Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Σηματων, Ελεγχου και Ρομποτικης

Extending self-supervised natural language models through human brain

Διπλωματικη Εργασια

ΝΙΚΟΛΑΟΣ ΛΟΥΚΑΣ

Επιβλέπων: Αλέξανδρος Ποταμιάνος
Αναπληρωτής Καθηγητής

Αθήνα, Ιούλιος 2021

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Σηματων, Ελεγχου και Ρομποτικης

Extending self-supervised natural language models through human brain

Διπλωματικη Εργασια

ΝΙΚΟΛΑΟΣ ΛΟΥΚΑΣ

Επιβλέπων: Αλέξανδρος Ποταμιάνος
Αναπληρωτής Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26η Ιουλίου 2021.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
....	....	....
Αλέξανδρος Ποταμιάνος	Ανδρέας-Γεώργιος Σταφυλοπάτης	Στέφανος Κόλλιας
Αναπληρωτής Καθηγητής	Καθηγητής	Καθηγητής

Αθήνα, Ιούλιος 2021

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Σηματων, Ελεγχου και Ρομποτικης

(Υπογραφή)

. . . .
Νικόλαος Λούκας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Σε αυτή τη διπλωματική εργασία, ασχολούμαστε με θέματα στους τομείς της Γνωστικής Επιστήμης και της Επεξεργασίας Φυσικής Γλώσσας. Ερευνούμε τις αναπαραστάσεις φυσικής γλώσσας στον ανθρώπινο εγκέφαλο και τις συγκρίνουμε με παραδοσιακές αναπαραστάσεις γλώσσας της μηχανικής μάθησης.

Σε αυτήν την εργασία, χρησιμοποιούμε πρώτα ένα γνωστό σύνολο δεδομένων fMRI για να αντιστοιχίσουμε παραδοσιακές αναπαραστάσεις λέξεων σε γνωστικές αναπαραστάσεις. Παρουσιάζουμε ένα μοντέλο εγκεφαλικής ενεργοποίησης, με παλινδρόμηση κορυφογραμμών απευθείας από αναπαρασάσεις Glove, αντί για ένα ενδιάμεσο σημασιολογικό μοντέλο χαρακτηριστικών που προτείνεται στη βιβλιογραφία, το οποίο χρησιμοποιεί ένα σύνολο λέξεων με τις ανάλογες μετρήσεις fMRI για να βρει μια αντιστοίχιση μεταξύ σημασιολογίας λέξεων και τοπικών εγκεφαλικών ενεργοποιήσεων. Στη συνέχεια, συγκρίνουμε αυτό το μοντέλο κωδικοποίησης, σε διάφορες παραλλαγές, με παραδοσιακές αναπαραστάσεις λέξεων σε ένα πρόβλημα σημασιολογικής ομοιότητας και συμπεραίνουμε ότι η απόδοσή του δεν επηρεάζεται συνολικά από το είδος του χώρου των αρχικών αναπαραστάσεων.

Στη συνέχεια, διερευνούμε πώς οι γνωστικές αναπαραστάσεις θα μπορούσαν να επηρεάσουν τις αναπαραστάσεις ενός γλωσσικού μοντέλου. Ενσωματώνουμε τις γνωστικές αναπαραστάσεις στα γλωσσικά μοντέλα προσθέτοντάς τις ως ερωτήματα στο επίπεδο προσοχής, προκειμένου να αποτυπώσουμε την εγκεφαλική πληροφορία αυτών των αναπαραστάσεων στη διαδικασία εκπαίδευσης τους. Αφού διαπιστώσουμε ότι βελτιώνεται η ικανότητά τους να προβλέπουν δεδομένα εγκεφάλου, δοκιμάζουμε την απόδοση των μοντέλων σε κλασσικά πειράματα επεξεργασίας φυσικής γλώσσας. Τα αποτελέσματά μας δείχνουν ότι, ακόμη και η περίπλοκη αρχιτεκτονική του BERT επηρεάζεται αρνητικά από τις θορυβώδεις εγκεφαλικές αναπαραστάσεις. Η δομή του πειράματος μας, αν και είναι πολλά υποσχόμενη, δεν μπορεί να εκμεταλλευτεί πλήρως την αξία των γνωστικών αναπαραστάσεων.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Βαθειά Μάθηση, Νευρωνικά Δίκτυα, Εγκεφαλικές Αναπαραστάσεις, fMRI, BERT, Διανύσματα Λέξεων, Επεξεργασία Φυσικής Γλώσσας

Abstract

In this diploma thesis, we are concerned with tasks in the domains of Cognitive Science and Natural Language Processing (NLP). We investigate natural language representations in human brain and comparing them with traditional machine learning representations. In this work, we developed a pipeline for extracting neural representations from fMRI datasets by using machine learning techniques, following literature's guideline. Moreover, we evaluate our work on downstream tasks and provide useful comparative tables.

In this work, we first utilize a well known fMRI dataset to map traditional word embeddings to cognitive representations. We present a neural activation model, with ridge regression directly from glove embeddings instead of an intermediate semantic feature model proposed in the literature, that uses a set words with available fMRI measurements in order to find a mapping between word semantics and localized neural activations. Then, we compare this encoding model, in several variations, with traditional word embeddings on a similarity task and conclude that its performance is not affected overall from the semantic or glove space.

Thereafter, we investigate how cognitive embeddings could affect a language model's representations. We incorporate cognitive embeddings into language models by adding them as queries in the attention layer, in order to induce the cognitive bias of these embeddings into their training process. After finding that their ability to predict brain recordings improves, we test the models' performance at NLP tasks. Our results indicate that, even the complex BERT architecture is negatively affected by the noisy neural representations. Our experiment setup, although it's very promising, can not fully exploit the potential of cognitive embeddings.

Keywords

Machine Learning, Deep Learning, Neural Networks, Brain Representations, fMRI, BERT, Word Embeddings, Natural Language Processing

σε όλους όσους περάσαμε μια κάλη στιγμή

Ευχαριστίες

Θα ήθελα καταρχάς να ευχαριστήσω τον καθηγητή Αλέξανδρο Ποταμιάνο που μου εδώσε την ευκαιρία να εκπονήσω τη διπλωματική μου εργασία στο εργαστήριο του. Οι χρήσιμες συμβουλές και τα σχόλια του με βοηθούσαν να συνεχίζω και να μη τα παρατάω. Επίσης θα ήθελα να ευχαριστήσω τους υποψήφιους διδάκτορες Γιώργο Παρασκευόπουλο και Ευθύμη Γεωργίου για το χρόνο που μου αφιέρωσαν και την υποστήριξη που μου προσέφεραν καθ' όλη τη διάρκεια της διπλωματικής μου. Παράλληλα ευχαριστώ τους συναδέλφους και φίλους Νικήτα και Δημήτρη για όλες τις επικοδομητικές συζητήσεις μας αλλά και τη ψυχολογική τους υποστήριξη.

Στη συνέχεια, θέλω να ευχαριστήσω την οικογένεια μου Βασίλη, Ντίνα, Βάγια και Φωτεινή που με έχουν στηρίξει κυριολεκτικά στα πάντα, καθώς και τη γιαγιά μου Βαΐα η οποία αποτέλει το μεγαλύτερο υποδείγμα ανιδιοτελούς αγάπης που έχω συναντήσει εκεί έξω.

Τέλος ευχαριστώ τους παιδικούς μου φίλους οι οποίοι αποτελούν τη δεύτερη οικογένεια μου αλλά και τους φοιτητικούς μου φίλους οι οποίοι σημάδεψαν λίγα απο τα καλύτερα χρόνια της ζωής μου. Για συντομία δεν θα αναφέρω τα ονόματα τους αλλά όλοι τους γνωρίζουν το πόσο σημαντικοί είναι για μένα.

Αθήνα, Ιούλιος 2021

Νικόλαος Λούκας

Contents

List of Figures

List of Tables

Κεφάλαιο 0

Εκτεταμένη Περίληψη

Σε αυτήν την διπλωματική εργασιά, θα προσπαθήσουμε να συνδυάσουμε αναπαραστάσεις λέξεων και εγκεφάλου για να μάθουμε κοινές αναπαραστάσεις αυτών των δύο. Στο [9], πρόσφατα εισήγαγαν το πρώτο μοντέλο ειδικά σχεδιασμένο για να συλλάβει τον τρόπο με τον οποίο ο εγκέφαλος αναπαριστά τη σημασία της γλώσσας. Με τη βελτιστοποίηση του μοντέλου BERT για την πρόβλεψη δεδομένων εγκεφαλικής ενεργοποίησης ανθρώπων από το σύνολο δεδομένων του [10], κατέληξαν σε αναπαραστάσεις που κωδικοποιούν περισσότερες πληροφορίες σχετικές με τη δραστηριότητα του εγκεφάλου και έτσι βελτιώνουν την ποιότητα της πρόβλεψης της εγκεφαλικής ενεργοποίησης.

Επιπλέον, το [11] διερεύνησε πώς τα γλωσσικά μοντέλα θα μπορούσαν να μάθουν από τον ανθρώπινο εγκέφαλο. Το πείραμά τους αποτελείται από αναπαραστάσεις από 4 πρόσφατα μοντέλα: ELMO, BERT, USE και T-XL και δεδομένα εγκεφαλικών ενεργοποιήσεων. Παρόμοια με τα [6, 7] έχουν χρησιμοποιήσει γραμμική παλινδρόμηση από την αναπαράσταση ενός μοντέλου με μια πρόταση, s, ως είσοδο, για να προβλέψουν την εγκεφαλική ενεργοποίηση της ίδιας πρότασης s. Με βάση την επιτυχία αυτής της μεθόδου, μπόρεσαν να πουν εάν ένα επίπεδο του μοντέλου μοιράζεται πληροφορίες με μία περιοχή του εγκεφάλου και προχωρήσαν περαιτέρω τροποποιώντας το επίπεδο του μοντέλου και παρατηρώντας πώς αλλάζει η ικανότητα πρόβλεψης των εγγραφών fMRI. Τα ευρήματά τους υποδηλώνουν ότι η αλλαγή ενός μοντέλου επεργασίας φυσικής γλώσσας για καλύτερη αντιστοίχιση με εγγραφές εγκεφάλου μπορεί να οδηγήσει σε καλύτερη κατανόηση της γλώσσας από το μοντέλο, καθώς πέτυχαν καλύτερη απόδοση σε πειράματα επεξεργασίας φυσικής γλώσσας με την τροποποιημένη έκδοση του μοντέλου από τη βασική δομή του BERT.

Η προσέγγισή μας, με βάση τα παραπάνω, προσπαθεί να τροποποιήσει τα γλωσσικά μοντέλα ενσωματώνοντας τις γνωστικές αναπαραστάσεις στη διαδικασία εκπαίδευσης. Ελέγχοντας πώς βελτιώνεται η ικανότητά τους να προβλέπουν τη δραστηριότητα του εγκεφάλου, θα προσπαθήσουμε να επιτύχουμε καλύτερη απόδοση του μοντέλου σε πειράματα επεξεργασίας φυσικής γλώσσας. Ως βασική μέθοδο, προσπαθούμε να προσθέσουμε τα δεδομένα απεικόνισης του εγκεφάλου στο επίπεδο προσοχής του μοντέλου. Με αυτόν τον τρόπο μετριάζεται η επίδραση των κακώς εκπαιδευμένων αναπαραστάσεων. Πειραματιζόμαστε επίσης με τα μοντέλα μας προσθέτοντας τις γνωστικές αναπαραστάσεις σε ένα μόνο επίπεδο προσοχής κάθε φορά. Η προηγούμενη δουλειά από το [7] δείχνει πολύ ελπιδοφόρα αποτελέσματα όταν συνδυάζει τις αναπαραστάσεις του εγκεφάλου και των λέξεων που μάλιστα ξεπερνούν αυτές των λέξεων σε κάποια πειράματα. Από όσα γνωρίζουμε, αυτή η μέθοδος του συνδυασμού αναπαραστάσεων δεν έχει χρησιμοποιηθεί ποτέ πριν και μπορεί να είναι ο τρόπος για να προχωρήσουμε ένα βήμα παραπέρα προς μοντέλα φυσικής γλώσσας που αντιλαμβάνονται τη λειτουργία του εγκεφάλου.

0.1 Σχετική Βιβλιογραφία

Ακολουθώντας τη συνήθη προσέγγιση για μεταφορά μάθησης στην Επεξεργασία Φυσικής Γλώσσας, οι γνωστικές αναπαραστάσεις θα μπορούσαν να χρησιμοποιηθούν για να αυξήσουν την απόδοση μοντέλων φυσικής γλώσσας. Η απλούστερη προσέγγιση βασίζεται σε μία μέθοδο, όπου οι γνωστικές αναπαραστάσεις χρησιμοποιούνται ως είσοδος με ή χωρίς συνένωση με παραδοσιακές λεξικές αναπαραστάσεις. Ένα προεκπαιδευμένο μοντέλο που αντιστοιχεί λέξεις σε ένα γνωστικό χώρο θα μπορούσε επίσης να εκπαιδευτεί από άκρο σε άκρο για μια συγκεκριμένη εργασία. Στο [12], χρησιμοποιούν δεδομένα fMRI, που προέρχονται από προτάσεις, σε συνδυασμό με στοιχεία κειμένου για να αυξήσουν την απόδοση σε ένα πείραμα προσθήκης ετικετών.

Η κοινή προσέγγιση για την ερμηνεία αναπαραστάσεων γλωσσικών μοντέλων είναι με η χρήση συγκεκριμένων εργασιών επεξεργασίας φυσικής γλώσσας, χαρακτηρισμού λέξεων ή αναγνώρισης συμπεριφοράς. Μερικοί ερευνητές χρησιμοποίησαν επανεκπαιδευμένα μοντέλα φυσικής γλώσσας για να προβλέψουν τη δραστηριότητα του εγκεφάλου και να αξιολογήσουν τις αναπαραστάσεις του εγκεφάλου. Αυτή η επανεκπαίδευση είναι ένα νέο βήμα στην επεξεργασία φυσικής γλώσσας και βασίζεται στην κωδικοποίηση πληροφοριών από δεδομένα μιας διαδικασίας πρόβλεψης (π.χ. οι αναπαραστάσεις του εγκεφάλου στην περίπτωσή μας) στις παραμέτρους του μοντέλου. Ο στόχος είναι η βελτιστοποίηση αυτών των μοντέλων ώστε να επωφεληθούν από πολλές πηγές πληροφοριών σχετικά με την επεξεργασία της γλώσσας στον εγκέφαλο.

Υπάρχει ελάχιστη προηγούμενη δουλειά που αξιολογεί ή βελτιώνει μοντέλα φυσικής γλώσσας μέσω εγγραφών του εγκεφάλου. Το [13] προτείνει να αξιολογηθεί εάν μια αναπαράσταση λέξεων περιέχει σημασιολογία που σχετίζεται με την εγκεφαλική λειτουργία, μετρώντας πόσο καλά προβλέπουν δεδομένα παρακολούθησης ματιών και εγγραφές λειτουργικού μαγνητικού συντονισμού. Παρόμοια το [14] πρότεινε ένα πλαίσιο για την αξιολόγηση λεξικών αναπαραστάσεων με βάση το πόσο αντανακλούν τη σημασιολογία του εγκεφάλου. Έξι τύποι αναπαράστασης λέξεων αξιολογήθηκαν με παλινδρόμηση σε δεδομένα fMRI, EEG και παρακολούθησης ματιών. Αναφέρουν συσχέτιση μεταξύ της εγκεφαλικής αξιολόγησης και της απόδοσης των αναπαραστάσεων στην αναγνώριση οντοτήτων και τις εργασίες απάντησης ερωτήσεων.

Οι Jain και Huth [15], αντιστοίχισαν επίπεδα από ένα LSTM μοντέλο σε δεδομένα fMRI, από άτομα που ακούνε ιστορίες, για να εξετάσουν την ποσότητα σημασιολογικής πληροφορίας που κρατείται σε κάθε περιοχή του εγκεφάλου. Στο [11] χρησιμοποίησαν δεσομένα εγκεφαλικής δραστηριότητας για να δείξουν ότι κάθε διαφορετική αναπαράσταση του μοντέλου κωδικοποιεί πληροφορίες σχετικές με την επεξεργασία γλώσσας ανάλογα με το μέγεθος της πρότασης εισόδου. Στα [11, 9] παρατήρησαν ότι τροποποιώντας το προεκπαιδευμένο μοντέλο BERT για καλύτερη πρόβλεψη δεδομένων εγκεφάλου, πέτυχαν καλύτερα αποτελέσματα σε πειράματα Επεξεργασίας Φυσικής Γλώσσας. Αυτό υποδηλώνει ότι η τροποίηση ενός μοντέλου επεξεργασίας φυσικής γλώσσας για καλύτερη αντιχτοίχιση με δεδομένα εγκεφάλου απο άτομα που εκτελούν κάποια λειτουργία της γλώσσας μπορεί να οδηγήσει σε καλύτερη κατανόηση της γλώσσας από το ίδιο το μοντέλο.

