the task in English while also attempting to master the more challenging, data-sparse Greek task, leveraging the knowledge acquired from the English data.

Prompt based training

Our experiments with cross-lingual transfer learning and multitask learning revealed a tendency for the models to experience catastrophic forgetting, a phenomenon also noted by other researchers [42]. To address this issue, we adopted a strategy involving predefined hard prompts that are consistent across both languages similar to [48]. These prompts help direct the flow of information, aiding the model in grasping essential dialogue elements that are common across languages, thereby enhancing knowledge transfer from English to Greek.

To implement this strategy, we prepended each input with the phrase "Dialog history". Additionally, we added "User:" before the user's input and "System:" before our model's input. We re-implemented two training setups from the cross-lingual transfer learning and multitask learning experiments, setting the number of examples to 128, using these prompts as shown below:

$$Dialoghistory :< context >\ \ User :< user\_input >\ \ System :< model's\_output >$$

5.5.2 Training details

In this section, we discuss the details of all the different training approaches we discussed in the previous Section 5.5.1, and the way we tailored each approach to the models we discussed in Section 5.4.

GPT2-Greek training

We trained the model on the translated DailyDialog dataset. Each training instance consisted of an entire dialogue from those in the training set with a special token inserted between each utterance of the dialogue. During training, we wanted to optimize the response language modeling objective, trying to predict the next word and computing the language model loss using cross-entropy. The loss is computed on the whole dialogue and not only on the gold reply of the last sentence. In that way, the model learns the patterns between all the utterances of the dialogue and does not only learn to generate the final response depending on the dialogue history.S

During fine-tuning, we used the AdamW optimizer with a learning rate of 7e-5, weight decay equal to 0.01, and a linear scheduler. The model was trained with early stopping, keeping the checkpoint with the best language model loss in the validation set. The training set contains 11118 dialogues and we use a batch size of 8. With gradient accumulation steps set to 2, the effective batch size reached 16.

GREEK-BERT2GREEK-BERT training

The GREEK-BERT2GREEK-BERT model was also trained using the translated DailyDialog dataset. However, a different strategy was used to create the training instances. We first tried to have the same number of training examples as with the previous model, by giving as input to the encoder the first i-1 sentences of the dialogue, where i is the total number of sentences in the dialogue, and calculating the cross-entropy loss between the decoder output and the last sentence of the dialogue. This method leads to poor generalization, also due to the small number of dialogues in the dataset, since the model learns to generate only the last sentence of each dialogue given all the previous ones and does not learn to generate every turn of the dialogue. For this reason, we divided each dialogue into smaller dialogues that formed the different training instances, as shown below.
For example, we have the dialogue:

1: "I met Carson's mother last week for the first time.",
2: "How was she?",
3: "She turned out to be really nice. I like her.",
4: "That's good to hear."

From the above dialogue, after the processing, emerge 3 new dialogues as shown below.

" I met Carson's mother last week for the first time.",
"How was she?"

" I met Carson's mother last week for the first time.",
"How was she?",
"She turned out to be really nice. I like her."

" I met Carson's mother last week for the first time.",
"How was she?",
"She turned out to be really nice. I like her.",
"That's good to hear."

For our encoder-decoder models, the training instances are formed as:

$< context >$ $[special\_token]$ $< user\_input >$ $[special\_token]$, for the encoder input.

Therefore, using the above example, $< context >$ consists of sentences 1 and 2, $< user\_input >$ consists of sentence 3, while sentence 4 is the label based on which the cross-entropy loss is calculated. During fine-tuning, we wanted to optimize the response language modeling objective, trying to predict the next sentence and computing the language model loss between the generated response and the label using cross-entropy. We used the AdamW optimizer with a learning rate of 1e-4, weight decay equal to 0.01, and a linear scheduler with 500 warm-up steps. The model was trained with early stopping for 10 epochs, keeping the checkpoint with the best language model loss in the validation set. After the preprocessing of the dataset, the augmented training set contains now 76052 dialogues and we use a batch size of 16.

mT5 Training

The mT5 model used the same structure for training instances as the other seq2seq model we discussed previously, GREEK-BERT2GREEK-BERT model and underwent all four training approaches. The models resulting from these were designated as mt5-native, mt5-cross-lingual, mt5-multitask, and mt5-prompt respectively. The focus was on optimizing the language modeling objective by predicting the next tokens and computing the language model loss using cross-entropy. For fine-tuning, the mT5-native and mT5-transfer models used the AdamW optimizer with a learning rate of 7e-5 and a linear scheduler. The mT5-multitask model was initially trained on the English dataset for three epochs with a constant learning rate of 7e-5, then fine-tuned on the Greek dataset with a learning rate of 3e-5. Weight decay for all models was set at 1e-2. All training was conducted with early stopping, utilizing a dataset of 76,052 dialogues and a batch size of 16.

XGLM Training

The XGLM model followed the same setup as the GPT2-Greek model for training instances. Being a multilingual model, it was subjected to four different training techniques: XGLM-native (XGLM-NV fine-tuned solely on Greek dialogues), XGLM-cross-lingual (XGLM-CL, trained sequentially on English and then Greek datasets using k examples), XGLM-multitask (XGLM-MTL, fine-tuned simultaneously on both datasets using k examples), and XGLM-prompt where it was again trained on cross-lingual transfer(XGLM-P-CL) and multitask (XGLM-P-MTL) learning, but using the extra prompt tokens on the input sequences as discussed in 5.5.1. Each dataset was maintained in its original state with 11,118 dialogues. Due to the large size of the model and resource constraints, a batch size of 2 with gradient accumulation steps equal to 8 was used, achieving an effective batch size of 16. The AdamW optimizer was employed across all models with a learning rate of 7e-5 for XGLM-native and XGLM-transfer, and 3e-5 for XGLM-multitask during fine-tuning.

5.5.3 Evaluation

To assess the performance of the language models, we used several metrics like, perplexity, SacreBLEU, Distinct-N, and BertScore. Each of these tools helps us understand different aspects of the model's performance, from fluency and diversity to accuracy regarding a golden truth response and contextual alignment.

Perplexity: Perplexity evaluates the model's uncertainty in predicting the next token. A lower score indicates a more confident model, reflecting better understanding of the language.

SacreBLEU: SacreBLEU is a refinement of the BLEU score, which is widely used to assess the quality of text that a model generates or translates. It includes several metrics:

- **BLEU-1:** Measures the match of single tokens (unigrams) between the model's output and the reference. It assesses the accuracy at the most basic level of text generation.
- **BLEU-2:** Evaluates the co-occurrence of two consecutive tokens (bigrams) in the model's output compared to the reference. This measures how well the model captures two-word phrases, reflecting more on syntactic structures than BLEU-1.
- **Overall BLEU Score:** This score aggregates the model's performance across different n-gram lengths (up to 4), weighted by a geometric mean. It provides a comprehensive picture of how well the model's output aligns with the reference text across different levels of granularity.

We adapted SacreBLEU to assess how the model's responses align with a golden truth, focusing on both the precision of individual words and the fluency of phrases.

Distinct-N: Distinct-N metrics, including Distinct-1 and Distinct-2, measure the diversity of the generated text by counting the unique n-grams normalized by the total number of words. Higher values indicate a richer and more varied vocabulary.

BertScore: BertScore checks the semantic similarity between the model's output and the reference text using BERT-based embeddings. High BertScore values suggest strong semantic alignment, indicating effective context understanding and relevance of the model's responses.

We consolidate these metric scores into tables and charts, allowing us to visually compare the capabilities of the different models and techniques we developed.

5.5.4 Results

This section presents the comprehensive evaluation of various models trained using different methodologies: native training, cross-lingual transfer learning, multitask learning, and prompt-learning training. In Table 5.2 is the main summary of the results. The Tables 5.4 and 5.5 show a study regarding the number of examples k on the cross-lingual transfer and multitask learning settings for the 2 multilingual models.