0.2 Σύνολα Δεδομένων

Στο πρώτο μέρος των πειραμάτων μας συγκρίνουμε τις γνωστικές αναπαραστάσεις λέξεων χρησιμοποιώντας το σύνολο δεδομένων που παρουσιάστηκε από τον **Mitchell** [6] για τις εγκεφαλικές αναπαραστάσεις. Όπως αναφέρουμε στο Κεφάλαιο 3, αυτό το σύνολο δεδομένων περιέχει δεδομένα fMRI από 9 συμμετέχοντες και τα ερεθίσματα τους είναι σκίτσα και τίτλοι απο 60 είδη αντικειμένων από 12 κατηγορίες. Όλα τα ερεθίσματα παρουσιάστηκαν 6 φορές κατά τη διάρκεια της κάθε συνεδρίας, με τυχαία σειρά κάθε φορά. Ζητήθηκε από τους συμμετέχοντες να σκεφτούν τις ίδιες ιδιότητες για τα αντικείμενα και στις 6 παρουσιάσεις.

Για τη σύγκριση των αναπαραστάσεων χρησιμοποιούμε το σύνολο δεδομένων **MEN**. Περιέχει δύο σύνολα με ζεύγη αγγλικών λέξεων (ένα για εκπαίδευση και ένα για αξολόγηση) μαζί με βαθμούς ομοιότητας που έχουν προστεθεί από ανθρώπους, μέσω crowdsourcing χρησιμοποιώντας το Amazon Mechanical Turk μέσω του CrowdFlower. Αυτό το σύνολο δεδομένων χρησιμοποιείται συνήθως για τη δοκιμή μοντέλων σε μετρήσεις σημασιολογικής ομοιότητας και συγγένειας.

Για να εξαγάγουμε τις **γνωστικές αναπαραστάσεις** μας για τις Ενότητες 4.5 και 4.7, χρησιμοποιούμε το σύνολο δεδομένων που παρουσίασε ο Pereira στο [16], όπου ήθελαν να αξιολογήσουν αφηρημένες έννοιες και προτάσεις fMRI όπως περιγράφουμε στο Κεφάλαιο 3. Χρησιμοποιούμε τα δεδομένα από το πρώτο πείραμα τους που αποτελείται από εγγραφές fMRI από 16 συμμετέχοντες. Τα ερεθίσματα τους αποτελούνται από 180 λέξεις που επιλέχθηκαν για να καλύψουν ένα μεγάλο μέρος του σημασιολογικού χώρου. Κάθε λέξη αντιπροσωπεύει ένα σύμπλεγμα λέξεων που βασίζεται στο χώρο των διανυσμάτων Glove (300 διαστάσεις). Τα ερεθίσματα παρουσιάστηκαν σε τρία πειράματα με πολλαπλές επαναλήψεις, ως μία πρόταση, ως εικόνα ή ως σύννεφο λέξεων. Αυτά τα προεπεξεργασμένα δεδομένα για κάθε συμμετέχοντα αποτελούνται από έναν πίνακα: λέξεις (180) x voxels (~ 200.000) και χάρτες αντιστοίχισης για 3D voxel και διανυσματικούς χώρους.

Για να βελτιώσουμε τα γλωσσικά μας μοντέλα, χρησιμοποιούμε το **WikiText** σύνολο δεδομένων [17], το οποίο είναι μια συλλογή με πάνω από 100 εκατομμυρία tokens που εξάγονται από το σύνολο των επαληθευμένων Good and Featured άρθρων της Wikipedia. Το σύνολο δεδομένων αυτό, είναι διαθέσιμο με την άδεια της Creative Commons Attribution-ShareAlike. Σε σύγκριση με την προεπεξεργασμένη έκδοση του Penn Treebank (PTB), το WikiText-2 είναι πάνω από 2 φορές μεγαλύτερο και το WikiText-103 είναι πάνω από 110 φορές μεγαλύτερο. Το σύνολο δεδομένων WikiText διαθέτει επίσης ένα πολύ μεγαλύτερο λεξιλόγιο και διατηρεί την αρχική κλίση, τα σημεία στίξης και τους αριθμούς - τα οποία όλα αφαιρούνται στο PTB. Δεδομένου ότι αποτελείται από πλήρη άρθρα, το σύνολο δεδομένων αυτό είναι κατάλληλο για μοντέλα που μπορούν να επωφεληθούν από τις μακροπρόθεσμες εξαρτήσεις μεταξύ των λέξεων.

Για τη διαδικασία πρόβλεψης fMRI χρησιμοποιούμε το σύνολο δεδομένων **Harry** απο το [10] που περιλαμβάνει δεδομένα MEG και fMRI που καταγράφονται από άτομα καθώς διαβάζουν ένα κεφάλαιο από το πρώτο βιβλίο του Χάρι Πότερ. Το κεφάλαιο περιελάμβανε 5176 λέξεις και διαβάστηκε από εννέα συμμετέχοντες για κάθε πείραμα. Τα δεδομένα του πειράματος των MEG για έναν συμμετέχοντα αποκλείστηκαν λόγω πολλών θορύβων, αφήνοντας 8 συμμετέχοντες.

0.3 Πειράματα με γνωστικές αναπαραστάσεις

Στα ακόλουθα πειράματα στοχεύουμε να διερευνήσουμε τις δυνατότητες των γνωστικών αναπαραστάσεων συγκρίνοντάς τα με γνωστές αναπαραστάσεις λέξεων όπως τα Word2vec [18]. Χρησιμοποιούμε παλινδρόμηση κορυφογραμμών για να μάθουμε την αντίστοιχιση κάθε λέξης στο χώρο του γνωστικών αναπαραστάσεων, ακολουθώντας τη μέθοδο από τα [6, 7]. Για την επιλογή των voxel, χρησιμοποιούμε τον μέσο συντελεστή Pearson όλων των επαναλήψεων του πειράματος για να ταξινομήσουμε τα voxels με βάση τη σταθερότητά τους και στη συνέχεια επιλέγουμε τα 500 με τις καλύτερες βαθμολογίες σταθερότητας.

Παρουσιάζουμε το accuracy των σωστών προβλέψεων για κάθε συμμετέχοντα:

Subject	Ridge Reg.	Athanasiou	Mitchell
1	0.91	0.84	0.83
2	0.69	0.82	0.76
3	0.80	0.76	0.78
4	0.87	0.79	0.72
5	0.75	0.78	0.78
6	0.60	0.65	0.85
7	0.76	0.75	0.73
8	0.66	0.68	0.68
9	0.76	0.68	0.82
average	0.75	0.75	0.77

Table 1. *Αντιχτοιχίζουμε απευθείας από διανύσματα Glove σε 500 σταθερά voxels και συγκρίνουμε με τα αποτελέσματα των [7] και [6]*

Συνολικά, καταλήγουμε ότι το μοντέλο κωδικοποίησης, είτε μέσω σημασιολογικού είτε μέσω Glove χώρου, δεν επηρεάζει σημαντικά την απόδοση.

Αξιολογούμε το μοντέλο εγκεφαλικής κωδικοποίησης που παρουσιάζεται παραπάνω με γραμμική αντιστοίχιση, στο σύνολο δεδομένων ομοιότητας MEN. Χρησιμοποιούμε πάλι το σύνολο δεδομένων του Mitchell και αναπαραστάσεις GloVe για τις λέξεις του πειράματος όπως αναφέρθηκε προηγουμένως.

Συγκρίνουμε την μέθοδο μας με μια κλασσική μέθοδο που χρησιμοποιεί w2vec 300-dim vectors. Παρουσιάζουμε το βασικό μας μοντέλο κωδικοποίησης, όπου χρησιμοποιούμε παλινδρόμηση στα δεδομένα fMRI ενός συμμετέχοντα κάθε φορά, επιλέγοντας τα καλύτερα 500 voxels, με την προαναφερθείσα μέθοδο βαθμολογίας σταθερότητας. Για ένα από τα πειράματά μας χρησιμοποιούμε το μοντέλο κωδικοποίησης στον μέσο όρο των δεδομένων από όλους τους συμμετέχοντες. Επιπλέον, η μέθοδος Hyperalignment που εξηγείται στο Κεφάλαιο 3, χρησιμοποιείται με το Shared Response Model από μόνη της για το πείραμα της σημασιολογικής ομοιότητας και στη συνέχεια σε συνδυασμό με αναπαραστάσεις w2vec με τις τελικές αναπαραστάσεις να είναι ο μέσος όρος των δύο. Τα αποτελέσματα παρουσιάζονται στον πίνακα 2.

Subset	w2vec	500 vox	avg (200 vox)	SRM	SRM - w2vec
All Concrete	0.73	0.67 (0.69)	0.63	0.66	**0.74**
Most & Least Sim	0.60	0.57 (0.62)	0.53	0.64	**0.65**
Least Similar	**0.21**	0.14 (0.36)	0.06	0.1	0.11
Most Similar	0.09	-0.02 (0.19)	0.20	0.20	**0.21**

Table 2. *Ο συντελεστής Spearman για τα διαφορετικά υποσύνολα ουσιαστικών και για διαφορετικές αναπαραστάσεις. Παρουσιάζουμε το μέσο όρο της βαθμολογίας των συμμετεχόντων, οι τιμές σε παρένθεση (·) υποδεικνύουν το μέγιστο μεταξύ των συμμετεχόντων. Για τον μέσο όρο των εγκεφαλικών δεδομένων (στήλη avg) αρχικά πήραμε τον μέσο όρο των εγγραφών από όλους τους συμμετέχοντες και στη συνέχεια προχωρήσαμε σε παλινδρόμηση για σημασιολογική ομοιότητα. Το SRM αναφέρεται στο Shared Response Model για Hyperalignment [8]*

Συνολικά, τα οφέλη από τη χρήση γνωστικών αναπαραστάσεων ως είσοδους για υπολογιστικές εργασίες δεν είναι σημαντικά με αποτέλεσμα μόνο μια μικρή αύξηση της απόδοσης. Μια αντίστοιχη ανάλυση από το [7] δείχνει ότι τα εγκεφαλικά δεδομένα κωδικοποιούν χρήσιμες σημασιολογικές πληροφορίες, αλλά μια προσέγγιση που βασίζεται απλά σε αντιχτοίχιση μέσω γραμμικής παλινδρόμησης μπορεί να μην είναι ο καλύτερος τρόπος για να τις εκμεταλευτούμε.

0.4 Τροποποίηση μοντέλων φυσικής γλώσσας με γνωστικές αναπαραστάσεις

Αφού δοκιμάσαμε τις γνωστικές αναπαραστάσεις αυτόνομες σε κάποια πειράματα, αξιοποιούμε την υπάρχουσα βιβλιογραφία και προτείνουμε την τροποποίηση γλωσσικών μοντέλων προσθέτοντας γνωστικές αναπαραστάσεις στις υπάρχουσες αρχιτεκτονικές τους. Τα πειράματα μας εστιάζουν στη προσθήκη των γνωστικών αναπαραστάσεων στο επίπεδο προσοχής ενός μοντέλου με διάφορους τρόπους. Επιλέγουμε αυτή τη μέθοδο λόγω του μεγάλου θορύβου και των λίγων δειγμάτων που χαρακτηρίζουν τα δεδομένα fMRI, καθιστώντας αυτές τις αναπαραστάσεις κακούς υποψηφίους για την εκπαίδευση ή την επανευκπαίδευση ενός γλωσσικού μοντέλου. Κατ' αρχάς, εξάγουμε τις γνωστικές αναπαραστάσεις μας με τις μεθόδους που περιγράφονται παραπάνω. Στη συνέχεια, δοκιμάζουμε την προτεινόμενη μέθοδο σε ένα μοντέλο LSTM, το οποίο λόγω του μικρότερου μεγέθους του μας επιτρέπει να προσθέσουμε τις γνωστικές αναπαραστάσεις στη διαδικασία εκπαίδευσης και όχι μόνο κατά τη διάρκεια της επανεκπαίδευσης. Τέλος, το BERT [5] τροποποιείται με πολλούς διαφορετικούς τρόπους και στη συνέχεια επανεκπαιδεύται στο Masked Language Modeling task.

Για το LSTM μοντέλο εκπαιδεύουμε τρία μοντέλα στο WikiText-2, πάνω στο Masked Language Modeling task.

1. Ένα **base SHA-RNN** μοντέλο με μόνο ένα επίπεδο προσοχής.

2. Ένα μοντέλο **finetuned** με τις γνωστικές αναπαραστάσεις στο επίπεδο προσοχής για 5 εποχές.

3. Ένα μοντέλο **trained all the way** με τις γνωστικές αναπαραστάσεις στο επίπεδο προσοχής

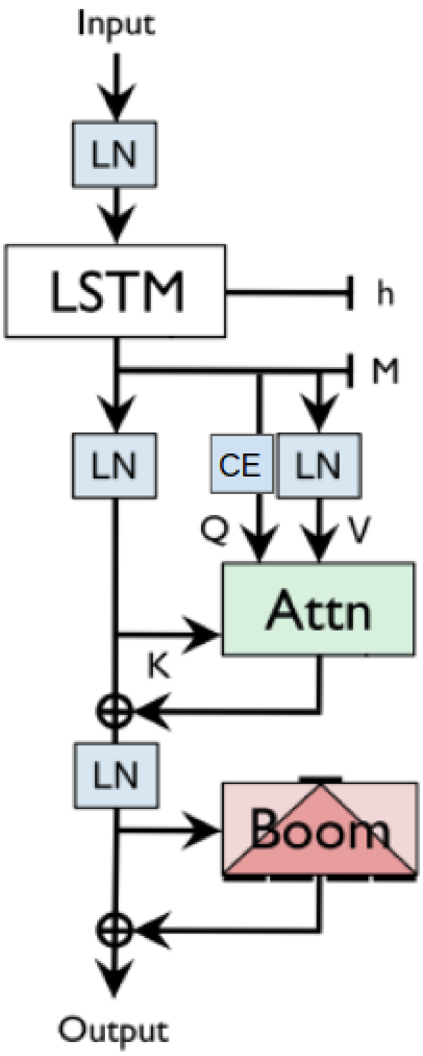


Figure 1. *Το SHA-RNN αποτελείται απο ένα RNN μοντέλο με επίπεδο προσοχής, και ένα "Boom" feed-forward επίπεδο με κανονικοποίηση. Το CE επίπεδο πρίν τα Q αναφέρεται στις γνωστικές αναπαραστάσεις.*

Συνολικά για τα αποτελέσματα με το BERT θα χρησιμοποιήσουμε:

1. Ένα απλό μοντέλο $BERT_{base}$ ως βάση για τα πειράματα μας.

2. Ένα $BERT_{base}$ όπου προσθέτουμε γνωστικές αναπαραστάσεις στο embedding layer μαζί με τα positional embeddings.

3. Ένα cognitive-BERT μοντέλο επανεκπαιδευμένο **μόνο** με γνωστικές αναπαραστάσεις στον πίνακα Q.

4. Ένα cognitive-add-BERT μοντέλο μετά την επανεκπαίδευση όπου **προσθέτουμε** τις γνωστικές αναπαραστάσεις στον πίνακα Q.

5. Ένα cognitive-BERT-LSTM μοντέλο επανεκπαιδευμένο **μόνο** με γνωστικές αναπαραστάσεις στον πίνακα Q, αφού πρώτα τις περάσουν από ένα επίπεδο LSTM.

6. Ένα cognitive-add-BERT-LSTM μοντέλο μετά την επανεκπαίδευση όπου **προσθέτουμε** τις γνωστικές αναπαραστάσεις στον πίνακα Q, αφού πρώτα τις περάσουν από ένα επίπεδο LSTM.

7. Ένα cognitive-add-1 μοντέλο, με γνωστικές αναπαραστάσεις μόνο στο επίπεδο προσοχής 1.

8. Ένα cognitive-add-2 μοντέλο, με γνωστικές αναπαραστάσεις μόνο στο επίπεδο προσοχής 2.

9. Ένα cognitive-add-6 μοντέλο, με γνωστικές αναπαραστάσεις μόνο στο επίπεδο προσοχής 6.

10. Ένα cognitive-add-11 μοντέλο, με γνωστικές αναπαραστάσεις μόνο στο επίπεδο προσοχής 11.

0.5 Πρόβλεψη fMRI

Σε αυτό το μέρος της δουλειάς μας, αξιολογήσαμε την απόδοση των μοντέλων LSTM χρησιμοποιώντας τη μέθοδο από το [11] στο σύνολο δεδομένων Harry, όπως αναφέρουμε παραπάνω. Χρησιμοποιούμε μόνο τα μοντέλα LSTM που παρουσιάσαμε προηγουμένως λόγω μικρότερων χρόνων εκπαίδευσης. Στόχος μας είναι να ελέγξουμε εάν εισάγοντας δεδομένα εγκεφάλου σε ένα γλωσσικό μοντέλο, είναι πιθανό οι αναπαραστάσεις του να αποτελούνται από περισσότερες πληροφορίες σχετικές με τον εγκέφαλο.