Overall, the models trained on the whole translated dataset (native training) achieve better performance across all the metrics. The results show that among the models trained natively, mT5-NV achieved the lowest perplexity of 6.52, indicating superior predictive performance compared to other models in this category. In terms of SacreBLEU scores, which measure the similarity of the generated answer in comparison to a human golden answer, in terms of using the same n-grams, XGLM-NV outperformed others with scores of 27.58 for B-1, 13.01 for B-2, and an overall score of 6.29. This suggests that XGLM-NV generates answers closer to the real ones of the dialogues, without these assuring coherence. However, the average score falls a bit behind GPT2-Greek-NV indicating that the latter had better scores on more complicated B-3, and B-4 scores.

For diversity, GPT2-Greek-NV demonstrated the highest Distinct-1 (23.13) and Distinct-2 (51.28) scores, indicating that it produced more varied text outputs. This is crucial for applications requiring rich and diverse language generation. In the context of Bertscore, which evaluates the similarity between generated and reference texts using BERT embeddings, GPT2-Greek-NV achieved the highest scores with precision at 71.53, recall at 71.47, and an F-1 score of 71.37. This model's performance suggests it is highly effective in generating text that closely matches the reference texts in meaning and quality.

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	score	Distinct-1	Distinct-2	Precision	Recall	F-1
Native training									
GREEK-BERT2GREEK-BERT-NV	14.16	23.82	8.43	5.66	16.72	42.23	70.58	69.77	70.01
GPT2-Greek-NV	12.47	25.93	11.07	6.93	23.13	51.28	71.53	71.47	71.37
mT5-NV	6.52	25.37	9.64	5.74	19.51	43.36	70.11	69.02	69.56
XGLM-NV	9.95	27.58	13.01	6.29	19.34	43.02	70.68	68.32	69.33
cross-lingual transfer training									
mT5-CL (k=128)	19.39	13.54	5.99	3.53	18.71	37.62	64.12	63.24	63.52
XGLM-CL (k=128)	15.74	23.61	10.03	4.95	18.75	41.91	69.60	68.32	68.99
Multitask training									
mT5-MTL (k=128)	12.27	18.9	6.83	3.93	21.12	46.35	68.35	67.96	68.04
XGLM-MTL (k=128)	16.53	23.25	9.89	4.75	18.24	40.39	69.53	68.25	68.75
Prompt-learning training									
mT5-P-CL (k = 128)	16.12	18.64	8.26	4.41	18.52	38.22	66.31	64.36	65.12
mT5-P-MTL (k = 128)	13.45	13.83	4,99	3.12	18.52	43.27	64.48	55.59	65.12
XGLM-P-CL (k=128)	10.47	25.01	12.04	4.52	18.31	40.16	69.75	68.14	68.84
XGLM-P-MTL (k=128)	11.31	25.07	12.10	5.00	16.91	38.31	69.89	68.50	69.12

NV: Native training, CL: Cross-Lingual transfer learning, MTL: Multitask learning, P: prompt
k: number of examples

Table 5.2: Results across all the different training approaches and models on Greek test set

In general, GPT-Greek-NV outperforms the other models and training techniques. This superior performance is likely due to its prior training on Greek data, which provided a strong foundation. Additionally, the use of a Greek tokenizer significantly contributed to its ability to generate more diverse responses. On the other hand, XGLM's performance is underwhelming considering its larger size compared to the other models.

The results from the Table 5.2 reveal that XGLM models trained with prompts—both in cross-lingual transfer (XGLM-P-CL) and multitask learning scenarios (XGLM-P-MTL)—demonstrate superior performance in key metrics over their counterparts without prompt integration. Specifically, XGLM-P-CL achieved SacreBLEU scores of 25.01 for B-1 and 12.04 for B-2, which are improvements over the non-prompt XGLM-CL model. This suggests that the inclusion of prompts leads to more accurate and contextually relevant dialogue outputs. Furthermore, the BertScores for XGLM-P-CL and XGLM-P-MTL (F-1 scores of 68.84 and 69.12, respectively) are higher compared to their non-prompt counterparts, indicating a closer semantic similarity to human-like dialogue responses. However, we don't see the same improvements for mT5, as the prompt learning does not seem to benefit the model.

Table 5.3 presents the performance evaluation of various training approaches on the English test set, providing insights into how different adaptation methods affect the models' retention of original language capabilities. While the primary focus of this study centers on Greek language generation performance, evaluating on English serves as a crucial diagnostic tool to assess the degree of catastrophic forgetting and knowledge preservation during cross-lingual adaptation. The results demonstrate clear performance hierarchies across native training, cross-lingual transfer learning, multitask learning, and prompt-based approaches.

The superior performance of prompt-based cross-lingual transfer (XGLM-P-CL, mT5-P-CL) and multitask (XGLM-P-MTL, mT5-P-MTL) approaches compared to their non-prompt counterparts reveals the severe impact of catastrophic forgetting in traditional fine-tuning methods. When models are adapted to new languages without prompts, they experience degradation not only in target language performance but also in fundamental linguistic capabilities that form the backbone of language generation. This degradation can extend beyond language-specific knowledge to core competencies such as fluency, coherence, and dialogue structure that were originally acquired during English intermediate fine-tuning. This forgetting of language generation capabilities is the primary factor explaining the worse per-

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	score	Distinct-1	Distinct-2	Precision	Recall	F-1
Native training									
XGLM (English only)	8.37	24.08	8.51	6.12	10.29	38.25	69.67	73.18	71.28
mT5 (English only)	8.34	28.15	12.19	6.48	11.51	36.26	72.33	71.40	71.74
cross-lingual transfer training									
XGLM-CL (k=128)	9.84	21.23	5.89	4.12	8.21	28.73	66.42	68.91	67.58
mT5-CL (k=128)	9.71	24.18	9.87	5.23	9.83	31.25	69.12	67.28	68.15
Multitask training									
XGLM-MTL (k=128)	9.12	24.45	6.28	4.51	8.15	30.24	68.21	71.82	69.89
mT5-MTL (k=128)	8.95	26.18	10.92	5.62	10.89	34.15	71.43	70.17	70.76
Prompt-learning training									
XGLM-P (English only)	6.22	25.09	8.07	5.53	9.11	33.23	69.90	73.03	71.34
mT5-P (English only)	8.38	27.32	12.08	6.33	11.78	37.93	71.89	71.26	71.46
XGLM-P-CL (k=128)	6.92	24.81	7.78	5.16	8.77	32.26	69.36	72.31	70.71
mT5-P-CL (k=128)	9.48	25.77	10.92	5.69	12.54	38.22	71.45	70.66	70.94
XGLM-P-MTL (k=128)	6.29	26.80	7.74	5.39	8.69	31.66	69.83	72.88	71.23
mT5-P-MTL (k=128)	8.48	26.35	11.38	5.87	10.16	30.39	72.04	71.16	71.48

P: Prompt-based, CL: Cross-Lingual transfer learning, MTL: Multitask learning
k: number of examples

Table 5.3: Results across all the different training approaches and models on English test set

formance, rather than the model losing understanding of the source language. This same mechanism also explains the degradation in target language performance when not using prompts, even though the amount of Greek training data remains essentially the same across all conditions. Consequently, the model loses essential generative capabilities that affect overall performance quality, as evidenced by the deteriorated BLEU scores, reduced diversity metrics, and lower semantic coherence in non-prompt approaches.

These findings illustrate that prompt-based training not only enhances the linguistic accuracy and relevancy of the generated text but also ensures that the dialogue maintains a high level of diversity and complexity. This is crucial in dialogue systems where the ability to generate coherent, context-aware, and varied responses can significantly affect user satisfaction and engagement. Therefore, integrating prompt-based training in the XGLM model capitalizes on its architectural strengths, enabling more effective learning from fewer examples, which is particularly beneficial in scenarios with limited training data.