Subject	Base	Finetuned	Trained
1	0.54	0.56	**0.57**
2	**0.61**	0.59	0.6
3	0.64	0.64	0.64
4	**0.51**	0.5	0.5
5	0.56	0.57	**0.57**
6	**0.61**	0.6	0.6
7	0.59	0.59	0.59
8	0.6	0.62	**0.63**
average	0.58	0.58	**0.59**

Table 3. *Συγκρίνουμε τα τρία μοντέλα μας με βάση το πόσο καλά προβλέπουν την εγκεφαλική ενεργοποίηση.*

0.6 Πειράματα

Μετά την τροποποίηση του BERT και του LSTM, δοκιμάζουμε πώς αυτές οι αλλαγές επηρεάζουν την ικανότητά τους να προβλέπουν τη γλώσσα δοκιμάζοντας την απόδοσή τους σε tasks επεξεργασίας φυσικής γλώσσας. Τρέχουμε τα μοντέλα μας σε επτά downstream tasks και συγκρίνουμε τα αποτελέσματά τους στον πίνακα 4.

	LSTM Models			**BERT Models**					
Task (μετρική)	**SHA-RNN base**	**finetuned**	**trained**	**BERT base**	**Emb-layer**	**Cogn**	**Cogn-add**	**Cogn-lstm**	**Cogn-add-lstm**
CoLA (Matth.)	**0.35**	0.09	0.01	**0.578**	0.0	0.012	0.21	0.0	0.267
SST-2 (Acc.)	**0.9**	0.73	0.67	**0.917**	0.777	0.802	0.915	0.813	0.901
MRPC (F1)	**0.84**	0.713	0.562	**0.907**	0.82	0.821	0.817	0.815	0.816
STS-B (Pears.)	**0.79**	0.34	0.13	**0.913**	0.068	0.105	0.825	0.109	0.841
QNLI (Acc.)	**0.798**	0.678	0.53	**0.893**	0.611	0.865	0.87	0.633	0.878
RTE (Acc.)	**0.592**	0.532	0.511	**0.714**	0.469	0.537	0.545	0.534	0.588
WNLI (Acc.)	**0.651**	0.543	0.512	0.436	**0.563**	0.408	0.422	0.408	0.478

Table 4. *Σύγκριση των μοντέλων, με τις γνωστικές αναπαραστάσεις σε όλα τα επίπεδα προσοχής του μοντέλου, σε επτά διαφορετικά downstream tasks. Για τα LSTM μοντέλα, το finetuned αναφέρεται στο μοντέλο που επανεκπαιδεύτηκε με τις γνωστικές αναπαραστάσεις στο επίπεδο προσοχης για 5 εποχές, ενώ το trained στο μοντέλο που εκπαιδεύτηκε απο το μηδέν με ενσωματωμένες τις γνωστικές αναπαραστάσεις σε όλη τη διαδικασία της εκπαίδευσης. Στα BERT μοντέλα παρουσιάζουμε ένα μοντέλο με τις εγγραφές εγκεφάλου στο embedding layer (στήλη Emb-layer) και στη συνέχεια ένα μοντέλο μόνο με γνωστικές αναπαραστάσεις στον πίνακα Q (στήλη Cogn) και ένα όπου προσθέτουμε τις γνωστικές αναπαραστάσεις στον πίνακα Q (στήλη Cogn-add). Τέλος, οι δύο παραπάνω αρχιτεκτονικές επαναλαμβάνονται αφού πρώτα περάσουμε τις γνωστικές αναπαράστασεις απο ενα επίπεδο LSTM (στήλες Cogn-lstm και Cogn-add-lstm).*

Task (μετρική)	**BERT base**	**Cogn-add-1**	**Cogn-add-2**	**Cogn-add-6**	**Cogn-add-11**
CoLA (Matth.)	**0.578**	0.08	0.397	0.431	0.296
SST-2 (Acc.)	**0.917**	0.916	0.916	0.913	0.912
MRPC (F1)	**0.907**	0.82	0.822	0.82	0.82
STS-B (Pears.)	**0.913**	0.833	0.832	0.832	0.833
QNLI (Acc.)	**0.893**	0.874	0.877	0.877	0.876
RTE (Acc.)	**0.714**	0.548	0.555	0.563	0.556
WNLI (Acc.)	0.436	**0.464**	0.437	0.45	0.422

Table 5. *Σύγκριση του απλού BERT μοντέλου και των δικών μας μοντέλων, με τις γνωστικές αναπαραστάσεις σε ένα μόνο επίπεδο προσοχής κάθε φορά, σε επτά διαφορετικά downstream tasks. Για όλα τα μοντέλα χρησιμοποιούμε τη μέθοδο όπου οι γνωστικές αναπαραστάσεις προστίθενται μάζι με τις λεξικές αναπαραστάσεις του BERT στον πίνακα Q. Ο αριθμός στον τίτλο κάθε στήλης υποδηλώνει το επίπεδο του μοντέλου στο οποίο προσθέτουμε τις γνωστικές αναπαραστάσεις.*

0.7 Συμπεράσματα

Αρχικά, από τα πρώτα πειράματα με τις γνωστικές αναπαραστάσεις, συμπεραίνουμε ότι η απόδοση του μοντέλου κωδικοποίησης, δεν επηρεάζεται συνολικά από το σημασιολογικό ή το Glove χώρο. Επιπλέον, για το πείραμα της σημασιολογικής ομοιότητας, όλες οι εκδόσεις του μοντέλου επιτυγχάνουν παρόμοια αποτελέσματα με το συνδυασμό των αναπαραστάσεων Word2vec και του Shared Response Model να εμφανίζουν ελαφρώς καλύτερα αποτελέσματα. Γενικά, οι γνωστικές αναπαραστάσεις δεν είναι σαφώς η καλύτερη επιλογή ως εργαλεία για υπολογιστικές εργασίες, με βάση τα προαναφερθέντα αποτελέσματα. Αυτά τα πειράματα, αν και είναι πολλά υποσχόμενα, δεν μπορούν να εκμεταλλευτούν πλήρως τις δυνατότητες των γνωστικών αναπαραστάσεων.

Στη συνέχεια, αφού εξετάσαμε περιπτώσεις όπου οι γνωστικές αναπαραστάσεις έχουν χρησιμοποιηθεί για να βελτιώσουν μοντέλα επεξεργασιας φυσικής γλώσσας, εστιάσαμε στη διερεύνηση του τρόπου με τον οποίο αυτές οι αναπαραστάσεις θα μπορούσαν να επηρεάσουν ένα γλωσσικό μοντέλο και ποιες τροποποιήσεις είναι πιο κατάλληλες για να ανακαλύψουμε τις δυνατότητές τους. Το BERT [5] ήταν το επίκεντρο της έρευνάς μας. Στηρίξαμε την πλειονότητα των πειραμάτων μας στην υπόθεση ότι ένας καλός τρόπος για να ενσωματώσουμε τις γνωστικές αναπαραστάσεις στην αρχιτεκτονική ενός γλωσσικού μοντέλου, είναι με την προσθήκη τους στον πίνακα Q του επιπέδου προσοχής. Με αυτόν τον τρόπο, θα μπορούσαμε να ενσωματώσουμε εγκεφαλική πληροφορία στη διαδικασία εκπαίδευσης του μοντέλου και επίσης να μετριάσουμε την επίδραση του θορύβου που συναντάμε σε δεδομένα εγκεφάλου. Δοκιμάζουμε πρώτα την προσέγγισή μας σε ένα μικρότερο μοντέλο LSTM, όπου χρησιμοποιώντας τη μέθοδο από το [11], διαπιστώνουμε ότι η ικανότητά του να προβλέπει δεδομένα εγκεφάλου βελτιώνεται και αυτό μπορεί να οδηγήσει σε βελτίωση της απόδοσης του μοντέλου σε tasks επεξεργασίας φυσικής γλώσσας.

Παρ' όλα αυτά, τα αποτελέσματά μας σε downstream tasks δείχνουν ότι ακόμη και η περίπλοκη αρχιτεκτονική του BERT επηρεάζεται αρνητικά από τις θορυβώδεις εγκεφαλικές αναπαραστάσεις. Τα πειράματά μας σε μικρότερα μοντέλα δείχνουν ότι για τα tasks όπου το αρχικό μοντέλο επιτυγχάνει ήδη καλά αποτελέσματα, η απόδοσή του διατηρείται ακόμη και με τις γνωστικές αναπαραστάσεις στο επίπεδο προσοχής. Αυτό το αναλύουμε με μεγαλύτερη λεπτομέρεια στην Ενότητα 4.7, όπου τα προτεινόμενα γνωστικά μας μοντέλα BERT χάνουν εντελώς την αποτελεσματικότητά τους για ορισμένα πειράματα λόγω των κακών αναπαραστάσεων του εγκεφάλου. Μια τελευταία σημείωση για τη δουλειά μας είναι ότι όταν δοκιμάσαμε να προσθέσουμε τις γνωστικές αναπαραστάσεις σε διαφορετικά επίπεδα του μοντέλου BERT, αποδείξαμε ότι τα μεσαία επίπεδα κατανέμουν καλύτερα τα θορυβώδη δεδομένα του εγκεφάλου σε σύγκριση με τα αρχικά και τα τελευταία επίπεδα, ακριβώς όπως είχαν αναφέρει στο [11]. Συνολικά, η μέθοδος μας φαίνεται να στερείται των απαραίτητων συστατικών που χρειάζονται οι εγκεφαλικές αναπαραστάσεις για να παρέχουν ανταγωνιστικά αποτελέσματα στο πλαίσιο των σύγχρονων τεχνικών επεξεργασίας φυσικής γλώσσας.

Chapter 1

Introduction

In this Diploma Thesis, we study methods of incorporating fMRI data into Natural Language Processing (NLP). Using machine learning methods and freshly proposed Transfer Learning techniques, we evaluate the performance of brain data on downstream natural language processing tasks. Our aim is to find ways, based on the common practices, to utilize the potential of neural representations.

1.1 Motivation

Conceptual Knowledge refers to the knowledge of understanding of concepts, principles, theories, models, classifications, etc. We learn conceptual knowledge through reading, viewing, listening, experiencing, or thoughtful, reflective mental activity. The question of how the human brain represents conceptual knowledge has been debated in many scientific fields. Brain imaging studies have shown that different spatial patterns of neural activation are associated with thinking about different semantic categories of pictures and words (for example, tools, buildings, and animals).

Several recent studies, on occasion of the successes of self-supervised NLP models, are trying to investigate these models' representations in order to study how people process and understand language. Their approaches had opened ways to understand the processing of longer word sequences, context and even suggest that having NLP models specifically designed to capture the way brain represents language meaning may lead to even more insight about natural language processing. However, there is no prior work that utilizes the actual brain representations and use them to improve the overall performance of the aforementioned models.

In this work, we propose methods to extend self-supervised natural language models by combining them with cognitive embeddings and find aspects where the latter prevail. Futhermore, we experimented on evaluating encoding models for mapping word embeddings to neural representations and tested how the language models' ability to predict brain activity changes after we incorporate cognitive embeddings in their training process. In each case, there is imperative need to discover methods for fMRI data to exploit their full potential. The specific ways that cognitive embeddings can be used to achieve model architectures that outperform the current popular models yet remain unclear, even though [7] shows that cognitive embeddings encode useful semantic information.

1.2 Thesis Contributions

Below, we summarize the main contributions of our work:

- We designed a novel functional pipeline for extracting cognitive embeddings from fMRI datasets.
- We compared the performance of several cognitive representations with traditional word emebeddings on a downstream similarity task.
- We fine-tuned language models while incorporating cognitive embeddings in their training process.
- We tested if the aforementioned fine-tuned models' representations become better at predicting brain activity.
- We evaluated the performance of our "cognitive" language models on several downstream natural processing tasks. We found that models perform better only in tasks where the base model already achieves great performance.
- We tested adding cognitive embeddings at a different attention layer of the BERT model each time. We found that the mid layers are better at distributing new unknown, to the model, information.

1.3 Chapter Outline

In the first part, we describe the necessary theoretical background in order for the reader to understand the structure and the contribution of this thesis. In particular, in Chapter 3, we present information regarding the use of cognitive data in NLP and computational models. We review related work both from a theoretical and a practical perspective, while describing commonly used datasets as well as frequently applied methods for voxel selection. In addition, we present neural alignment methods and different ways of mapping between voxel space and lexical embeddings. In Chapter 2, we present an overview of machine learning theory, focusing on the models that we used in this project.

In the second part, we describe in detail the experiments that we conducted. More specifically, in Chapter 4, we describe the methodology used and the preprocessing conducted in order to develop a pipeline for extracting cognitive embeddings. We analyze the results of our work and we present comparative tables in order to illustrate the differences between each of our proposed methods.

Finally, in Chapter 5, we further discuss our findings and present a summary of this thesis, as well as future directions for inducing neural representations into natural language processing methods.

Chapter 2

Machine Learning

2.1 Introduction

Machine Learning (ML) is a field of Artificial Intelligence (AI) that creates systems with the ability to automatically learn and improve without being explicitly programmed to calculate or solve problems. The ML algorithms allow computers to be trained on input data and use statistical analysis to extract values that fall within a specific range. The learning process begins with observations, which are examples, or empirical results or instructions, so that patterns can be identified in the data and better decisions can be obtained in the future, based on the examples we have. The primary purpose is to enable computers to learn automatically, without human intervention or help, and adjust their actions accordingly.

Machine Learning is divided into three broad categories depending on the way the learning process takes place. The first category is called *Supervised Learning*. During its training phase the corresponding labeled outputs (labels) are listed together with the available data input. This labeling is usually done by humans and therefore the process is quite time consuming. The second category, that does not carry marked data but attempts to discover structures in existing observations, is called *Unsupervised Learning*. The third and last category is called *Reinforcement Learning* and treats the training system as an agent that aims to maximize a profit, defined by a Reward Function, by interacting with a dynamic environment. Consequently, the first category seeks to create a model that illustrates input data to output data, the latter seeks to identify an undercurrent structure in the input data and the third aims at optimal decision making.

In addition, two interesting categories that are often encountered in the relevant literature is that of *Semi-Supervised Learning* and *Meta Learning*. The first is a mix of Supervised and Unsupervised Learning, where some data are highlighted while most are not and the second focuses on how a system can learn how to learn.

2.2 Types of Machine Learning

Machine Learning algorithms are split in three main categories: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

2.2.1 Supervised Learning

In Supervised Learning, there are input variables X and output variables y. The goal is to learn a mapping function from input to output through an algorithm:

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

A Supervised Learning model aims to approach the display function so well that when new input data X are entered into the model, the corresponding output variables y can be predicted with success. The different types of problems observed in Supervised Learning arise from the diversity of the output y. In Classification problems, y is discrete, while in Regression problems the output y is continuous. In Classification problems, given an input, the model should classify it into a category, while in Regression problems, the model should return a continuous value as an output [19], [20].

2.2.2 Unsupervised Learning

In Unsupervised Learning, the training data are vectors X that do not contain labels for each input data. Therefore, the goal of Unsupervised Learning is to find patterns when there are no "correct answers", or when they are impossible to be calculated. Unsupervised Learning mainly solves Clustering problems, where the goal is to separate input data in different clusters, based on a given metric [20]. In addition, another category of Unsupervised Learning are Generative Models. These models mimic the process of creating training data. A good Generative Model should be able to create new data that look like the original.

2.2.3 Reinforcement Learning

Reinforcement Learning differs from the previous two categories because it focuses on optimal decision making. The core parts of Reinforcement Learning are an environment, and an agent which interacts with it over time. The agent performs actions based on observations, and then receives a reward from the environment. This process continues in a loop. The behavior of the agent depends on a function that maps the observations of the environment to actions.

2.3 Machine Learning Algorithms

2.3.1 Cost Function

The goal of any Supervised Learning algorithm is to return a function $f()$ which accurately matches the input examples to the corresponding labels. To quantify the loss (error) of the model, a *Cost Function* is used, that predicts $\hat{y}$ when the actual label is y. Usually, the *Cost Function* $L(\hat{y}, y)$ assigns a numeric value to the predicted output $\hat{y}$ given the actual output y. It must have an infimum, which means that the lower the error value the better the prediction. Function parameters are set in order to minimize L loss in the training examples.

Given a train set $(x_{1:n}, y_{1:n})$, a cost function L per sample and a function $f(x; \Theta)$, we define the total loss as the average loss on all training data:

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

The goal is to find the optimal parameters Θ that minimize the total error:

$$\hat{\Theta} = \arg_{\Theta} \min \mathcal{L}(\Theta) = \arg_{\Theta} \min \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(f(x;\Theta), y_i) \tag{2.3}$$

Some common cost functions are the following:

- **Mean Squared Error (MSE)**, which calculates the mean squared prediction error:

$$J(\partial) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - P_i)^2, \tag{2.4}$$

  where the prediction error is the difference between the true value (Y_i) and the predicted value (P_i) for an instance and ∂ is the parameter vector of the network. MSE is used with regression models [21].

- **Mean Absolute Error (MAE)**, which calculates the mean of the absolute prediction error:

$$J(\partial) = \frac{1}{n}\sum_{i=1}^{n}|Y_i - P_i|, \tag{2.5}$$

  where Y_i is the true value and P_i the predicted value for an instance and ∂ is the parameter vector of the network [22].

- **Cross Entropy Loss Function**, which uses the concept of *cross-entropy*. Cross-entropy is mathematically defined as:

$$H(p, q) = -\sum_k p_k \log q_k, \tag{2.6}$$

where p and q are the true and the predicted probability distributions respectively. The more the two distributions differ, the higher the value of the cross-entropy. Cross-entropy loss function is widely used in classification problems. Based on the definition of cross-entropy, the goal of the cross-entropy loss function is to minimize the cross-entropy between the model's distribution and the distribution of the given data [23], [20].

2.3.2 Logistic Regression

Logistic Regression is used to solve linear classification problems. It differs from other simple classifiers on how the probability of an input sample $x \in R^d$ belonging to a class, is calculated. In a binary classification problem with classes $y = \{0, 1\}$, we apply the sigmoid function, on the output vector of a function $f()$, which compresses the values of the vector in range (0,1).

$$P(y = 1|x) = \frac{1}{1 + \epsilon^{-f(x)}} \tag{2.7}$$

where $f(x)$ could be a linear function. If this probability is greater than 0.5, then sample x is categorized into the first class. Otherwise, it's categorized into the second class with probability $P(y = 0|x) = 1 - P(y = 1|x)$. The Cost Function we aim to minimize, is the *cross-entropy*, defined by the relationship:

$$J(w) = -[y \log(P(y = 1|x)) + (1 - y) \log(1 - P(y = 1|x))] \tag{2.8}$$

2.3.3 Word Embeddings

The idea behind *Word Embeddings* is that we would like vectors of similar words to have values close to each other. While word similarity is difficult to determine and depends on the particular problem, modern approaches draw inspiration from distributional hypothesis [24], arguing that words have a similar meaning when they appear in similar context. Word2vec [18] attempts to generate distributional numerical representations of words, which encode the similarity of the words. Different methods create supervised training examples, in order to predict the word based on the context, or to predict the context based on the word. The most important set of pre-trained word vectors is word2vec. Word2vec is a language model approach, applied to a finite number of words.

Word2vec

Word2vec consists of four structural elements. The Continuous Bag of Words (CBOW) and skip-gram are the two suggested algorithms, while Negative Sampling [18] and Hierarchical Softmax are the two suggested training methods. As shown in Figure **??**, the CBOW algorithm given the context of a word, tries to predict that word. In the Skipgram model, on the other hand, given a word, it attempts to predict the distribution of the words that compose the context of that word. Furthermore, Negative Sampling is based on sampling "negative" examples while Hierarchical Softmax proposes an efficient tree structure for calculating the probabilities of each word in the dictionary.

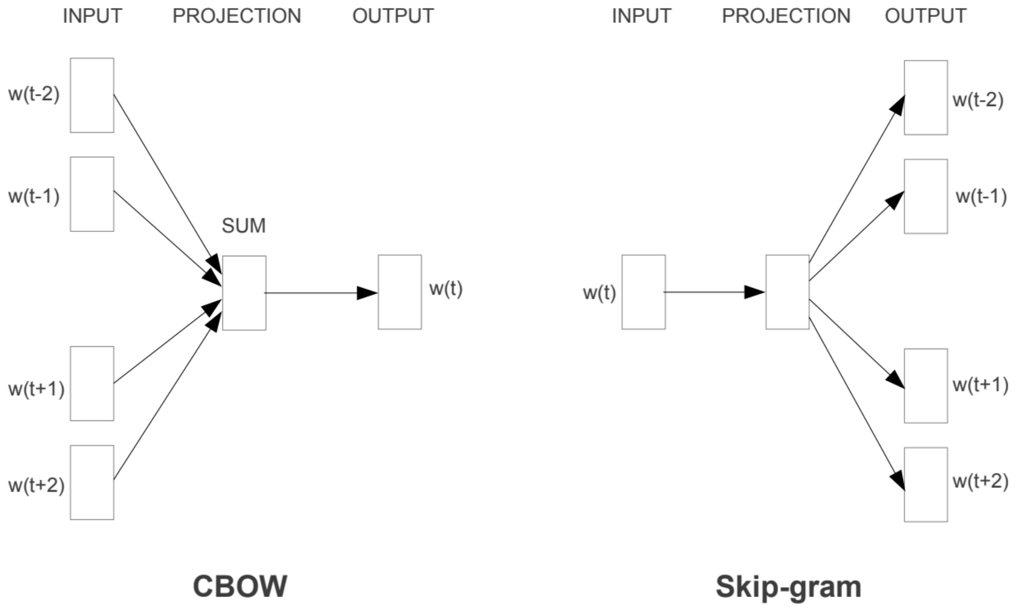


Figure 2.1. *The two suggested Word2Vec algorithms: CBOW (Left) and Skipgram (Right). Source: [1]*

Glove

In contrast to previous methods, Glove [25] is based on a model that predicts the probability of a word j, that appears in the context of a word i. Learning is achieved with least squares as the cost function. In addition, with this method, all text statistics are used and the creation of a vector space that includes important information of the meanings of the words, is achieved.

2.4 Deep Learning

Deep Learning belongs to the general field of Machine Learning (ML). Methods, such as, Deep Neural Networks, Recurrent Neural Networks and Convolutional Neural Networks have been successfully used in recent years to solve computer vision, voice recognition, natural language processing, bioscience and forecasting problems. The term "deep" derives from the existence of multiple layers in these networks. Deep Learning, is the modern version of Machine Learning. Efficient network training requires a great load of data, usually thousands of samples. Also, even though it's not necessary, the parallel processing of data on a Graphics Card (GPU) with appropriate libraries has greatly accelerated the duration of training, compared to the execution on a Central Processing Unit (CPU). The exponential growth of data due to the internet, as well as the rapid, driven by the video game industry, growth in graphics cards, are the reasons why, deep learning is the state of the art in terms of developing Artificial Intelligence Models.

2.4.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a powerful and robust type of neural networks, which are particularly useful because of their internal memory. The connections between the units in an RNN create a directed graph on a sequence. This allows the network to exhibit dynamic time behavior for a time sequence. RNNs use their internal state (memory) to process sequential inputs to the network. Intuitively, RNNs have the ability to remember important input information they've received, which allows them to make accurate predictions for the following data. As shown in Figure 2.7, the basic RNNs are nodes organized in a sequential order. The RNN first takes x_0 from the input sequence and extracts h_0 (hidden state). The hidden state h_0 together with x_1 are the input for the next step. Respectively, h_1 together with x_2 are the input for the next step and so on. Therefore, an RNN model remembers the context of the entry from the training process.

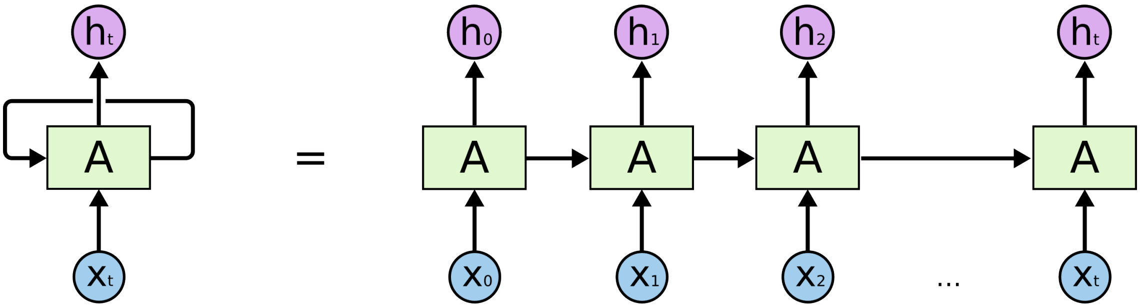


Figure 2.2. *A basic Recurrent Neural Network. Source: [2]*

Consequently, for each time point t, the equations that describe the function of an RNN are:

$$h_t = f_h(W_{hh}h_{t-1} + W_{hx}x_t + b_h) \tag{2.9}$$

$$y_t = f_y(W_{yh}h_t + b_y) \tag{2.10}$$

where h_t denotes the hidden state at time t, x_t the input vector at time t, y_t the output vector at time t, b_h the bias for h, b_y the bias for y and f_x, f_h are the activation functions for x and h respectively. There are three different weight tables: W_{hx} (weights from the entry to the hidden layer), W_{hh} (weights from the hidden layer to the hidden layer), and W_{yh} (weights from the hidden layer to the output layer).

Bi-directional RNN

As we mentioned above, RNNs capture sequential data information which they've received at time t and encode them in their hidden state. However, they are also likely to obtain more information by reading a given sequence backwards, in order to make more accurate predictions.

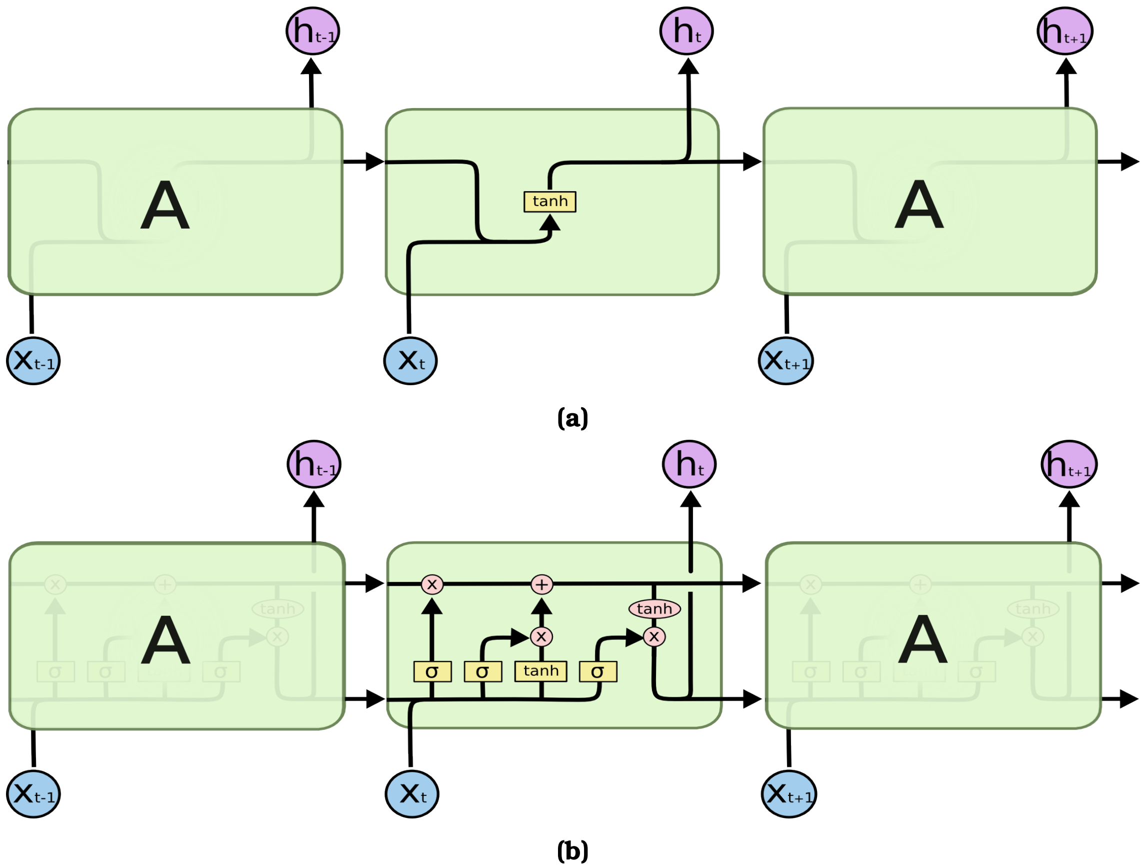
So, in a bi-directional RNN, we encode the input sequence from beginning to end (forward RNN), but also the sequence from the end to the beginning (backward RNN). Then we combine the hidden states of the two RNNs to find the hidden state for each time point. Specifically, we calculate separately the hidden state of the forward RNN $\overrightarrow{h_t}$ at time t, but also the corresponding hidden state of the backward RNN $\overleftarrow{h_t}$, and combine them to calculate the final hidden state at each time point. As a result, the hidden state at time t is simply the combination of the two vectors: $h_t = \overrightarrow{h_t} \| \overleftarrow{h}_{T-t}$. The same also applies for all $T+1$ time points of the input sequence.

2.4.2 Long Short Term Memory (LSTM) Neural Networks

A subcategory of Recurrent Neural Networks (RNNs) described above, are the LSTMs. They were originally proposed by Hochreiter and Schmidhuber in 1997 [26] and they've been studied and developed by researchers since then. The results from using these networks in time series data are very promising, offering solutions to a variety of modern problems.

LSTMs are designed to address one of the key issues of RNNs; learning long-term dependencies. For example, an RNN designed to accept 3 words from a sentence in order to predict the next, does not have the ability to "remember" important previous information from the whole sentence. This problem had been identified by [27].

The basic idea behind an LSTM, is based on the existence of interconnected portals (gates) that control (encourage or not) the flow of information from one point to another. The structure of this portal contains a sigmoid function through which an input vector passes, which is then multiplied by a second vector to produce the final output.



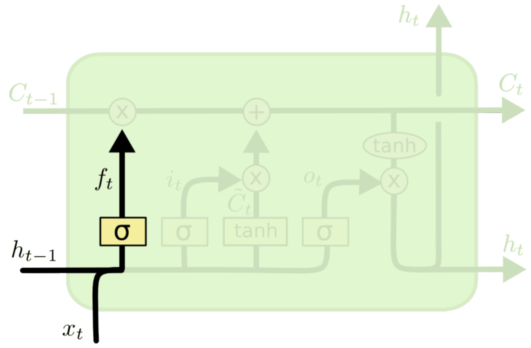
(a)

(b)

Figure 2.3. *The cell structure of a simple Recurrent Neural Network (a) compared to the structure of an LSTM (b)*

The analysis of an LSTM "cell" is presented below step by step:

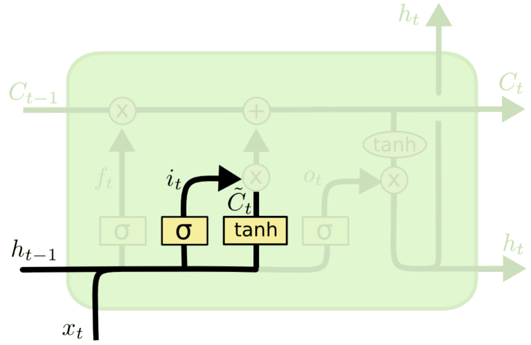
- **Forget gate:** At this point, we select the junk information contained in the previous hidden state h_{t-1} and the new input x_t, and we "forget" it through the portal. A number is produced between [0, 1] which is multiplied by the internal values of the state C_{t-1}, choosing what information will remain.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 2.4. *Forget gate of an LSTM.*

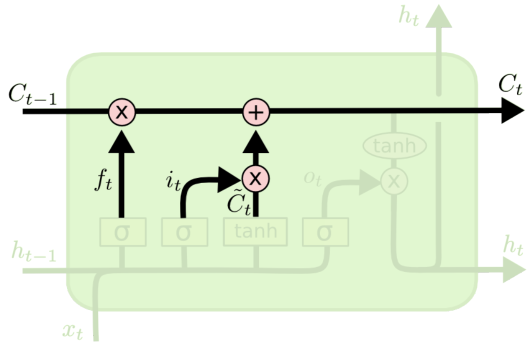
- **Store Update gate:** In the next step, we decide what new information we will store in the internal state C. This is done, first by selecting the values to be updated (sigmoid) as well as through the *tanh* portal which produces a candidate vector $\tilde{C}_t$ for these values.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 2.5. *Store gate of an LSTM.*

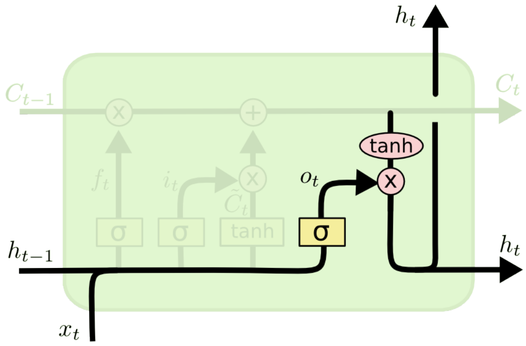
Then we repeat the first step, forgetting the useless information, and adding the new information $\tilde{C}_t$ multiplied by the percentage i_t of their change.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 2.6. *Update gate of an LSTM.*

- **Output gate:** Finally, we decide, what result will be produced at the output. To achieve this, the state calculated from the previous steps C_t is filtered, passed through a *tanh* function and multiplied by the output of a sigmoid gate, so that we output to the next stage only the parts we decided to.



$$o_t = \sigma \left(W_o \; [h_{t-1}, x_t] \; + \; b_o \right)$$
$$h_t = o_t * \tanh \left(C_t \right)$$

Figure 2.7. *Output gate of an LSTM.*

2.4.3 Attention Mechanisms

The basic idea behind the attention mechanism is that not all vectors of a sequence, contribute the same to the context. So, the model should not use all the vectors equally to make a prediction, but focus on the parts of the input that contain the most relevant information about a particular problem. To implement this approach, we use an attention mechanism [28, 29] to find the relative significance of each input vector of a sequence. In order to focus on the vectors that contain the most important information, a weight a_i is set in the hidden step, corresponding to each h_i vector. Then, the finite representation r of the entire input sequence is computed, as the weighted sum of all hidden states.

$$\begin{aligned} e_i &= \tanh\left(W_h h_i + b_h\right), \quad e_i \in [-1, 1] \\ a_i &= \frac{\exp\left(e_i\right)}{\sum_{t=1}^{T} \exp\left(e_t\right)}, \quad \sum_{i=1}^{T} a_i = 1 \\ r &= \sum_{i=1}^{T} a_i h_i \end{aligned} \tag{2.11}$$

where W_h and b_h are the weights of the attention layer.