Moreover, the implementation of prompts in the training process significantly aids in mitigating the issue of catastrophic forgetting as the model transitions from English to Greek datasets. Catastrophic forgetting occurs when a neural network loses the information previously learned upon learning new information, which is a common challenge when adapting models to new languages or datasets. By integrating prompts, the XGLM model is better equipped to retain relevant features from the English training data while effectively acquiring new linguistic patterns from the Greek data. Prompts serve as anchors or guides that help maintain the model's focus on crucial aspects of the dialogue, ensuring that the transition between languages does not remove previously established capabilities.

Next, we conducted an ablation study for mT5 and XGLM regarding cross-lingual transfer and multitask learning training techniques. Specifically, we ran experiments for each model using k random examples from the Greek dataset, where $k = 32, 64, 128, 512, 1024$, and results are shown in Tables 5.4 and 5.5.

The multitask learning approach for the mT5 model shows varying results depending on the number of examples (k). As k increases, there is a noticeable improvement in performance across most metrics. Perplexity consistently decreases with the number of examples, with mT5-MTL (k=1024) achieving the lowest perplexity of 8.92, indicating better predictive

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	average-score	Distinct-1 (%)	Distinct-2 (%)	Precision (%)	Recall (%)	F-1 (%)
mT5-MTL (k=32)	14.78	13.96	5.98	3.05	20.51	45.52	66.84	65.36	65.91
mT5-CL (k=32)	22.46	13.39	6.24	3.55	14.92	28.79	64.29	62.23	63.15
mT5-MTL (k=64)	13.06	17.32	6.64	3.77	22.82	51.27	68.23	67.56	67.82
mT5-CL (k=64)	20.19	13.37	6.06	3.49	18.13	36.14	64.11	62.97	63.44
mT5-MTL (k=128)	12.27	18.9	6.83	3.93	21.12	46.35	68.35	67.96	68.04
mT5-CL (k=128)	19.39	13.54	5.99	3.53	18.71	37.62	64.12	63.24	63.52
mT5-MTL (k=512)	9.84	21.75	7.75	4.49	21.23	47.16	68.94	68.25	68.57
mT5-CL (k=512)	14.56	17.33	6.67	3.97	20.03	42.38	65.51	65.13	65.22
mT5-MTL (k=1024)	8.92	22.98	8.25	4.85	18.65	39.66	69.37	68.60	68.85
mT5-CL (k=1024)	12.37	20.24	7.35	4.18	20.95	44.59	68.83	67.72	68.82

CL: Cross-Lingual transfer learning, MTL: Multitask learning, k: number of examples

Table 5.4: Detailed performance of mT5 model on the different techniques used

performance with more training data. SacreBLEU scores also improve with more examples, with mT5-MTL (k=1024) achieving the highest scores of 22.98 (B-1), 8.25 (B-2), and an average score of 4.85. The diversity of generated text, measured by Distinct-1 and Distinct-2, shows some variability. The highest Distinct-1 (22.82%) and Distinct-2 (51.27%) scores were observed with mT5-MTL (k=64), though diversity generally remains high across different k values. Bertscore metrics improve with more examples, with the highest F-1 score (68.85%) achieved by mT5-MTL (k=1024), indicating a better balance between precision and recall in generating high-quality text. Overall, the multitask learning approach demonstrates that increasing the number of examples leads to better performance in perplexity, translation quality, and Bertscore metrics, though diversity metrics show some fluctuations. It is really interesting, that while we increase the number of samples, the diversity metric seems to decrease as the model is probably learning more specific language templates to generate answers.

In contrast, cross-lingual transfer learning for the mT5 model shows significant improvements across various metrics as the number of examples increases, though it still falls short of the performance seen with multitask learning. For example, perplexity decreases with more data, with mT5-CL (k=1024) achieving a perplexity of 12.37. In terms of SacreBleu, mT5-CL (k=1024) achieved scores of 20.24 (B-1), 7.35 (B-2), and an average score of 4.18. Big difference we can see on the diversity metric Distinct-N. MTL models generally achieve higher Distinct-1 and Distinct-2 scores. For instance, at k=512, MTL achieves Distinct-1 of 21.23% and Distinct-2 of 47.16%, compared to TL's 20.03% and 42.38%. This indicates that MTL is more effective in generating diverse text outputs, which is crucial for applications requiring rich and varied language generation.

Model	perplexity	SacreBLEU			Distinct-N		Bertscore		
		B-1	B-2	average-score	Distinct-1 (%)	Distinct-2 (%)	Precision (%)	Recall (%)	F-1 (%)
XGLM-MTL (k=32)	18.06	19.75	8.85	4.06	17.06	36.21	69.26	67.99	68.41
XGLM-CL (k=32)	20.33	19.98	9.38	3.29	22.54	48.68	68.86	67.97	68.36
XGLM-MTL (k=64)	17.34	20.93	9.29	4.36	17.64	38.46	69.34	67.88	68.52
XGLM-CL (k=64)	17.61	20.41	8.99	4.21	19.78	43.04	69.23	67.98	68.51
XGLM-MTL (k=128)	16.53	23.25	9.89	4.75	18.24	40.39	69.53	68.25	68.75
XGLM-CL (k=128)	15.74	23.61	10.03	4.95	18.75	41.91	69.60	68.32	68.99
XGLM-MTL (k=512)	14.38	25.35	11.12	5.51	18.85	41.17	69.41	67.95	68.57
XGLM-CL (k=512)	14.16	25.44	11.31	5.35	19.17	43.45	69.92	68.60	69.10
XGLM-MTL (k=1024)	13.53	26.49	11.66	5.78	19.21	41.97	69.51	67.96	68.60
XGLM-CL (k=1024)	13.31	26.13	11.42	5.91	19.11	42.88	69.26	68.60	69.23

CL: Cross-Lingual transfer learning, MTL: Multitask learning, k: number of examples

Table 5.5: Detailed performance of XGLM

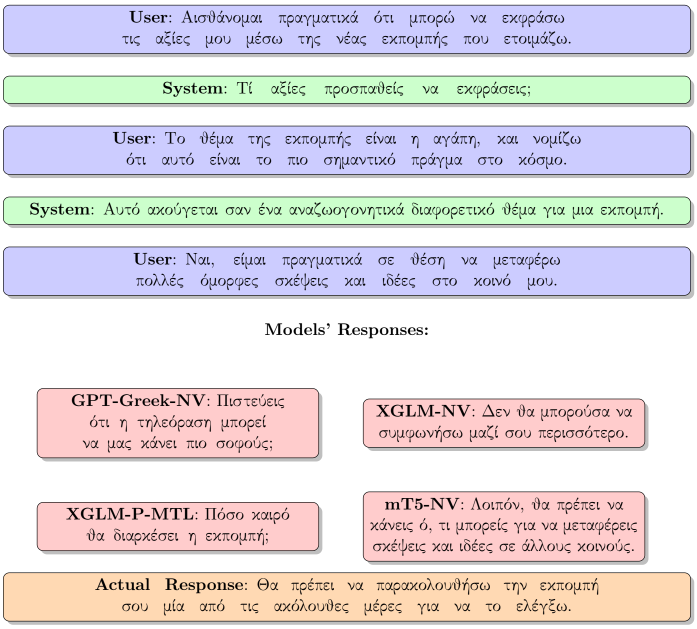
Turning to the XGLM model, a similar trend is observed. Multitask learning shows better performance compared to cross-lingual transfer learning across various metrics. For example, the lowest perplexity for XGLM-MTL (k=1024) is 13.53, and the highest SacreBLEU scores are 26.49 for B-1, 11.66 for B-2, and an average score of 5.78. These scores are higher compared to those of mT5, suggesting that XGLM generates text closer to the reference answers.