2.4.4 Transformers

In this section, we make a general introduction to Transformers [4] since they form the basis of our models (BERT [5]). Until recently, popular models were based on recursion or convolutions and they used to connect the Encoder with the Decoder through an Attention Mechanism. The Transformers rely entirely on this Mechanism and lead to more efficient implementations by allowing the parallelization of calculations by dramatically reducing training time. Transformers are focused on solving machine translation problems and their architecture explains this fact since the Encoder, encodes information from one language into an intermediate representation, which is then passed to the Decoder and finally ends up in another language. This architecture, of course, with some modifications can be applied to a wide range of problems. As shown in Figure 2.8, a transformer consist of a set of 6 encoders connected to another set of 6 decoders. All the encoders as well as the set of the decoders consists of layers with similar structure but of course have different weight values. The encoders and decoders accept inputs from the lower layers and carry them to the higher. Each encoder consists of two subsystems. The first is a Self-Attention Layer, that allows the encoder to match the dependencies of each word, with the other words in the sentence. The second subsystem is a Feed Forward Neural Network. The decoder architecture is similar, with the only difference, between the two subsystems described above, being an extra intermediate layer called the Encoder-Decoder Attention. In a similar way, with the Seq2Seq models [30], this subsystem is responsible to locate and focus its attention on specific elements of the input that have already been encoded by the encoder

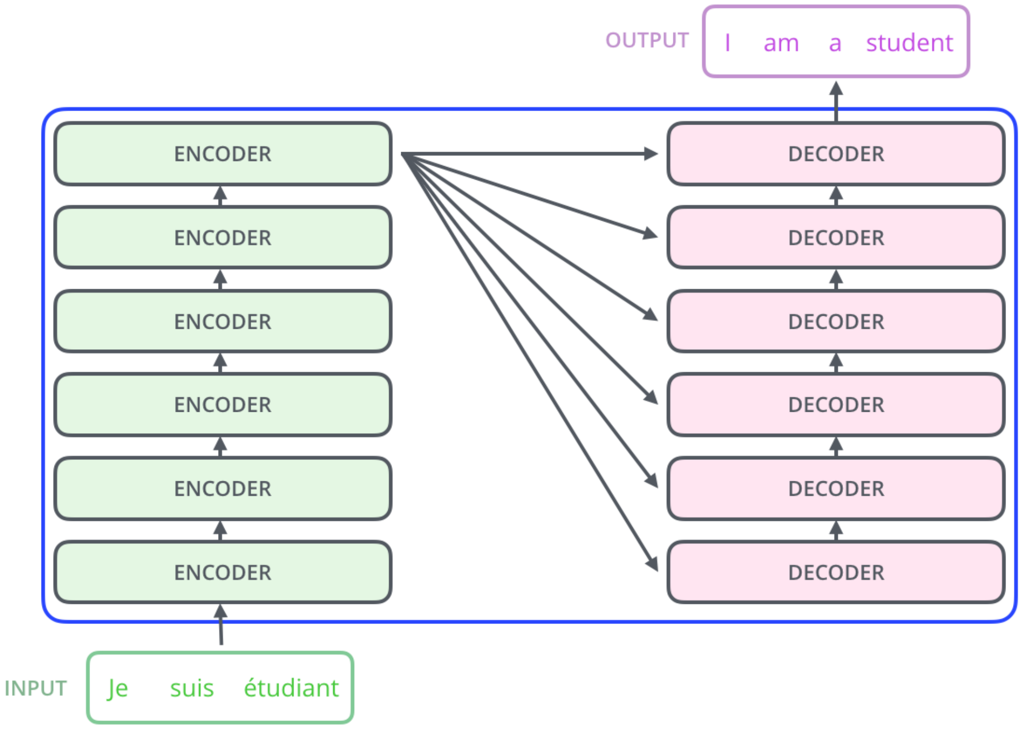


Figure 2.8. *The general structure of a Transformer. Source: [3]*

The next part of the transformer implementation is when the input enters the first encoder of the set of 6 encoders described before. As discussed in previous sections, the system needs to receive the words in the form of word vectors. Every word is transformed into an appropriate vector representation and crosses a specific path in the network. The Feed Forward Neural Network does not hold correlations between these paths and therefore the words. This suggests that the paths in this system can be paralleled. The system responsible for these correlations between the words is the Self-Attention layer.

The outputs of the Self-Attention layer are calculated in 2 basic steps. The first step is to create three (3) vectors from the word vectors of the sentence. Specifically, the Query, Key and Value matrices are constructed. These vectors, of size 64, much smaller than the dimensions (512) of hidden vectors, practically are the result of the multiplication of the input vectors with specific arrays, the parameters of which are optimized during the training process. Of course, the size mentioned before is an architectural choice but it is generally proven that at this size it is possible to have a stable representation, able to gather all the information between the dependencies of the words of the sentence.

Multi-Head Attention is also a common practice. Specifically, the Transformers introduced in [4] have 8 attention heads which essentially means that 8 different sets of W^Q, W^K and W^V arrays are generated for each Self-Attention layer of each encoder and each decoder. Therefore, the optimization parameters increase significantly but this technique aims to improve performance as the model can focus better on different parts of the sentence and each head can also give a different form of attention for the model to focus on. Consequently, multiple representation subspaces can occur. The Feed Forward Neural Network expects a different input size from the result of the operations mentioned before. So, once we concatenate all the results, we multiply this table with another W^O table, the parameters of which are optimized through training phase. Finally, the result of this operation is passed to the Feed Forward Neural Network.

From the above, one can conclude, that Transformers do not take into account the actual sequence of the words in a sentence. Therefore, it is necessary to include vectors that represent the "order" of the words in the sentence. These vectors, called positional embeddings, are learned by the system during the training process and they essentially indicate the "order" of the words. These embeddings are added with the corresponding word vectors in order to provide meaningful distances between the embedding vectors, once they're projected into Q/K/V vectors and during dot-product attention.

Self-Attention

Self-Attention utilizes three matrices, K, V and Q to calculate the attention and is described, as mentioned before, by the equation:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \tag{2.12}$$

where d_K denotes the dimensionality of the keys and queries.

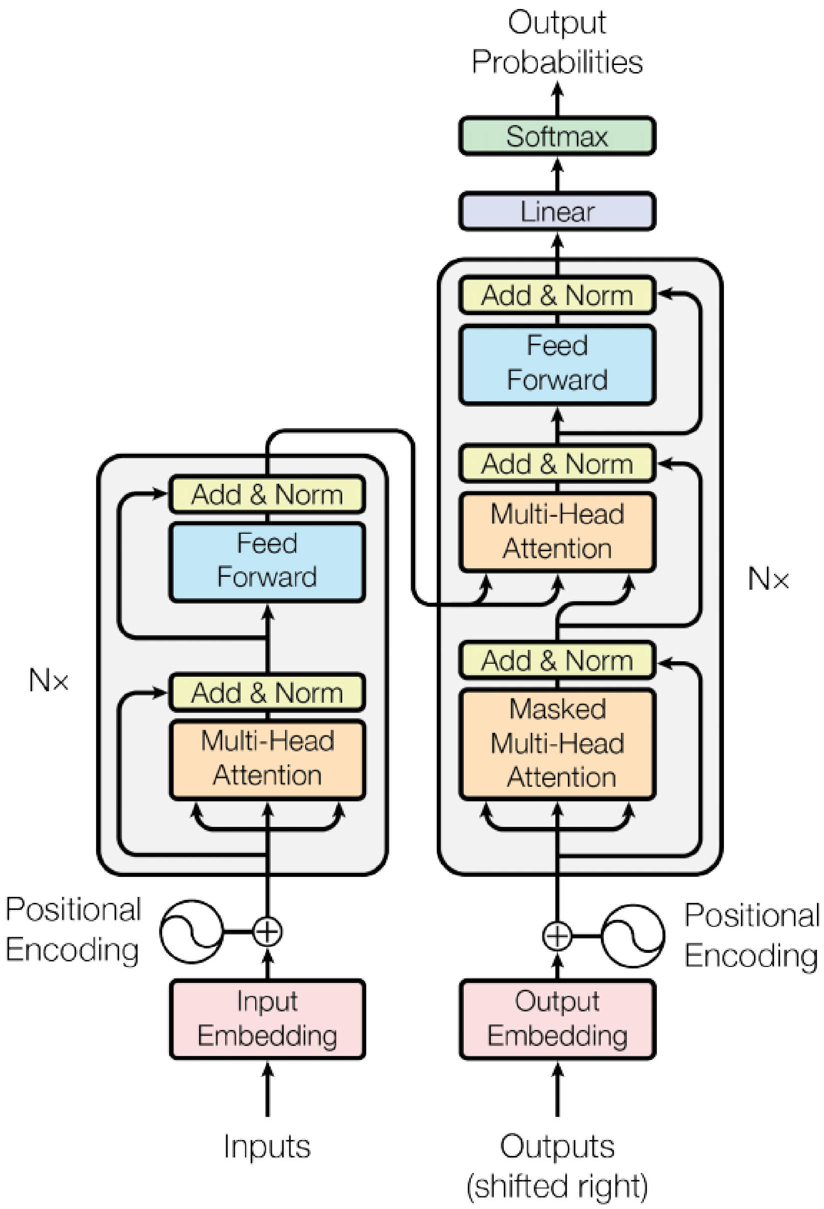


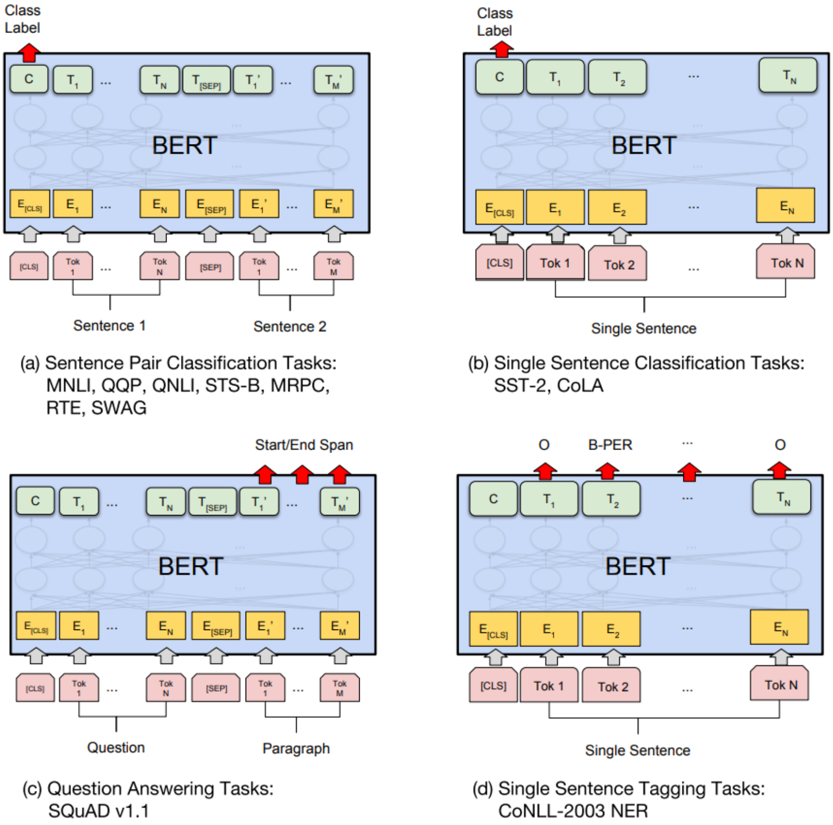
Figure 2.9. *The inner structure of a Transformer. Source: [4]*

BERT

In 2018, the Natural Language Processing (NLP) community made several important successes, with the releases of Allen AI ELMo models [31], OpenAI OpenGPT [32] and Google BERT [5]. Since then, Transfer Learning Techniques have been the center of attention and they began to see use in a wide range of applications because with minimal time, effort, data and computing power, researchers have been able to achieve significantly better results in a variety of problems. The only difference, is that a pre-trained model is needed, to be adapted and optimized (finetuning), in many cases together with a small subnet, to solve a specific task.

BERT (Bidirectional Encoder Representations from Transformers) [5] is based on several dominant ideas, such as those of Semi-Supervised Sequence Learning [33], ELMo [31], ULMFiT [34], OpenAI Transformer [32] and Transformers [4]. Specifically, BERT's training was implemented with the idea of semi-supervised sequential learning on a huge number of texts from books and Wikipedia, among other sources. The model during the training process focuses on a specific problem of language modeling called masked word prediction. This way, it learns to detect language patterns and obtains the ability to process linguistic texts as well.

In addition to extracting high quality language features from a text, BERT, along with small neural network add-ons, can solve a variety of classification problems, entity recognition and question answering, as illustrated in Figure 4.9 below. Furthermore, BERT, like ELMo, can also be used to produce contextualized word embeddings. At this point, it's worth noting that there are two versions of BERT. The first is the base model (12 Encoders - 768 hidden size) which is comparable in size to that of the OpenAI Transformer and the second is a much larger model (16 Encoders - 1024 hidden size) which obviously leads to significantly better results as previously described.



(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 2.10. *Examples of BERT's application in multiple tasks. Source: [5]*

For our BERT implementation, we used the publicly available library from HuggingFace in PyTorch [35]. In particular, for all downstream tasks, we use the BertForSequenceClassification model [36], which consists of the classic BERT model with an extra linear layer for classification.

The first step to pass a text through BERT, is the transformation of this text into a form that the model identifies. Except from the tokenization of the sentence (WordPieces [37]), special tokens should be added and each element should be given the appropriate IDs, which are also the indicators of the dictionary. For example, in text classification, the sentence is required to start with the token [CLS] indicating the classification and ending with the token [SEP]. All sentences should be padded or truncated to a specific preset length, which affects both the computational and time performance of the system. The maximum sequence length that the system can accept, as input, is 512 tokens. In addition, the system expects an attention mask that indicates which input elements correspond to essential information and which should be ignored by the system due to padding. Each of the 512 outputs of the base system is represented by a vector of 768 values. The first vector, corresponds to the first token, in this case [CLS], so is commonly used for classification.It is noteworthy, that the researchers who published BERT, also give some suggestions on the parameters to be optimized.

2.5 Transfer Learning

Most NLP models today rely on pre-trained word representations, such as word2vec [18] and GloVe [25], to initialize their embedding layer. While such pre-trained word vectors are capable of modeling the semantic similarities of words, they have limitations that do not allow them to model polysemy, or metaphorical use of language etc. Therefore, they are not able to model all the subtle aspects and concepts of natural language. To address this problem, pre-trained representations of language models have been proposed, which give a good representation of the context [38, 31], and assign a different word vector each time (even for the same word), depending on its "environment".

A major advantage of Transfer Learning, is related to the faster development and implementation of applications that use a pre-trained model. These models are already able to detect language features due to their training and as a result the engineers or programmers only have to combine them with other smaller networks and fine-tune them on each specific problem. Nonetheless, because of their computationally and financially costly training process with a huge volume of text, the final pre-trained model requires significantly less data, time and computing power for fine-tuning. The important advantage that makes these techniques popular are the state of the art results obtained on a variety of problems.

In conclusion, what has been proposed, is a model which understands basic features of the language, regardless of the problem, in order to save several hours of training on each specific task. This direction of Natural Language Processing is identical with what happened in recent years in the field of Computer Vision where large pre-trained models are used that have already learned to separate the basic "components" of an image, such as lines or angles [39].

Chapter 3

Cognitive Background

In this chapter, we present a general overview of the most common methods for fMRI preprocessing. First, we present some relevant Natural Language fMRI datasets. Second, We make a detailed analysis on voxel selection techniques and methods to combine different voxel spaces. Finally, we focus on mapping methods between language and neural representations. All these methods are the basis for our work presented in the following chapters.

3.1 Introduction

The seminal work of Mitchell [6] demonstrated that fMRI signals encode meaningful semantic information for concrete nouns, which can be effectively used to map between distributed semantic representations (DSM) and voxel activations. This was the first computational model to predict brain patterns associated with unknown words (lexical expansion). Many others have attempted since to extend this initial work, and the use of cognitive data in NLP and computational models remains an open field of research.

In the following subsections we review related work both from a theoretical and a practical perspective.Specifically, we describe commonly used datasets as well as frequently applied methods for voxel selection, neural alignment and mapping between voxel space and lexical embeddings. In the next chapter, we also compare the performance of cognitive embeddings and lexical embeddings in downstream tasks and we review cases where cognitive embeddings have been used to enhance or improve task-based models.

3.2 Datasets & Stimuli

Generally the most common neuroimaging modality used in semantic mapping is functional MRI (fMRI), which records the blood-oxygen response in the whole brain. In comparison to other methods (EGG,MEG) fMRI offer good spatial resolution (≈ 1-3 mm), with a limited temporal resolution (≈ 1-2 sec). The information signal in MRI is inherently noisy and preprocessing is a crucial step for extraction of semantic information. The stimuli can be single words, displayed one at a time as text in an isolated way, or in context as a word cloud or a series of sentences with a common theme. Whole sentences in the form of narratives have been commonly used (e.g. reading of a book passage) both as visual (text) and auditory input. As words tend to evoke visual neural responses, Images have also been extensively used as an experimental stimulus.

The dataset introduced by **Mitchell** [6] contains fMRI scans from 9 participants, and stimuli are line drawings and noun labels of 60 concrete objects from 12 semantic categories. All stimuli were presented 6 times during the scanning session, in a different random order each time. Participants were asked to think of the same item properties across the 6 presentations.

Pereira [16] wanted to evaluate abstract concepts and sentence fMRI. Tha data consists of fMRI scans for three experiments , with 16, 8, 6 participants respectively. In experiment 1, stimuli consist of 180 concept words selected to cover the semantic space. Each word represents a cluster of words based on Glove vector space (300 dim). The stimuli were shown in three paradigms with multiple repetitions, in a sentence, as an image, or in a word cloud. In experiments 2 and 3, the stimuli consisted of a collection of sentences for different topics, unrelated to concepts in experiment 1. One fMRI image was captured for each sentence.

The dataset presented in [40] (**MOUS**) is a massive 204 participant study with both visual and auditory stimuli. The participants were native speakers of **Dutch**. The total stimulus set consisted of 360 sentences in Dutch. The visual subjects read words one at a time in a sentence, in the correct and in a scrambled order. 60 sentences were shown to each subject in blocks of five sentences alternating between blocks with sentences and blocks with word lists.

BOLD5000 [41] is a functional MRI dataset that is based on responses from almost 5000 diverse real world images that overlap with typical computer vision datasets (SUN, COCO, ImageNet). Data was collected from four participants and images are comprised of 1000 indoor and outdoor scenes of 250 categories (SUN), 2000 objects embedded in realistic context (COCO) and 1916 objects of mostly singular objects (ImageNet).

The dataset by [10] consists of magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI) data recorded from people as they read a chapter from **Harry Potter**. The chapter included 5176 words and was recorded from nine participants for each experiment. For the MEG experiment data for one participant had too many artifacts and was excluded, leaving 8 participants.

The dataset in [42] includes fMRI and EEG acquisitions while participants listened to the first chapter of **Alice**'s Adventure in Wonderland, which comprises of 2,129 words in 84 sentences and has a reasonable syntactic diversity. For the fMRI data, there are anatomical and functional scans for 26 subjects and for the EGG data there are scans for 49 subjects. The dataset is annotated with predictors that range from prosody to morphology to syntax.

The dataset collection [43, 44, 45, 46] contains high-resolution fMRI data from 20 participants in response to prolonged auditory stimulation with the feature film "**Forrest Gump**" in **German**. In addition, it contains acquisitions, including raw and structurally aligned data, from the same participants with 25 music clips, with and without speech content, as stimuli. Moreover, for 7 participants, empirical ultra high-field fMRI data are included for orientation decoding in visual cortex. Finally, for 15 participants there are acquisitions for retinotopic mapping, a localizer paradigm for higher visual areas, and another 2 hour movie recording with simultaneous 1000 Hz eyetracking.

The dataset [47] includes fMRI scans where 90 participants were reading in their native language (**Farsi**, **Chinese** , **English** , 30 each) 40 short personal stories that had been collected from weblogs. Each story was roughly 150 words and was presented over the course of 3 slides of text, each displayed for 12 sec. The data is not publicly available.

The dataset [48] when it is released will include a corpus of translations of the children's story **The Little Prince** in 26 languages annotated with dependency graphs. Additionally, a subset of the corpus will be provided as time-aligned synthetic speech, generated using Google's Text-to-Speech Synthesis engine, along with corresponding EEG data for 20 participants.

The dataset [49] contains fMRI acquisitions from 29 **Chinese–Japanese** bilingual speakers who were asked to assess 48 pairs of images, 48 pairs of corresponding Chinese captions and 48 pairs of corresponding Japanese captions for coherence. The images depicted one or two people performing common daily activities and each pair was a sequence of coherent or incoherent events.

3.3 Mapping Methods

3.3.1 Voxel Selection

The number of voxels in a brain varies with respect to the voxel size and the shape of the subject's brain. The activity measured in many of these voxels is most likely not related to language processing, and might change due to physical processes like the noise perception in the scanner. In these cases, learning a mapping model from the stimulus representation to the voxel activation will not succeed because the stimulus has no influence on the variance of the voxel signal. Whole-brain evaluations of mapping models thus only have limited informative value. For this reason, effective voxel selection is crucial for extracting semantic information from brain data. In previous work, different voxel selection models have been applied to analyze only a subset of interesting voxels. We note that predominantly a gray matter mask is applied beforehand for an initial noise reduction and computational efficiency.

- Restricting the brain response to voxels that fall within a pre-selected set of regions of interests can be considered as a **theory-driven analysis**. In [11], they reduced the voxels by using previous knowledge about groups of regions of interests. Past experiments have found that a set of regions in the temporo-parietal and frontal cortices are activated in language processing and are collectively referred to as the language network. Group 1 is consistently activated across subjects when they listen to disconnected words or to complex fragments like sentences or paragraphs and group 2 is consistently activated only when they listen to complex fragments. Researchers in [50] select regions related to sentence comprehension.

- A more **information-driven** approach proposed by [51]. So-called searchlight analyses move a sphere through the brain to select voxels (comparable to sliding a context window over text) and analyze the predictive power of the voxel signal within the sphere.

- Mitchell [6] analyze all six brain responses for the same stimulus and select 500 voxels that exhibit a consistent variation in activity across all stimuli. A voxel can be represented by a matrix $M_u = T \times N$, where $T = 6$ is the number of trials, $N = 60$ is the number of stimuli. Each subject was shown the same words multiple times. Thus voxel stability s_u can be calculated as the average Pearson coefficient r for all trial pair combinations :

$$s_u = \frac{1}{\binom{T}{2}} \sum_{i=1}^{T} \sum_{j=i}^{T} (M_u[i,:], M_u[j,:]) \tag{3.1}$$

As noted by [52] for datasets where trials are not present (i.e. only one stimulus presentation per participant), a prediction driven metric can be used to select informative voxels. Notably [15] estimated a separate encoding model for each voxel and calculated model performance for a single voxel as the Pearson correlation coefficient between real and predicted responses. Gauthier and Ivanova [53] recommend to evaluate voxels based on explained variance. Lastly [12] use 10-PCA for low-dimensional representations.

The above prediction driven approaches are the most commonly adopted premises in the literature. As [54] mention, each area, represented by a voxel, responds largely independently of the other areas, thus a separate model is needed to fit responses in each cortical voxel.

3.3.2 Combining different voxel spaces

A common problem when working with neuroimaging data is combining activations across participants, in order to reduce noise and and compute a shared semantic representation for stimuli. Each subject's activations belong to a separate voxel space, and a multi-space alignment must be performed. For this purpose there are two types of alignment techniques.

(a) Anatomical alignment , to align voxel spaces to a common template using anatomical features from structural MRI. However shape, size, and spatial location of functional areas differ across subjects, motivating a **(b)** Functional alignment that maximises correlation of activations for the same stimulus across participants. Most commonly Hyperalignment (HA) [55] methods are used. At their core a Procrustes transformation maps neural activities in a shared high dimensional space, such as stimuli representations across subjects have the maximal correlation. It is considered an 'anatomy free' method.

Problem formulation : let T = number of stimuli , V = number of voxels, m = number of subjects. We define $\{\mathbf{X} \in \mathbb{R}^{T \times V}\}_{i=1}^{m}$ the fmri experiment. We assume that $\mathbf{X}_i$ matrices are aligned in time, i.e., row s of each $\mathbf{X}_i$ is recorded under the same stimulation.

The original HA objective can be expressed as :

$$\max_{\mathbf{R}_i, \mathbf{R}_j} \sum_{i=1}^{m} \sum_{j=i+1}^{m} tr(\mathbf{R}_i^T \mathbf{X}_i^T \cdot \mathbf{X}_j \mathbf{R}_j) \tag{3.2}$$

Intuitively we aim to find a rotation $R_i \in \mathbb{R}^{V \times V}$ of each subject's i voxel space, such that the correlation between all subjects is maximum. We assume X_i are noisy rotations of a common neural template Y. Although the mathematical formulation and goal of the problem are simple, the plethora of alignment methods across the literature suggest it is a challenging task. We attribute the difficulty to the highly idiosyncratic nature of brain semantics, and the low signal to noise ratio of fMRI.

In [8], they cast the problem to a probabilistic setting, with added dimensionality reduction. Each subject's voxel space X_i is modeled as a rotation of the shared response S plus an error term . The shared response for each stimulus is a latent variable observed by each subject's response. The optimization problem is solved by the EM algorithm. The model referred to as Shared Response Model (SRM) is used in later works [56] for fMRI sentence classification. Researchers in [57] observed that usually after alignment a supervised classification task is performed (e.g. regression), their model uses a two term loss function to compute alignment and fit supervision task at the same time, $\mathcal{L}_{total} = (1-)\mathcal{L}_{Align} + \mathcal{L}_{Sup}$ achieving a small increase in performance.

Following the idea that supervision can increase the quality of alignment, [58] leverage the class labels of some fMRI datasets by including the within class and between class covariance matrices in the optimization function. Supervised HA [59] extends this idea, by first mapping in a supervised space where each class has a distinct activation across subjects, and then projects to a neural space with distinct representations for each stimulus. The intermediate space contributes to better within class correlation with reduced computation. Finally [60], generalize to a non-linear transformation of voxel spaces using DNNs. Essentially this method can find a custom non-linear space for each subject and then align the neural activities form this non-linear space to a shared space. For an implementation of many HA methods we refer to easyfmri.

Our general assumption is that brains across subjects share a common semantic representation. Relying instead on the individual or culture-specific way people process semantic information, we can model cognitive embeddings as a mixture of cognitive distributions (e.g. a GMM). The individual responses can be identified by first computing the shared response and then removing it from each subjects space . Alternatively an eigenvalue decomposition could be used, concat all fmri images of v voxels to a matrix $X \in \mathbb{R}^{n \times v}$ and compute eigenvalue decompositionof XX^T .

Commonly, a simple average of activations across subjects or a selection of the best performing subject is also used in many fMRI experiments. Of course averaging activations without properly aligning voxel spaces will result in poor performance. A concatenation of low dimensional representations across subjects can also result in meaningful diverse features.

3.3.3 Voxel space ↔ Lexical embeddings

Due to the lack of large fMRI datasets, the most common method that is employed for obtaining lexical cognitive embeddings from fmri data is linear or ridge regression. However, neural networks have also been used. A model mapping from a lexical space to a voxel space is referred to as an encoder, and respectively a decoder in the opposite direction.

In [6, 7], the activation of voxel v for word w is given by

$$y_v(w) = \sum_{i=1}^{m} c_{v,i} f_i(w), \quad \forall v = 1 \cdots V, \tag{3.3}$$

where V is the total number of voxels, $f_i(w)$ is a function that estimates the association between seed word i and word w and $c_{v,i}$ are learned weights that are estimated via regression by utilising fMRI data for known words. Some authors (Anderson et al 2016) have also used similarity encoding, where the activation for an unknown word w is computed as a sum of activations of known words u_i, weighted by the similarity $sim(u_i, w)$.

Mitchell, for stimuli representation , used the co-occurrence similarity with 25 seed verbs manually selected with respect to psycholinguistic criteria and their relatedness to basic sensory and motor activities. Athanasiou [7] follows the same approach, deriving cognitive embeddings and evaluating their performance in NLP downstream tasks (MEN, ESSLLI, Sensicon, SNLI).

Several works evaluated the initial mapping by Mitchell, [61] report that by automatically choosing the set of verbs leads to equally good results. [62, 63] use WordNet based features for the 25 seed words, achieving comparable resutls. [64] conclude that no input representation is better overall at predicting brain activations, although morphological and dependency based models seem to perform better. [65] used a 65 experiential attribute that span different aspects of experience in neurobiological systems, ratings for each semantic dimension were crowd-sourced. [66] review many semantic models for input represenation, including dependency, association and image based. They conclude that visual information is a stronger predictor of brain activity than linguistic information for concrete nouns. [67] use low dimensional co-occurence vectors and sentence fMRI data to map words to cortical areas. They use a generative model, with a probability distirbution for semantic category clusters in the brain, and emission probabilities modeled as Gaussians.

Pereira [16] use a ridge regression to predict GloVe vectors from voxel activations. They show that a decoder learned in the isolated word setting, can accurately classify sentences from their fMRI with different levels of granularity. In contrast with earlier works the stimuli include abstract nouns. Recent works [68, 14] attempt to map conventional word embeddings (e.g. GloVe) to cognitive embeddings, using a neural network with one hidden layer. Specifically, [68] report that by using neural networks, both encoding and decoding accuracy is improved compared to a linear regression model on the same input.

For many datasets the stimuli consist of sentences, often from large narratives. Generally due to the low temporal resolution of fMRI no clear word boundaries exist, and an fmri Image corresponds to a set of words. Due to the sequential nature of the data, we could use an LSTM to map between word embeddings and neural activations. The common neural image for a set of words can be predicted as a function of the corresponding token hidden states . [12] address the low temporal resolution problem, by sliding a Gaussian window across tokens (acounting for Haymodynamic delay). They use the resulting representations with an HMM to improve performance in POS induction.

Researchers from [69, 70], conclude that LSTM sentence representations correlate well with brain data. [15] use a ridge regression on top of an LSTM pretrained for language modeling to map sentence stimuli to fMRI responses. They notice that LSTMs encode context and are better at predicting activations of individual words in a sentence.

Chapter 4

Experiments

4.1 Introduction

In our research, we'll try to combine word and brain representations to learn joint word-brain embeddings. In [9], they have recently introduced the first model specifically designed to capture the way the brain represents language meaning. By fine-tuning the BERT model to predict recordings of brain activity of people from the Harry Potter dataset [10], they ended up with representations that encode more brain-activity-relevant information and thus improve the quality of the brain activity prediction.

Furthermore [11] investigated how language models could learn from human brain. Their experiment consists of representations from 4 recent models: ELMO, BERT, USE and T-XL and data of brain scans from the Harry Potter dataset from [10]. Similar to [6, 7] they've performed linear regression from a model's layer with a sentence, *s*, as input, to predict the brain activation of the same sentence *s*. Based on the success of this method they were able to say if a layer share information with a predicted brain region and then go further to modify the layer and observing how the ability to predict the fMRI recordings changes. Their findings suggest that altering an NLP model to better align with brain recordings may lead to better language understanding by the NLP model, since they achieved better performance at NLP tasks with the altered version of the model than the base BERT architecture.

Our approach, based on the above, is trying to modify language models by incorporating cognitive embeddings in the training process. By checking how their ability to predict brain activity improves, we will try to achieve better language model performance at NLP tasks. As a basic step, we try adding the brain representation vector in the attention layer. This way the effect of poorly trained representations is mitigated. We also experiment with our models by adding the cognitive embeddings at only one layer at a time. Prior work from [7] shows very promising results when combining brain and word representations that even outperform the latter in some downstream tasks. To the best of our knowledge, this setting of combined embedding potential has never been utilized before and it may could be the way to take a step further towards brain activity aware language models.

4.2 Related Work

Following the usual approach for transfer learning in NLP, cognitive embeddings could be used to augment and increase performance of task-based models. The simplest approach is future-based, where cognitive embeddings are used as input with or without fusion with traditional embeddings. A pretrained model mapping words to cognitive space could also be fine-tuned end-to-end for a specific task. In [12], they use fMRI features derived from sentences combined with text features to increase perfomance in a POS tagging task.

The common approach for interpreting language model representations is by using specific NLP tasks, word annotations or behavioral measures. Some researchers used fine-tuned language models to predict brain activity and evaluate the brain representations. Such fine-tuning is a new paradigm in learning about human language processing and it relies on encoding information from targets of a prediction task (e.g. the brain representations in our case) into the model parameters. The goal is to optimize these models to take advantage of multiple sources of information about language processing in the brain.

There is little prior work that evaluates or improves NLP models through brain recordings. [13] proposes to evaluate whether a word embedding contains cognition-relevant semantics by measuring how well they predict eye tracking data and fMRI recordings. Similarly [14] proposed a framework for intrinsic word embedding evaluation based on how much they reflect brain semantics. Six types of word embeddings were evaluated by regressing on fMRI, EEG and eye tracking data. They report corellation between the cognitive evaluation and perfromance in Named-entity recognition (NER) and Question Answering tasks.

Jain and Huth [15], aligned layers from a Long Short-Term Memory (LSTM) model to predict fMRI recordings of subjects listening to stories to differentiate between the amount of context maintained by each brain region. [11] used brain activity recordings to show that different network representations encode information relevant to language processing at different context lengths. Both [11, 9] observed that by modifying the pretrained BERT model to better capture brain-relevant language information they achieved higher accuracy results at NLP tasks. This finding suggests that altering an NLP model to better align with brain recordings of people processing language may lead to better language understanding by the NLP model.