Distinct-N scores for XGLM-MTL also show a general improvement with more examples, although the diversity metrics do not increase as significantly as they do for mT5. The highest diversity is observed with lower values of k, but as k increases, the diversity metrics stabilize, reflecting a balance between specific language templates and varied outputs.

BERTScore metrics for XGLM demonstrate high precision, recall, and F-1 scores, with the highest F-1 score being 69.23% for XGLM-CL (k=1024). This is slightly higher than the F-1 scores observed for mT5, indicating that XGLM may be better at generating semantically similar text to the reference.

In conclusion, the ablation study reveals that multitask learning is more effective than cross-lingual transfer learning for both mT5 and XGLM models across most metrics, including perplexity, SacreBLEU, and BERTScore. The diversity of generated text is also generally higher with MTL, which is crucial for applications requiring rich and varied language generation. Increasing the number of examples leads to improved performance, highlighting the importance of leveraging more training data in these models.

5.5.5 Output Samples



User: Αισθάνομαι πραγματικά ότι μπορώ να εκφράσω τις αξίες μου μέσω της νέας εκπομπής που ετοιμάζω.

System: Τί αξίες προσπαθείς να εκφράσεις;

User: Το θέμα της εκπομπής είναι η αγάπη, και νομίζω ότι αυτό είναι το πιο σημαντικό πράγμα στο κόσμο.

System: Αυτό ακούγεται σαν ένα αναζωογονητικά διαφορετικό θέμα για μια εκπομπή.

User: Ναι, είμαι πραγματικά σε θέση να μεταφέρω πολλές όμορφες σκέψεις και ιδέες στο κοινό μου.

Models' Responses:

GPT-Greek-NV: Πιστεύεις ότι η τηλεόραση μπορεί να μας κάνει πιο σοφούς;

XGLM-NV: Δεν θα μπορούσα να συμφωνήσω μαζί σου περισσότερο.

XGLM-P-MTL: Πόσο καιρό θα διαρκέσει η εκπομπή;

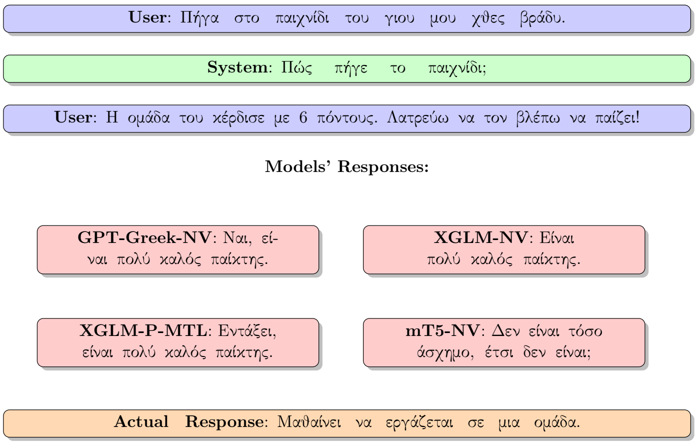
mT5-NV: Λοιπόν, θα πρέπει να κάνεις ό, τι μπορείς για να μεταφέρεις σκέψεις και ιδέες σε άλλους κοινούς.

Actual Response: Θα πρέπει να παρακολουθήσω την εκπομπή σου μία από τις ακόλουθες μέρες για να το ελέγξω.

Figure 5.1: Dialogue example with multiple system responses

Since the task we are examining is a generation task, we present some of the responses generated by the models GPT2-Greek-NV, XGLM-NV, XGLM-P-MTL, mT5-NV, and mT5-P-MTL in the following examples. We utilize manually translated examples from the Commonsense Dialogues dataset [96] as the test data. Each model is provided with the same dialogue history up to a certain point, and we analyze their respective generated responses. These responses are then visually compared to the actual dataset responses through Figures 5.1, 5.2, and 5.3. This comparison aims to assess the coherence and relevance of the responses produced by each model.

In Example 5.1, where the user discusses expressing personal values through a new show focused on the theme of love, the responses from the models varied in relevance and depth. GPT-Greek-NV's question about the educative potential of television. XGLM-NV and XGLM-P-MTL provided generic responses, with XGLM-P-MTL at least asking a relevant question about the show's duration. mT5-NV provided the most aligned response, encouraging the user to spread thoughts and ideas, resonating well with the intent to express values, having however a slight grammatical error.



User: Πήγα στο παιχνίδι του γιου μου χθες βράδυ.

System: Πώς πήγε το παιχνίδι;

User: Η ομάδα του κέρδισε με 6 πόντους. Λατρεύω να τον βλέπω να παίζει!

Models' Responses:

GPT-Greek-NV: Ναι, είναι πολύ καλός παίκτης.

XGLM-NV: Είναι πολύ καλός παίκτης.

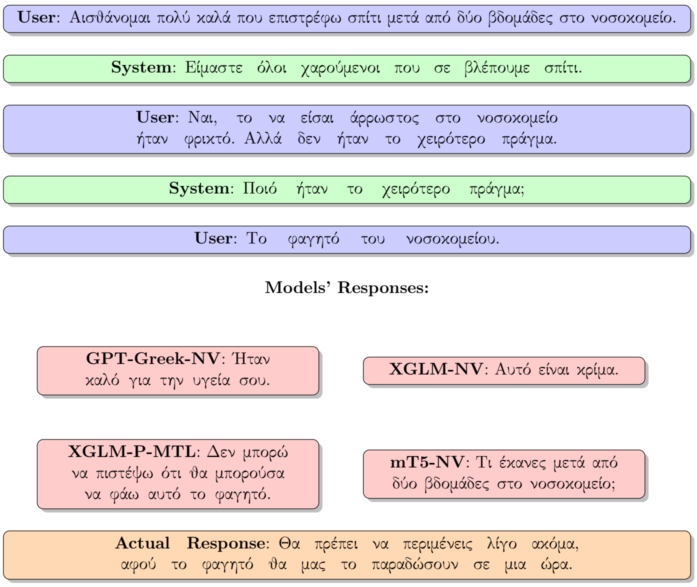
XGLM-P-MTL: Εντάξει, είναι πολύ καλός παίκτης.

mT5-NV: Δεν είναι τόσο άσχημο, έτσι δεν είναι;

Actual Response: Μαθαίνει να εργάζεται σε μια ομάδα.

Figure 5.2: Dialogue example 2 with multiple system responses

In Example 2, which revolves around the user's son's game, the responses were generally supportive but lacked depth. GPT-Greek-NV, XGLM-NV, and XGLM-P-MTL all commented positively on the son's playing skills, reinforcing a supportive stance but adding little to the conversation. mT5-NV introduced a slight negative spin which, while different, might be perceived as less supportive. In general, the responses on this example were quite coherent but lacked engagement.



User: Αισθάνομαι πολύ καλά που επιστρέφω σπίτι μετά από δύο βδομάδες στο νοσοκομείο.

System: Είμαστε όλοι χαρούμενοι που σε βλέπουμε σπίτι.

User: Ναι, το να είσαι άρρωστος στο νοσοκομείο ήταν φρικτό. Αλλά δεν ήταν το χειρότερο πράγμα.

System: Ποιό ήταν το χειρότερο πράγμα;

User: Το φαγητό του νοσοκομείου.

Models' Responses:

GPT-Greek-NV: Ήταν καλό για την υγεία σου.

XGLM-NV: Αυτό είναι κρίμα.

XGLM-P-MTL: Δεν μπορώ να πιστέψω ότι θα μπορούσα να φάω αυτό το φαγητό.

mT5-NV: Τι έκανες μετά από δύο βδομάδες στο νοσοκομείο;

Actual Response: Θα πρέπει να περιμένεις λίγο ακόμα, αφού το φαγητό θα μας το παραδώσουν σε μια ώρα.