4.3 Datasets

In the first part of our experiments we compare cognitive and word embeddings by utilizing the dataset presented by **Mitchell** [6] for the neural representations. As we mentioned in Chapter 3, this dataset contains fMRI scans from 9 participants, and stimuli are line drawings and noun labels of 60 concrete objects from 12 semantic categories. All stimuli were presented 6 times during the scanning session, in a different random order each time. Participants were asked to think of the same item properties across the 6 presentations.

For the comparison of the embeddings we use the **MEN** Test Collection dataset. It contains two sets of English word pairs (one for training and one for testing) together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk via the CrowdFlower interface. This collection is commonly used to test models on semantic similarity and relatedness measures.

To extract our **cognitive embeddings** for Sections 4.5 and 4.7, we use the dataset introduced by Pereira [16], where they wanted to evaluate abstract concepts and sentence fMRIs as we described is Chapter 3. We use the data from the first experiment which consists of fMRI scans from 16 participants. The stimuli consist of 180 concept words selected to cover a big part of the semantic space. Each word represents a cluster of words based on Glove vector space (300 dim). The stimuli were shown in three paradigms with multiple repetitions, in a sentence, as an image, or in a word cloud. These prepossessed data for each participant consists of an array : words (180) x voxels ($\sim$ 200.000) and mapping indexes for 3D voxel and vector spaces.

To finetune our language models, we use the **WikiText** language modeling dataset [17], which is a collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. The dataset is available under the Creative Commons Attribution-ShareAlike License. Compared to the preprocessed version of Penn Treebank (PTB), WikiText-2 is over 2 times larger and WikiText-103 is over 110 times larger. The WikiText dataset also features a far larger vocabulary and retains the original case, punctuation and numbers - all of which are removed in PTB. As it is composed of full articles, the dataset is well suited for models that can take advantage of long term dependencies.

For the fMRI prediction task we utilize the **Harry** dataset by [10] which includes magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI) data recorded from people as they read a chapter from Harry Potter. The chapter included 5176 words and was recorded from nine participants for each experiment. For the MEG experiment data for one participant had too many artifacts and was excluded, leaving 8 participants.

4.4 Experiments with Cognitive Embeddings

In the following experiments we aim to ivestigate the potential of cognitive embeddings by comparing them with known word embeddings like Word2vec [18]. We use ridge regression to learn the corresponding representation of each word into the cognitive embeddings vector space, following the pipeline from [6, 7]. For the voxel selection we use the average Pearson coefficient of all trials to sort the voxels based on their stability and then choose the 500 with best stability scores.

4.4.1 Encoding Model

For the encoding model, we map the representation of 60 stimuli words from [6] to fMRI. For the initial model we learn a mapping :

$$y_u = \sum_{i=1}^{D} c_{u,i} \cdot s_i \tag{4.1}$$

or in matrix form :

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{s} \tag{4.2}$$

Where **y** are the voxel activations for 500 stable voxels, **s** is the glove embedding vector, **W** the learned regression matrix.

A voxel can be represented by a matrix $M_u = T \times N$, where $T = 6$ is the number of trials, $N = 60$ is the number of stimuli. Each subject was shown the same words multiple times. Thus voxel stability s_u can be calculated as the average Pearson coefficient r for all trial pair combinations:

$$s_u = \frac{1}{\binom{T}{2}} \sum_{i=1}^{T} \sum_{j=i}^{T} (M_u[i,:], M_u[j,:]) \tag{4.3}$$

We select the 500 most stable voxels for each subject.

We evaluate the mapping as a **leave-out 2 procedure**: for all $\binom{60}{2}$ pairs, we train on 58 words and validate on 2 remaining. Correct prediction means that sum of the cosine similarities of the correct matched pairs is greater than the false matched pair:

$$cos(p_1, i_1) + cos(p_2, i_2) > cos(p_2, i_1) + cos(p_1, i_2) \tag{4.4}$$

where p_1, p_2 are the model predictions associated with ground truth fMRI i_1, i_2

.

We note that this metric lacks strictness, and could be replaced by strict matching:

$$cos(p_1, i_1) > cos(p_2, i_1) \cap cos(p_2, i_2) > cos(p_2, i_1) \tag{4.5}$$

or similar metrics, as noted by [52].

We report the accuracy of correct predictions for each participant:

Subject	Ridge Reg.	Athanasiou	Mitchell
1	0.91	0.84	0.83
2	0.69	0.82	0.76
3	0.80	0.76	0.78
4	0.87	0.79	0.72
5	0.75	0.78	0.78
6	0.60	0.65	0.85
7	0.76	0.75	0.73
8	0.66	0.68	0.68
9	0.76	0.68	0.82
average	0.75	0.75	0.77

Table 4.1. *We map directly from Glove embedding vectors to 500 stable voxels and compare with Athanasiou and Mitchell*

Results are comparable with [7]. The Difference in results is attributed to the different similarity function $f_i(w)$ calculation. Athanasiou sets $f_i(w)$ as the (normalized) co-occurrence frequency of the ith seed s_i and word w, as shown in Figure 4.1, estimated on a large corpus of results of web queries to Yahoo.

Overall the encoding model, semantic or glove space, does not greatly affect performance.

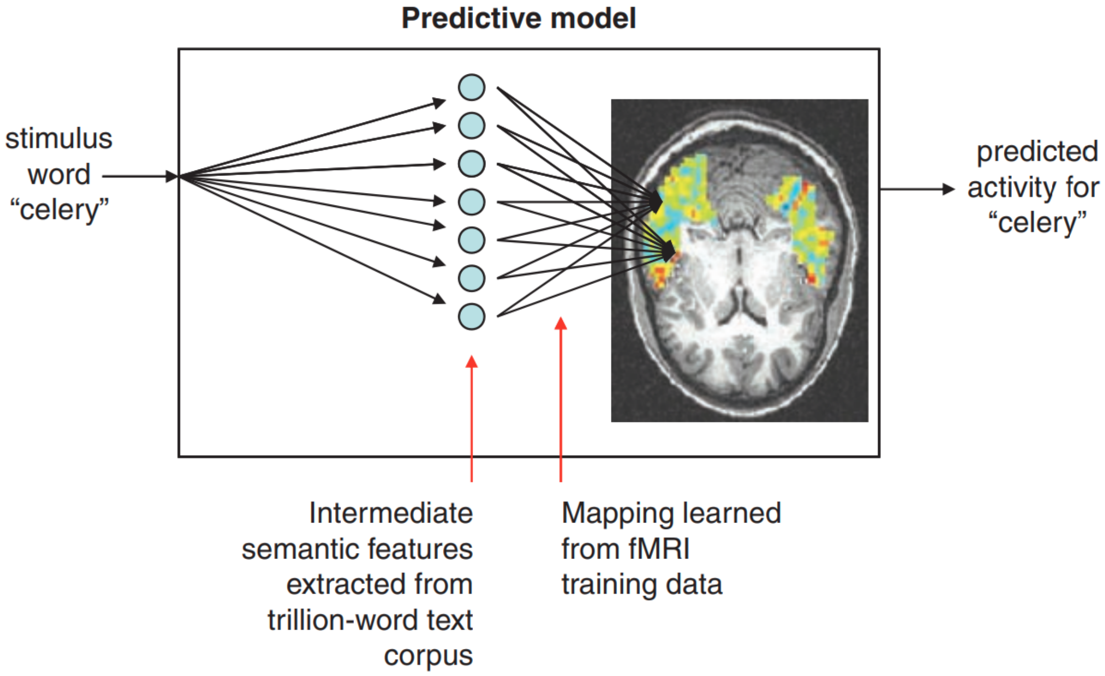

Predictive model
stimulus word "celery"
predicted activity for "celery"
Intermediate semantic features extracted from trillion-word text corpus
Mapping learned from fMRI training data

Figure 4.1. *Intermediate semantic feature model, each word w is mapped to a vector $< f_1(w), f_2(w), ..., f_{25}(w) >$, where $f_i(w)$ is the similarity of w with semantic feature s_i. This is the word representation used for the encoding task in [6]*

4.4.2 Comparing Cognitive and Traditional Embeddings

We evaluate our neural encoding model presented above with a linear mapping, in the MEN similarity dataset. We use the widely tested Mitchell dataset, and GloVe embeddings for stimuli representation as mentioned before.

The MEN Test Collection contains English word pairs with human-assigned similarity judgments. Following the work of Athanasiou [7] we calculate the similarity of two words, w_1, w_2 with respect to the neural model as follows:

$$sim(w_1, w_2) = \sum_{u=1}^{V} b_u(y_u(w_1) - y_u(w_2))^2 \tag{4.6}$$

This takes the form of a weighted euclidean distance, where the weights **b** are determined by regression on the MEN train-set.

For our experiment we selected only **concrete noun** pairs , as neural embeddings have been shown to work better with concrete words . We used concreteness ratings in a scale of 5 from (ratings), selecting only words with concreteness $>$ 4.2. This resulted in 1161 training pairs (547 unique words) and 577 test pairs (455 unique words). We selected 86 similar and 37 disimilar pairs after thresholding on similarity with 0.85 and 0.1. respectively. The metric for all evaluations is the Spearman correlation.

We compare our approach with a baseline using w2vec 300-dim vectors. We present our basic encoding model, where we use regression on the fMRI data of one participant each time, choosing the best 500 voxels with the aforementioned method of stability scores. For one of our experiments we use the encoding model on the averaged data from all participants. In addition, the method of Hyperalignment, explained in Chapter 3, is used with the Shared Response Model on its own for the similarity task and then in combination with w2vec embeddings with the finals representations being the average. Results are reported on Table 4.2.

Subset	w2vec	500 vox	avg (200 vox)	SRM	SRM - w2vec
All Concrete	0.73	0.67 (0.69)	0.63	0.66	**0.74**
Most & Least Sim	0.60	0.57 (0.62)	0.53	0.64	**0.65**
Least Similar	**0.21**	0.14 (0.36)	0.06	0.1	0.11
Most Similar	0.09	-0.02 (0.19)	0.20	0.20	**0.21**

Table 4.2. *Spearman coefficient for the test-set for different subsets of concrete nouns, and different embeddings. We report the mean score across participants, values in parentheses* $(\cdot)$ *indicate the maximum across participants. For the neural averaged (column avg) we first averaged embeddings from all participants and then proceeded to regression for similarity. SRM refers to the Shared Response Model for Hyperalignment [8]*

Overall the benefits of using cognitive embeddings as features for computational task are not substantial resulting in a small increase in performance. A close analysis of related work by [7] shows that neural data encode useful semantic information, but a feature-based approach with a linear regression mapping may not be the best way to exploit it. Our experiments are promising but we have not yet established the unique flavor of neural representation that is complementary to that of distributional embeddings.

4.5 Modifying Language Models with cognitive embeddings

After testing cognitive embeddings on downstream tasks, we leverage from the existing literature and propose trying to modify Language Models by adding cognitive embeddings in their existing architectures. Our experiments focus on adding the cognitive embeddings in the Attention Layer in a variety of ways. We choose this method due to the great noise and small amount of samples that defines fMRI data, making these representations a bad candidate for the training or fine-tuning of a language model. First, we extract our cognitive embeddings with the methods described in Section 4.4. Then we test our proposed method on an LSTM model, which due to its smaller size allows us to add the cognitive embeddings in the training process and not just during the fine-tuning. Finally, BERT [5] is altered in many different ways and then fine-tuned on the Masked Language Modeling task.

4.5.1 Extracting Cognitive Embeddings

To begin with, we have to obtain the congitive embeddings for our experiment. We use the dataset from the experiment 1 from [16], which consists of 18 participants and 180 stimuli words for each one of them. The stimuli were shown in three paradigms with multiple repetitions, in a sentence, as an image, or in a word cloud. As a result, we end up with 3 brain representations for each word so we imitate the approach from [6] and calculate the stability of each voxel as the average Pearson coefficient between these 3 trials. Thus, in this case each voxel can be represented by a matrix $M_u = T \times N$, where $T = 3$ is the number of "trials", $N = 180$ is the number of stimuli. We map the representation of the 180 stimuli words to fMRI, as explained before. For the initial model we learn a mapping:

$$y_u = \sum_{i=1}^{D} c_{u,i} \cdot s_i \tag{4.7}$$

or in matrix form :

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{s} \tag{4.8}$$

Where **y** are the voxel activations for 500 stable voxels, **s** is the glove embedding vector, **W** the learned regression matrix.

After training our model, we produce representations for all words in our models vocabulary. For each subject and each word we select the 500 most stable voxels, as explained above, since with this method we achieve the best results overall on the similarity task. Finally for each word we get the mean representations across all subjects so that we end up with our cognitive word embeddings.

4.5.2 Condition an LSTM Language Model

For our first experiment, we use an LTSM based language model from [71]. The model uses a single attention layer and a modified feedforward layer similar to that in a Transformer, which is referred as Boom layer. Scaled dot-product attention [4] is used, which utilizes three matrices, K, V and Q to calculate the attention as:

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

Given a dictionary we may consider the keys and values be represented by matrices K and V, while Q is a query that contextualizes the attention weights (d_K denotes the dimensionality of the keys and queries).

We modify this attention mechanism by using the cognitive embeddings of the input sequence as our query matrix Q. For every input word of our vocabulary we get its cognitive embedding by utilizing the regression encoding model described before. In Figure 4.2 we present the complete architecture of our model.

We train three models on WikiText-2, on the Language modeling task. All three models are trained with batch size 8 for 32 epochs with learning rate $2e - 3$ and then fine-tuned for 5 epochs with learning rate $e - 3$ as proposed by model's author [1].

1. A **base SHA-RNN** model with a single attention layer.

2. A model **finetuned** with cognitive embeddings in the attention layer for 5 epochs.

3. A model **trained all the way** with cognitive embeddings in the attention layer.

[1] https://github.com/Smerity/sha-rnn

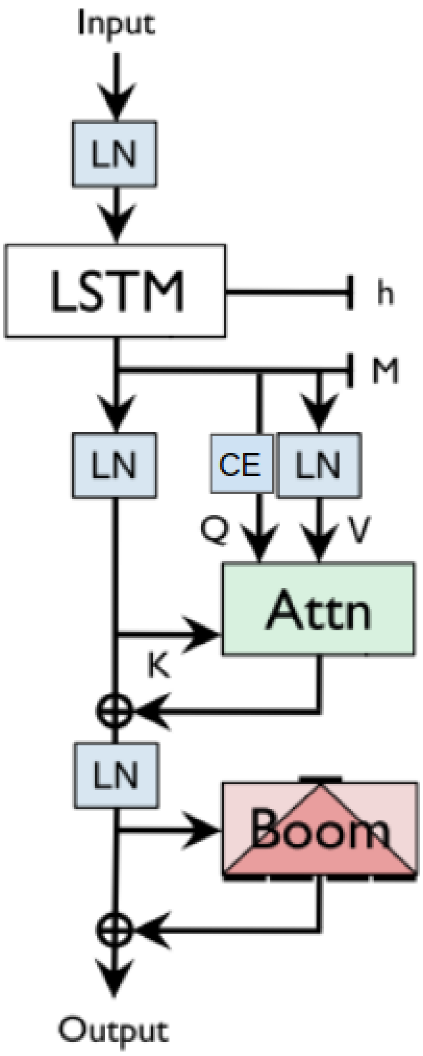


Figure 4.2. *The SHA-RNN is composed of an RNN, pointer based attention, and a "Boom" feed-forward with layer normalization. The CE layer before Q stands for the cognitive embeddings.*

4.5.3 Condition BERT

For our next experiment, we use BERT [5], one of the widely adopted pre-training approach for model initialization, the architecture of which is the encoder of Transformer [4]. In BERT training the authors used two kinds of objective function: (1) Masked language modeling (MLM), where 15% words in a sentence are masked and BERT is trained to predict them with their surrounding words. (2) Next sentence prediction (NSP), where BERT is trained to predict whether two input sequences are adjacent.

Following the approach we explained before, we aim to modify BERT in order to incorporate the cognitive embeddings in its structure. Apart from a simple

$BERT_{base}$ model, as a baseline, we first try to add cognitive embeddings in the embedding layer, imitating the concept of positional embeddings [4].