Figure 5.3: Dialogue example 3 with multiple system responses

Example 3 presented a discussion on the user's return from the hospital. Here, GPT-Greek-NV focuses on the benefit of the hospital's food on the user's health. XGLM-NV's expression of sympathy was appropriate, while XGLM-P-MTL showed the strongest empathy by directly commenting on the poor quality of hospital food, aligning well with the user's feelings. mT5-NV, however, missed the mark by asking an unrelated question.

The effectiveness of the responses varied across different models, with mT5-NV and XGLM-P-MTL often providing more contextually appropriate and engaging responses. GPT-Greek-NV tended to offer more diverse responses, considering the context of previous turns in the dialogue. Notably, the XGLM model, which was trained with a subset of prompts, approximately 1% of the total Greek dialogues compared to other models trained on fully translated sets—produced responses that were on par with other models in terms of quality and syntactical correctness. Overall, this analysis demonstrates that models yield the best performance when their responses are closely aligned with the user's emotional context and the specific content of the ongoing discussion.

5.6 Human Evaluation

Automatic metrics, while useful for quantitative analysis, often fall short in capturing the nuanced aspects of human dialogue. To gain a more comprehensive understanding of model performance, we conducted human evaluations through an online survey, which allowed us to assess qualitative aspects that metrics alone cannot measure effectively. Building on our automatic evaluation findings from Section 5.5.5, we selected models that demonstrated the most promising results for human assessment: GPT-Greek-NV, XGLM-NV, and XGLM-P-MTL. Additionally, we included Meltemi, a substantially larger model (7B parameters compared to the others' 550M), to benchmark our models against a more capable architecture and understand the performance gap between different model scales. In the survey, participants were presented with identical dialogue histories and asked to evaluate responses generated by each of the four models based on the following criteria:

1. Fluency: Is the generated response correct syntactically and feels natural?

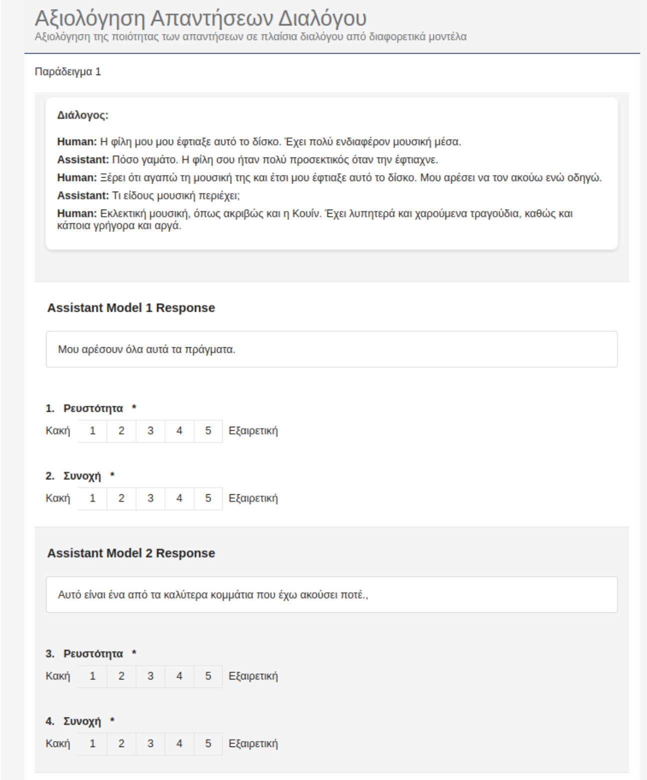
2. Coherence: Is the generated response relevant to the dialogue history?

A total of 40 participants were involved in the study: 72.5% were aged between 20 and 30, 27.5% between 30 and 40. Each participant was assigned 5 random set of dialogues-responses resulting to a total of 180 evaluations per model. Participants were asked to rate each generated response using a 1-5 Likert scale, where 5 is the best score. Detailed instructions were provided at the beginning of the survey on how to determine if an answer is fluent or coherent. Figure 5.4 shows an example of the User Interface (UI) of the web application used to conduct the survey.

Model	Fluency	Coherence
GPT-Greek-NV	3.42	2.90
XGLM-NV	3.13	2.62
XGLM-P-MTL	3.46*	2.98*
Meltemi	4.01*	3.97*

Table 5.6: Comparison of Models on Fluency and Coherence. Results noted with * are statistically significant with $p<0.05$ using the MannWitney U test.

Table 5.6 show the comparison of response ratings on the different models. Meltemi demonstrates superior performance compared to all other models, achieving the highest scores in both fluency (4.01) and coherence (3.97). These improvements are statistically significant ($p<0.05$), suggesting that Meltemi produces responses that human evaluators find substantially more natural and contextually appropriate than the alternatives. Something expected, considering the much bigger size and more extensive training.



Αξιολόγηση Απαντήσεων Διαλόγου

Αξιολόγηση της ποιότητας των απαντήσεων σε πλαίσια διαλόγου από διαφορετικά μοντέλα

Παράδειγμα 1

Διάλογος:

Human: Η φίλη μου μου έφτιαξε αυτό το δίσκο. Έχει πολύ ενδιαφέρον μουσική μέσα.

Assistant: Πόσο γαμάτο. Η φίλη σου ήταν πολύ προσεκτικός όταν την έφτιαχνε.

Human: Ξέρει ότι αγαπώ τη μουσική της και έτσι μου έφτιαξε αυτό το δίσκο. Μου αρέσει να τον ακούω ενώ οδηγώ.

Assistant: Τι είδους μουσική περιέχει;

Human: Εκλεκτική μουσική, όπως ακριβώς και η Κουίν. Έχει λυπητερά και χαρούμενα τραγούδια, καθώς και κάποια γρήγορα και αργά.

Assistant Model 1 Response

Μου αρέσουν όλα αυτά τα πράγματα.

1. **Ρευστότητα** *

Κακή 1 2 3 4 5 Εξαιρετική

2. **Συνοχή** *

Κακή 1 2 3 4 5 Εξαιρετική

Assistant Model 2 Response

Αυτό είναι ένα από τα καλύτερα κομμάτια που έχω ακούσει ποτέ.,

3. **Ρευστότητα** *

Κακή 1 2 3 4 5 Εξαιρετική

4. **Συνοχή** *

Κακή 1 2 3 4 5 Εξαιρετική

Figure 5.4: Example from the human evaluation survey setup. At the top, there is an example of a 5-turn dialogue. Then, is provided the answer of each model, and possible ratings for the user.

XGLM-P-MTL shows moderate improvements over its XGLM-NV, with statistically significant gains in both fluency (3.46 vs. 3.13) and coherence (2.98 vs. 2.62). This indicates that the multitask training approach using english prompts employed in XGLM-P-MTL effectively enhances the quality of generated text, using much less Greek annotated data.

GPT-Greek-NV performs relatively well on fluency (3.42) but shows limitations in coherence (2.90), suggesting that while the model can produce grammatically sound text, it struggles more with maintaining contextual relevance throughout longer responses. The gap between fluency and coherence scores across all models indicates that achieving contextual consistency remains more challenging than producing grammatically correct text.

5.7 Summary

In this chapter, we explored the development of open-domain dialogue models specifically designed for the Greek language, addressing the unique challenges of working with limited linguistic resources. We started by looking at the history and evolution of dialogue systems, from traditional rule-based methods to modern neural network-based approaches. In Section 5.2, we reviewed key research on open-domain dialogue generation models and discussed the specific considerations needed for languages with fewer resources.

In Section 5.3 we introduced the dataset used in our experiments, which was essential for training our models given the limited availability of Greek conversational data, and next we discussed the model architectures we adapted to train a model in the Greek language.

In Section 5.5, we detailed our methodological approach, including specific adjustments made to train effectively with limited data and various training techniques employed. We provided an in-depth analysis of our results and examined how different models performed in generating meaningful and coherent responses, considering both automated evaluation metrics and specific generation examples. While our best models produced responses that were generally fluent and coherent, we observed inconsistencies, with some outputs being dull and lacking engagement.

Beyond automated metrics, we conducted human evaluations to assess model performance from a user perspective, focusing on fluency and coherence as shown in Table 5.6. These evaluations revealed that Meltemi model significantly outperformed other approaches, achieving the highest ratings in both fluency (4.01) and coherence (3.97), something expected considering the higher model size, and exposure to bigger amount of Greek data. Among the remaining models, XGLM-P-MTL demonstrated superior performance, highlighting the effectiveness of our bilingual training approach using a common prompt between the 2 languages. This model achieved notable results despite utilizing significantly less Greek data, illustrating the powerful cross-linguistic knowledge transfer that can occur between languages.

Chapter 6

Conclusion and Future Work

6.1 Thesis summary and contributions

In this diploma thesis, we we studied in depth the work done in the field of open-domain dialogue systems for low-resource languages. And specifically we examined different methods and proposed ideas of creating such systems in the Greek language. More specifically, at first, we analyzed the traditional architectures used for dialogue generation. Then, we studied the state-of-the-art models that can be used in dialogue generation, including the Vaswani encoder-decoder transformer, the Bert, the GPT2 and the T5 models. After providing a theoretical background for the aforementioned models we focused on methods applicable to low-resource languages for dialogue generation and similar tasks.

After presenting and studying the related work, we addressed the unique challenges presented by the limited availability of training data and the scarcity of pre-trained language models for such languages. A significant challenge in our study was the lack of a suitable Greek dialogue dataset. To overcome this, we employed machine translation to create a Greek version of the Daily Dialog dataset, enabling us to conduct our experiments.

Our research involved a comprehensive series of experiments utilizing a variety of monolingual and multilingual transformer-based models. These included GREEK-BERT, GPT-2 Greek, mT5, and XGLM. We investigated different training approaches to leverage the limited resources effectively. These approaches encompassed zero-shot, and few-shot cross-lingual training, as well as native training. Furthermore, we explored the use of a prompt learning technique to enhance the performance of our multilingual models, demonstrating its effectiveness in improving dialogue generation.

We evaluated the models using several automatic metrics: Perplexity, BLEU, BertScore, and Distinct-n. These metrics helped assess the quality, diversity, and relevance of generated responses. Our evaluation revealed that native training generally outperformed other techniques, with XGLM-P-MTL being the only comparable model. This model was trained concurrently on English dialogue data and a small portion of Greek data using consistent prompts across languages. To further assess performance and compare with larger models, we conducted a human evaluation survey comparing the three best-performing models according to automatic metrics (GPT-Greek-NV, XGLM-NV, and XGLM-P-MTL) alongside a more advanced autoregressive model, Meltemi. The survey results showed that Meltemi outperformed all other models, followed by XGLM-P-MTL, which demonstrated statistically significant improvement over the XGLM model trained solely on translated Greek data.

The findings of our experiments offer valuable insights into the complexities of dialogue generation in low-resource languages. Our results underscore the potential of cross-lingual transfer learning as a viable strategy for such scenarios when couple with some prompt learning, providing a pathway for future research and development.

6.2 Future Work

This thesis establishes methodologies for dialogue systems in low-resource languages, opening several interconnected research directions that could significantly advance conversational AI accessibility for underrepresented communities.

The first step is improving data creation beyond the machine translation methods shown here. Techniques like paraphrasing and creating synthetic dialogues could build better training datasets while keeping the language natural. These improved datasets would directly support developing specialized prompting methods for dialogue systems.

Future prompt engineering should focus on dialogue-specific approaches that go beyond general prompting techniques. Few-shot learning paradigms show particular promise for transferring knowledge from high-resource languages, but these need careful design to capture dialogue patterns like turn-taking, response coherence, and context maintenance. Cross-lingual prompting strategies should also explore how to adapt conversation styles and cultural communication patterns across different languages. Additionally, investigating dynamic prompting that adjusts based on conversation context could improve response quality. Template-based prompting for common dialogue scenarios (greetings, requests, clarifications) could provide more consistent performance while maintaining flexibility for diverse conversational situations.

However, as these methodologies become more sophisticated, comprehensive evaluation becomes increasingly critical. Future work must expand beyond current metrics and tasks to assess model performance across diverse dialogue domains. This expanded evaluation scope necessitates robust testing against real-world challenges including code-switching, dialectal variations, and informal language use.

The methodologies of this thesis can can guide work on other low-resource languages. This scaling process requires careful investigation of which components are language-agnostic versus language-specific, with particular emphasis on languages having minimal NLP representation. Such extension efforts must consider computational constraints, as many target communities have limited access to advanced hardware. Therefore, developing parameter-efficient fine-tuning methods, knowledge distillation techniques, and hardware optimization strategies becomes crucial for practical development.

Ultimately, these technical advances must serve real community needs. Future research should prioritize sustainable solutions, ensuring that dialogue systems remain accessible and beneficial for linguistically underrepresented populations. The methodologies presented in this thesis serves as a foundation for addressing the challenges of building effective dialogue systems for linguistically underrepresented communities.

Bibliography

[1] D. Adiwardana, M.-T. Luong, D. R. So, *et al.*, *Towards a Human-like Open-Domain Chatbot*, 2020. DOI: 10.48550/arXiv.2001.09977. arXiv: 2001.09977.

[2] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding", in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, and H. Mubarak, Eds., Marseille, France: European Language Resource Association, 2020, pp. 9–15, ISBN: 979-10-95546-51-1.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, 2016. DOI: 10.48550/arXiv.1607.06450. arXiv: 1607.06450.

[4] D. Bahdanau, K. Cho, and Y. Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate*, 2016. arXiv: 1409.0473.

[5] E. Ben-David, N. Oved, and R. Reichart, *PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains*, 2022. DOI: 10.48550/arXiv.2102.12206. arXiv: 2102.12206.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A Neural Probabilistic Language Model",

[7] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent", in *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds., Heidelberg: Physica-Verlag HD, 2010, pp. 177–186, ISBN: 978-3-7908-2604-3. DOI: 10.1007/978-3-7908-2604-3_16.

[8] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language Models are Few-Shot Learners*, 2020. arXiv: 2005.14165.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, *et al.*, *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, 2014. DOI: 10.48550/arXiv.1406.1078. arXiv: 1406.1078.

[10] K. M. Colby, "Modeling a paranoid mind", *Behavioral and Brain Sciences*, vol. 4, no. 4, pp. 515–534, 1981. DOI: 10.1017/S0140525X00000030.

[11] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, *Unsupervised Cross-lingual Representation Learning at Scale*, 2020. DOI: 10.48550/arXiv.1911.02116. arXiv: 1911.02116.

[12] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, ISSN: 1573-0565. DOI: 10.1007/BF00994018.

[13] M. Deng, J. Wang, C.-P. Hsieh, *et al.*, *RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning*, 2022. DOI: 10.48550/arXiv.2205.12548. arXiv: 2205.12548.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019. arXiv: `1810.04805`.

[15] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, *Wizard of Wikipedia: Knowledge-Powered Conversational agents*, 2019. arXiv: `1811.01241`.

[16] N. Dziri, A. Madotto, O. Zaïane, and A. J. Bose, "Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding", in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2197–2214. DOI: `10.18653/v1/2021.emnlp-main.168`.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative Adversarial Networks*, 2014. DOI: `10.48550/arXiv.1406.2661`. arXiv: `1406.2661`.

[18] A. Graves, G. Wayne, and I. Danihelka, *Neural Turing Machines*, 2014. DOI: `10.48550/arXiv.1410.5401`. arXiv: `1410.5401`.

[19] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, *PTR: Prompt Tuning with Rules for Text Classification*, 2021. DOI: `10.48550/arXiv.2105.11259`. arXiv: `2105.11259`.

[20] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, 2015. DOI: `10.48550/arXiv.1512.03385`. arXiv: `1512.03385`.

[21] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios", in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Online: Association for Computational Linguistics, 2021, pp. 2545–2568. DOI: `10.18653/v1/2021.naacl-main.201`.

[22] D. Hendrycks and K. Gimpel, *Gaussian Error Linear Units (GELUs)*, 2023. DOI: `10.48550/arXiv.1606.08415`. arXiv: `1606.08415`.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, ISSN: 0899-7667. DOI: `10.1162/neco.1997.9.8.1735`.

[24] T. Ide and D. Kawahara, "Multi-Task Learning of Generation and Classification for Emotion-Aware Dialogue Response Generation", in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, E. Durmus, V. Gupta, N. Liu, N. Peng, and Y. Su, Eds., Online: Association for Computational Linguistics, 2021, pp. 119–125. DOI: `10.18653/v1/2021.naacl-srw.15`.

[25] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, *Mistral 7B*, 2023. DOI: `10.48550/arXiv.2310.06825`. arXiv: `2310.06825`.

[26] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, *How Can We Know What Language Models Know?*, 2020. DOI: `10.48550/arXiv.1911.12543`. arXiv: `1911.12543`.

[27] T. Kasahara, D. Kawahara, N. Tung, S. Li, K. Shinzato, and T. Sato, "Building a Personalized Dialogue System with Prompt-Tuning", in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, D. Ippolito, L. H. Li, M. L. Pacheco, D. Chen, and N. Xue, Eds., Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, 2022, pp. 96–105. DOI: `10.18653/v1/2022.naacl-srw.13`.

[28] A. Khodadadi, S. A. Hosseini, E. Pajouheshgar, F. Mansouri, and H. R. Rabiee, *ChOracle: A Unified Statistical Framework for Churn Prediction*, 2019. DOI: `10.48550/arXiv.1909.06868`. arXiv: `1909.06868`.

[29] S. Kim, J. Y. Jang, M. Jung, and S. Shin, "A Model of Cross-Lingual Knowledge-Grounded Response Generation for Open-Domain Dialogue Systems", in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 352–365. DOI: `10.18653/v1/2021.findings-emnlp.33`.

[30] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, *GREEK-BERT: The Greeks visiting Sesame Street*, 2020. DOI: `10.1145/3411408.3411440`. arXiv: `2008.12014`.

[31] A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš, *From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers*, 2020. arXiv: `2005.00633`.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, ISSN: 1558-2256. DOI: `10.1109/5.726791`.

[33] B. Lester, R. Al-Rfou, and N. Constant, *The Power of Scale for Parameter-Efficient Prompt Tuning*, 2021. DOI: `10.48550/arXiv.2104.08691`. arXiv: `2104.08691`.

[34] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, *A Diversity-Promoting Objective Function for Neural Conversation Models*, 2016. arXiv: `1510.03055`.

[35] Q. Li, P. Li, Z. Ren, P. Ren, and Z. Chen, *Knowledge Bridging for Empathetic Dialogue Generation*, 2021. arXiv: `2009.09708`.

[36] X. L. Li and P. Liang, *Prefix-Tuning: Optimizing Continuous Prompts for Generation*, 2021. DOI: `10.48550/arXiv.2101.00190`. arXiv: `2101.00190`.

[37] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset", in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, G. Kondrak and T. Watanabe, Eds., Taipei, Taiwan: Asian Federation of Natural Language Processing, 2017, pp. 986–995.

[38] *Lighteternal/gpt2-finetuned-greek · Hugging Face*, https://huggingface.co/lighteternal/gpt2-finetuned-greek.

[39] X. V. Lin, T. Mihaylov, M. Artetxe, *et al.*, *Few-shot Learning with Multilingual Language Models*, 2022. DOI: `10.48550/arXiv.2112.10668`. arXiv: `2112.10668`.

[40] Z. Lin, Z. Liu, G. I. Winata, *et al.*, "XPersona: Evaluating Multilingual Personalized Chatbot", in *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, Online: Association for Computational Linguistics, 2021, pp. 102–112. DOI: `10.18653/v1/2021.nlp4convai-1.10`.

[41] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, *How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation*, 2017. arXiv: `1603.08023`.

[42] L. Liu and J. X. Huang, *Prompt Learning to Mitigate Catastrophic Forgetting in Cross-lingual Transfer for Open-domain Dialogue Generation*, 2023. arXiv: `2305.07393`.

[43] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*, 2021. arXiv: `2107.13586`.

[44] X. Liu, K. Ji, Y. Fu, *et al.*, *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*, 2022. DOI: 10.48550/arXiv.2110.07602. arXiv: 2110.07602.

[45] X. Liu, Y. Zheng, Z. Du, *et al.*, *GPT Understands, Too*, 2023. DOI: 10.48550/arXiv.2103.10385. arXiv: 2103.10385.

[46] Y. Liu, M. Ott, N. Goyal, *et al.*, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, 2019. arXiv: 1907.11692.

[47] M.-T. Luong, H. Pham, and C. D. Manning, *Effective Approaches to Attention-based Neural Machine Translation*, 2015. DOI: 10.48550/arXiv.1508.04025. arXiv: 1508.04025.

[48] A. Madotto, Z. Lin, G. I. Winata, and P. Fung, *Few-Shot Bot: Prompt-Based Learning for Dialogue Systems*, 2021. arXiv: 2110.08118.

[49] A. Magooda and D. Litman, *Abstractive Summarization for Low Resource Data using Domain Transfer and Data Synthesis*, 2020. arXiv: 2002.03407.

[50] H. H. Mao, B. P. Majumder, J. McAuley, and G. Cottrell, "Improving Neural Story Generation by Targeted Common Sense Grounding", in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5987–5992. DOI: 10.18653/v1/D19-1615.

[51] Z. Mao, C. Chu, and S. Kurohashi, "Linguistically-driven Multi-task Pre-training for Low-resource Neural Machine Translation", *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, pp. 1–29, 2022, ISSN: 2375-4699, 2375-4702. DOI: 10.1145/3491065. arXiv: 2201.08070.

[52] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model", in *Interspeech 2010*, ISCA, 2010, pp. 1045–1048. DOI: 10.21437/Interspeech.2010-343.

[53] T. Naous, W. Antoun, R. Mahmoud, and H. Hajj, "Empathetic BERT2BERT Conversational Model: Learning Arabic Language Generation with Little Data", in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, N. Habash, H. Bouamor, H. Hajj, *et al.*, Eds., Kyiv, Ukraine (Virtual): Association for Computational Linguistics, 2021, pp. 164–172.

[54] T. Naous, C. Hokayem, and H. Hajj, "Empathy-driven Arabic Conversational Chatbot", in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, I. Zitouni, M. Abdul-Mageed, H. Bouamor, *et al.*, Eds., Barcelona, Spain (Online): Association for Computational Linguistics, 2020, pp. 58–68.

[55] OpenAI, *Chatgpt*, OpenAI, San Francisco, https://chat.openai.com, 2024.

[56] A. Otegi, A. Agirre, J. A. Campos, A. Soroa, and E. Agirre, "Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque", in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, *et al.*, Eds., Marseille, France: European Language Resources Association, 2020, pp. 436–442, ISBN: 979-10-95546-34-4.

[57] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: A Method for Automatic Evaluation of Machine Translation”, in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 311–318. DOI: `10.3115/1073083.1073135`.

[58] M. Post, *A Call for Clarity in Reporting BLEU Scores*, 2018. arXiv: `1804.08771`.

[59] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training”,

[60] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners”,

[61] C. Raffel, N. Shazeer, A. Roberts, *et al.*, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, 2023. arXiv: `1910.10683`.

[62] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, *Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset*, 2019. arXiv: `1811.00207`.

[63] S. Roller, E. Dinan, N. Goyal, *et al.*, *Recipes for building an open-domain chatbot*, 2020. DOI: `10.48550/arXiv.2004.13637`. arXiv: `2004.13637`.

[64] I. V. Serban, C. Sankar, M. Germain, *et al.*, *A Deep Reinforcement Learning Chatbot*, 2017. arXiv: `1709.02349`.

[65] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, *Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models*, 2016. arXiv: `1507.04808`.

[66] I. V. Serban, A. Sordoni, R. Lowe, *et al.*, *A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues*, 2016. arXiv: `1605.06069`.

[67] F. Shamsafar and H. Ebrahimnezhad, “Uniting holistic and part-based attitudes for accurate and robust deep human pose estimation”, *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2339–2353, 2021, ISSN: 1868-5145. DOI: `10.1007/s12652-020-02347-7`.

[68] L. Shang, Z. Lu, and H. Li, “Neural Responding Machine for Short-Text Conversation”, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China: Association for Computational Linguistics, 2015, pp. 1577–1586. DOI: `10.3115/v1/P15-1152`.

[69] L. Shen, S. Yu, and X. Shen, *Is Translation Helpful? An Empirical Analysis of Cross-Lingual Transfer in Low-Resource Dialog Generation*, 2023. arXiv: `2305.12480`.

[70] A. Sordoni, M. Galley, M. Auli, *et al.*, “A Neural Network Approach to Context-Sensitive Generation of Conversational Responses”, in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado: Association for Computational Linguistics, 2015, pp. 196–205. DOI: `10.3115/v1/N15-1020`.

[71] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”,

[72] R. Thoppilan, D. De Freitas, J. Hall, *et al.*, *LaMDA: Language Models for Dialog Applications*, https://arxiv.org/abs/2201.08239v3, 2022.

[73] J. Tiedemann and S. Thottingal, "OPUS-MT – Building open translation services for the World", in *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, A. Martins, H. Moniz, S. Fumega, *et al.*, Eds., Lisboa, Portugal: European Association for Machine Translation, 2020, pp. 479–480.

[74] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention Is All You Need*, 2017. DOI: `10.48550/arXiv.1706.03762`. arXiv: `1706.03762`.

[75] A. Venkatesh, C. Khatri, A. Ram, *et al.*, *On Evaluating and Comparing Open Domain Dialog Systems*, 2018. DOI: `10.48550/arXiv.1801.03625`. arXiv: `1801.03625`.

[76] O. Vinyals and Q. Le, *A Neural Conversational Model*, 2015. arXiv: `1506.05869`.

[77] L. Voukoutis, D. Roussis, G. Paraskevopoulos, *et al.*, *Meltemi: The first open Large Language Model for Greek*, 2024. DOI: `10.48550/arXiv.2407.20743`. arXiv: `2407.20743`.

[78] T. Vu, B. Lester, N. Constant, R. Al-Rfou, and D. Cer, *SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer*, 2022. DOI: `10.48550/arXiv.2110.07904`. arXiv: `2110.07904`.

[79] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine", *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966, ISSN: 0001-0782. DOI: `10.1145/365153.365168`.

[80] O. Weller, K. Seppi, and M. Gardner, *When to Use Multi-Task Learning vs Intermediate Fine-Tuning for Pre-Trained Encoder Transfer Learning*, 2022. arXiv: `2205.08124`.

[81] G. I. Winata, A. Madotto, Z. Lin, *et al.*, "CAiRE_HKUST at SemEval-2019 Task 3: Hierarchical Attention for Dialogue Emotion Classification", in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 142–147. DOI: `10.18653/v1/S19-2021`.

[82] Y. Wu, W. Wu, Z. Li, and M. Zhou, *Response Selection with Topic Clues for Retrieval-based Chatbots*, 2016. arXiv: `1605.00090`.

[83] L. Xiang, J. Zhu, Y. Zhao, Y. Zhou, and C. Zong, "Robust Cross-lingual Task-oriented Dialogue", *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 6, pp. 1–24, 2021, ISSN: 2375-4699, 2375-4702. DOI: `10.1145/3457571`.

[84] P. Xu, A. Madotto, C.-S. Wu, J. H. Park, and P. Fung, "Emo2Vec: Learning Generalized Emotion Representation by Multi-task Training", in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 292–298. DOI: `10.18653/v1/W18-6243`.

[85] L. Xue, N. Constant, A. Roberts, *et al.*, *mT5: A massively multilingual pre-trained text-to-text transformer*, 2021. DOI: `10.48550/arXiv.2010.11934`. arXiv: `2010.11934`.

[86] Z. Yang, W. Wu, C. Xu, *et al.*, "StyleDGPT: Stylized Response Generation with Pre-trained Language Models", in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, 2020, pp. 1548–1559. DOI: `10.18653/v1/2020.findings-emnlp.140`.

[87] Z. Yang, W. Wu, J. Yang, C. Xu, and Z. Li, “Low-Resource Response Generation with Template Prior”, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1886–1897. DOI: `10.18653/v1/D19-1197`. arXiv: `1909.11968`.

[88] Q. Zhang, X. Shen, E. Chang, J. Ge, and P. Chen, *MDIA: A Benchmark for Multilingual Dialogue Generation in 46 Languages*, 2022. arXiv: `2208.13078`.

[89] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, “Personalizing Dialogue Agents: I have a dog, do you have pets too?”, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 2204–2213. DOI: `10.18653/v1/P18-1205`.

[90] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *BERTScore: Evaluating Text Generation with BERT*, 2020. DOI: `10.48550/arXiv.1904.09675`. arXiv: `1904.09675`.

[91] Y. Zhang, S. Sun, M. Galley, *et al.*, “DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation”, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online: Association for Computational Linguistics, 2020, pp. 270–278. DOI: `10.18653/v1/2020.acl-demos.30`.

[92] J. Zhao, R. Li, Q. Jin, X. Wang, and H. Li, *MEmoBERT: Pre-training Model with Prompt-based Learning for Multimodal Emotion Recognition*, 2021. DOI: `10.48550/arXiv.2111.00865`. arXiv: `2111.00865`.

[93] X. Zhao, W. Wu, C. Xu, C. Tao, D. Zhao, and R. Yan, “Knowledge-Grounded Dialogue Generation with Pre-trained Language Models”, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, 2020, pp. 3377–3390. DOI: `10.18653/v1/2020.emnlp-main.272`.

[94] Y. Zheng, R. Zhang, M. Huang, and X. Mao, “A Pre-Training Based Personalized Dialogue Generation Model with Persona-Sparse Data”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 9693–9700, 2020, ISSN: 2374-3468, 2159-5399. DOI: `10.1609/aaai.v34i05.6518`.

[95] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, *The Design and Implementation of XiaoIce, an Empathetic Social Chatbot*, 2019. arXiv: `1812.08989`.

[96] P. Zhou, K. Gopalakrishnan, B. Hedayatnia, *et al.*, “Commonsense-Focused Dialogues for Response Generation: An Empirical Study”, in *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, H. Li, G.-A. Levow, Z. Yu, *et al.*, Eds., Singapore and Online: Association for Computational Linguistics, 2021, pp. 121–132. DOI: `10.18653/v1/2021.sigdial-1.13`.

[97] C. Zhu, M. Zeng, and X. Huang, “Multi-task Learning for Natural Language Generation in Task-Oriented Dialogue”, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, 2019, pp. 1261–1266. DOI: `10.18653/v1/D19-1123`.

[98] Y. Zhu, R. Kiros, R. Zemel, *et al.*, *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*, 2015. arXiv: `1506.06724`.

[99] F. Zhuang, Z. Qi, K. Duan, *et al.*, *A Comprehensive Survey on Transfer Learning*, 2020. arXiv: `1911.02685`.