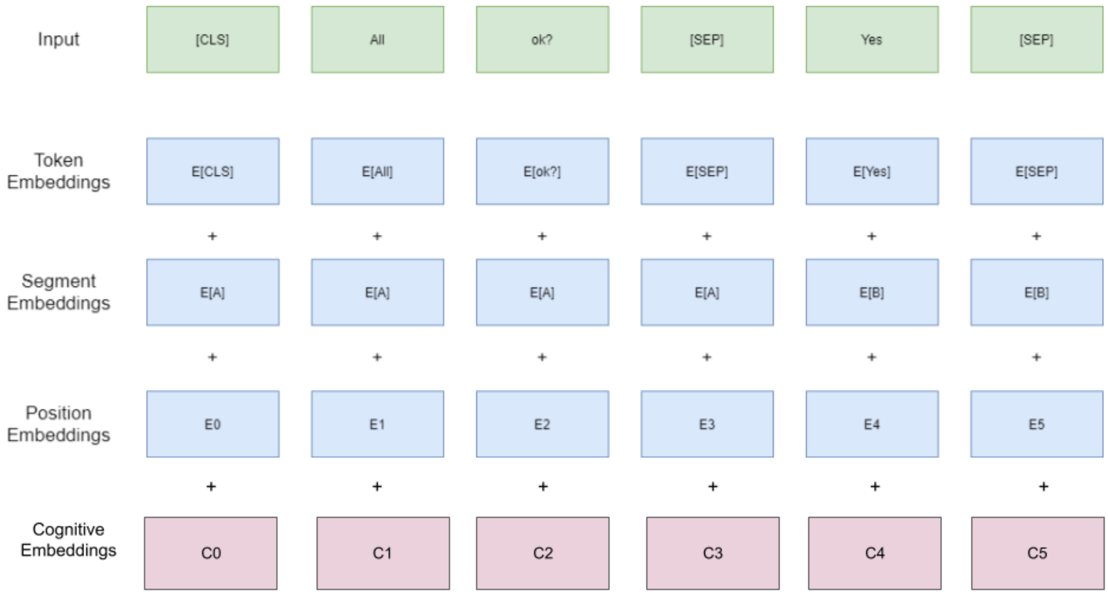
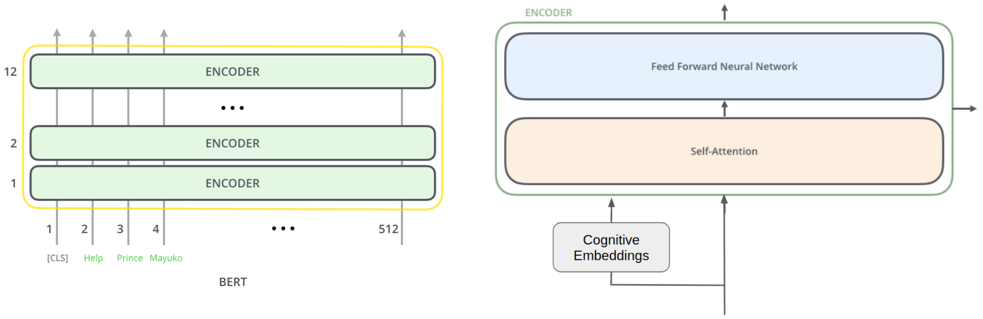


Figure 4.3. *Adding cognitive embeddings alongside positional and segment embeddings.*

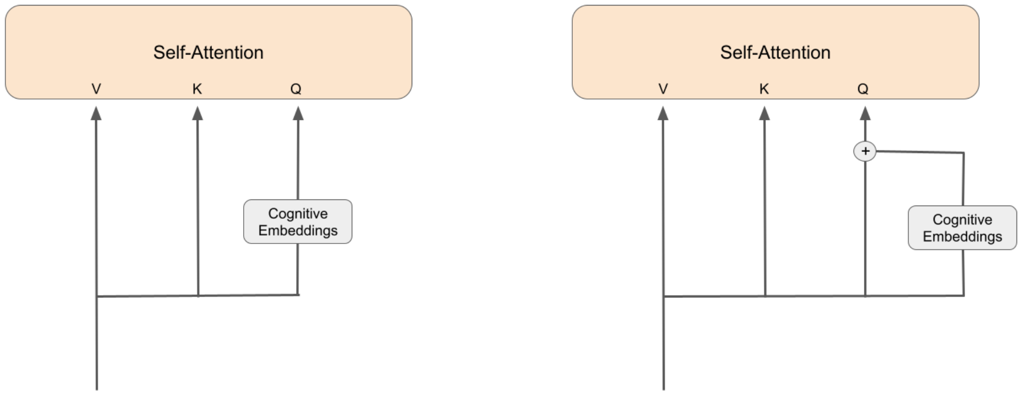
We leverage from the pre-trained $BERT_{base}$ architecture, which consists of 12 Transformer layers, and we modify each self-attention layer by using our cognitive embeddings in the query matrix Q. We propose two architectures, one where the query matrix Q consists only of our cognitive embeddings and one where we add our embeddings to the existing BERT embeddings.



(a) *$BERT_{base}$ architecture*

(b) *Transformer encoder structure*

Figure 4.4. *We keep the original BERT architecture and just add the cognitive embeddings in each Self Attention layer.*

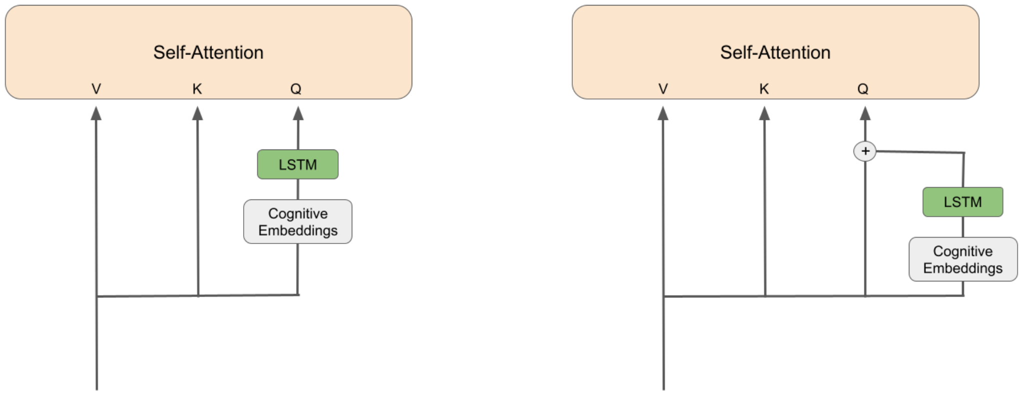


(a) *Replacing the embeddings in query matrix Q with cognitive embeddings*

(b) *Adding cognitive embeddings in query matrix Q*

Figure 4.5. *Detailed analysis of our modifications in BERT's self-attention layer, for each of our models.*

We also try to reduce the natural noise that exists in fMRI data by passing our cognitive embeddings, first through an LSTM layer in our two aforementioned architectures.



(a) *Replacing the embeddings in query matrix Q with cognitive embeddings, after passing them through an LSTM layer*

(b) *Adding cognitive embeddings in query matrix Q, after passing them through an LSTM layer*

Figure 4.6. *Architecture of our models after adding an LSTM layer for the cognitive embeddings to pass through.*

Our last approach, is trying to add cognitive embeddings in only one attention layer of the BERT model each time. A similar set-up was used in [11], where after observing that the layers in the first half of the base BERT model benefit from uniform attention for predicting brain activity, they tested how the same alterations affect BERT's ability to predict language by testing its performance on natural language processing tasks. Their findings suggest that it's better to alter attention in layers 1 through 6, a single layer at a time. Leveraging from their work, we add our cognitive embeddings in layers 1,2, and 6. Also a model with cognitive embeddings in the attention at layer 11 is used to contrast the performance of the other layers. For this modification we use the architecture where we add cognitive embeddings to the existing BERT embeddings in the attention layer, since this approach gets better results overall.

All our models are fine-tuned for 4 epochs in the Masked language modeling (MLM) task, on the WikiText-2 dataset. In conclusion, for our results we'll use:

1. A simple $BERT_{base}$ model as a baseline.
2. A $BERT_{base}$ where we add cognitive embeddings in the embedding layer alongside with positional embeddings.
3. A cognitive-BERT model fine-tuned **only** with cognitive embeddings in the query matrix Q.
4. A cognitive-add-BERT model after fine-tuning where we **add** cognitive embeddings in the query matrix Q.
5. A cognitive-BERT-LSTM model fine-tuned **only** with cognitive embeddings in the query matrix Q, after passing them through an LSTM layer first.
6. A cognitive-add-BERT-LSTM model after fine-tuning where we **add** cognitive embeddings in the query matrix Q, where we first pass them through an LSTM layer.
7. A cognitive-add-1 model, with cognitive embeddings only in the attention layer 1.
8. A cognitive-add-2 model, with cognitive embeddings only in the attention layer 2.
9. A cognitive-add-6 model, with cognitive embeddings only in the attention layer 6.
10. A cognitive-add-11 model, with cognitive embeddings only in the attention layer 11.

4.6 Predict fMRI's

In this segment of our work, we'll evaluate our LSTM models' performance by utilizing the setup from [11] on the Harry dataset, as we mentioned in 4.1. We only use our LSTM models due to smaller training times. Our goal is to check whether by inducing brain data into a language model, it is possible for its representations to consist of more brain relevant information.

The representation of a language model given an input sentence s, is mapped to the brain activity that corresponds to the same sentence s. The words presented to the participants one at a time at a rate of $0.5s$ each and every fMRI was acquired at a rate of $2s$. Therefore, the features of contiguous words are first grouped, by the interval in which they were presented, and averaged to get one final representation for each fMRI. Finally, PCA is applied for dimensionality reduction, before we get the average features.

For the prediction of each neural image we use a concatenated vector z_t, formed of 4 previous features $[x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}]$, where x_t is the feature of the words corresponding to the fMRI y_t, at time t. We include these features from previous volumes in order to account for the hemodynamic delay which is measured around $6s$. Afterwards, we use a separate ridge regression to predict each voxel activation, similar to what we've done for our cognitive embeddings extraction. The regularization parameter for each voxel is chosen by a 10-fold CV independently.

Finally, to evaluate our models, we classify a contiguous chunk of real data, of length 20 time intervals, with a variation of pairwise classification, as commonly done in [6, 16, 7]. Because our experiment doesn't have multiple repetitions for each fMRI, we raise the number of the time intervals we use at a time, to avoid the close to chance accuracy which the noisy fMRI data are likely to give us.

Subject	Base	Finetuned	Trained
1	0.54	0.56	**0.57**
2	**0.61**	0.59	0.6
3	0.64	0.64	0.64
4	**0.51**	0.5	0.5
5	0.56	0.57	**0.57**
6	**0.61**	0.6	0.6
7	0.59	0.59	0.59
8	0.6	0.62	**0.63**
average	0.58	0.58	**0.59**

Table 4.3. *We compare our three models based on how well they predict brain activation.*

4.7 Downstream Tasks

After modifying BERT we test how these alterations affect its ability to predict language by testing its performance on natural language processing tasks. We run our models on seven downstream tasks and compare their results.

4.7.1 The Corpus of Linguistic Acceptability

CoLA The Corpus of Linguistic Acceptability [72] consists of English acceptability judgments drawn from books and journal articles on linguistic theory. Each example is a sequence of words annotated with whether it is a grammatical English sentence. Following the authors, we use Matthews correlation coefficient [73] as the evaluation metric, which evaluates performance on unbalanced binary classification and ranges from -1 to 1, with 0 being the performance of uninformed guessing.

base SHA-RNN	finetuned	trained
0.35	0.09	0.01

Table 4.4. *Comparing all LSTM models on CoLA.*

$BERT_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
0.578	0.0	0.012	0.21	0.0	0.267

Table 4.5. *Comparing cognitive-BERT models with vanilla BERT on CoLA.*

$BERT_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
0.578	0.08	0.397	0.431	0.296

Table 4.6. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on CoLA.*

4.7.2 The Stanford Sentiment Treebank

SST-2 The Stanford Sentiment Treebank (Socher et al., 2013) consists of sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence. We use accuracy as the evaluation metric.

base SHA-RNN	finetuned	trained
0.9	0.73	0.67

Table 4.7. *Comparing all LSTM models on SST-2.*

$BERT_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
0.917	0.777	0.802	0.915	0.813	0.901

Table 4.8. *Comparing cognitive-BERT models with vanilla BERT on SST-2.*

$BERT_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
0.917	0.916	0.916	0.913	0.912

Table 4.9. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on SST-2.*

4.7.3 Microsoft Research Paraphrase Corpus

MRPC The Microsoft Research Paraphrase Corpus [74] is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent. Because the classes are imbalanced (68% positive), we follow common practice and report both accuracy and F1 score.

metric	base SHA-RNN	finetuned	trained
acc.	**0.78**	0.678	0.543
f1	**0.84**	0.713	0.562

Table 4.10. *Comparing all LSTM models on MRPC.*

metric	$BERT_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
acc.	**0.863**	0.703	0.705	0.698	0.691	0.693
f1	**0.907**	0.82	0.821	0.817	0.815	0.816

Table 4.11. *Comparing cognitive-BERT models with vanilla BERT on MRPC.*

metric	$BERT_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
acc.	**0.863**	0.703	0.708	0.705	0.703
f1	**0.907**	0.82	0.822	0.82	0.82

Table 4.12. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on MRPC.*

4.7.4 Semantic Textual Similarity Benchmark

STS-B The Semantic Textual Similarity Benchmark [75] is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated with a similarity score from 1 to 5 and the task is to predict these scores. Follow common practice, we evaluate using Pearson and Spearman correlation coefficients.

metric	base SHA-RNN	finetuned	trained
pears.	**0.79**	0.34	0.13
spear.	**0.79**	0.321	0.156

Table 4.13. *Comparing all LSTM models on STS-B.*

metric	$BERT_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
pears.	**0.913**	0.068	0.105	0.825	0.109	0.841
spear.	**0.91**	0.035	0.074	0.823	0.098	0.839

Table 4.14. *Comparing cognitive-BERT models with vanilla BERT on STS-B.*

metric	$BERT_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
pears.	**0.913**	0.833	0.832	0.832	0.833
spear.	**0.91**	0.833	0.831	0.831	0.831

Table 4.15. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on STS-B.*

4.7.5 Question NLI

QNLI The Stanford Question Answering Dataset [76] is a question-answering dataset consisting of question-paragraph pairs, where one of the sentences in the paragraph (drawn from Wikipedia) contains the answer to the corresponding question (written by an annotator). [77] converted the task into sentence pair classification by forming a pair between each question and each sentence in the corresponding context, and filtering out pairs with low lexical overlap between the question and the context sentence. The task is to determine whether the context sentence contains the answer to the question. This modified version of the original task removes the requirement that the model select the exact answer, but also removes the simplifying assumptions that the answer is always present in the input and that lexical overlap is a reliable cue. This process of recasting existing datasets into NLI is similar to methods introduced in [78] and expanded upon in [79]. They call the converted dataset QNLI (Question-answering NLI). As an evaluation metric for this task, we use accuracy.

base SHA-RNN	finetuned	trained
0.798	0.678	0.53

Table 4.16. *Comparing all LSTM models on QNLI.*

$BERT_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
0.893	0.611	0.865	0.87	0.633	0.878

Table 4.17. *Comparing cognitive-BERT models with vanilla BERT on QNLI.*

$BERT_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
0.893	0.874	0.877	0.877	0.876

Table 4.18. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on QNLI.*

4.7.6 Recognizing Textual Entailment

RTE The Recognizing Textual Entailment (RTE) datasets come from a series of annual textual entailment challenges. [77] combined the data from RTE_1 [80], RTE_2 [81], RTE_3 [82], and RTE_5 [83]. Examples are constructed based on news and Wikipedia text. They converted all datasets to a two-class split, where for three-class datasets they collapsed *neutral* and *contradiction* into *not_entailment*, for consistency. For our results we use accuracy.

base SHA-RNN	finetuned	trained
0.592	0.532	0.511

Table 4.19. *Comparing all LSTM models on RTE.*

$BERT_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
0.714	0.469	0.537	0.545	0.534	0.588

Table 4.20. *Comparing cognitive-BERT models with vanilla BERT on RTE.*

$BERT_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
0.714	0.548	0.555	0.563	0.556

Table 4.21. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on RTE.*

4.7.7 Winograd NLI

WNLI The Winograd Schema Challenge [84] is a reading comprehension task in which a system must read a sentence with a pronoun and select the referent of that pronoun from a list of choices. The examples are manually constructed to foil simple statistical methods: Each one is contingent on contextual information provided by a single word or phrase in the sentence. To convert the problem into sentence pair classification, [77] constructed sentence pairs by replacing the ambiguous pronoun with each possible referent. The task is to predict if the sentence with the pronoun substituted is entailed by the original sentence and accuracy is recommended as an evaluation metric.

base SHA-RNN	finetuned	trained
0.651	0.543	0.512

Table 4.22. *Comparing all LSTM models on WNLI.*

BERT$_{base}$	Emb-layer	Cogn	Cogn-add	Cogn-lstm	Cogn-add-lstm
0.436	**0.563**	0.408	0.422	0.408	0.478

Table 4.23. *Comparing cognitive-BERT models with vanilla BERT on WNLI.*

BERT$_{base}$	Cogn-add-1	Cogn-add-2	Cogn-add-6	Cogn-add-11
0.436	**0.464**	0.437	0.45	0.422

Table 4.24. *Comparing cognitive-BERT models, with only one modified attention layer each time, with vanilla BERT on WNLI.*

4.8 Experimental Discussion

Overall, the method we used in section 4.6, proposed by [11], shows that natural language models incorporated with cognitive embeddings may contain some relevant brain information. Our LSTM model's representations, after a full training with cognitive embeddings in the attention layer, achieve better results at predicting brain activity of subjects reading complex natural text. This method of neural network representations and brain activity alignment, indicates that our setup may lead to better language understanding by the NLP model.

In section 4.7, we first test the LSTM language models on the CoLA dataset and notice a drop in the performance after fine-tuning the model with cognitive embeddings, and an even bigger drop when the model is trained all the way with the cognitive embeddings in the attention layer. This shows us that the robustness of the model is negatively affected by the noisy brain data. CoLA is one of the tasks where even the base SHA-RNN model doesn't achieve great results, thus it struggles to sustain its efficiency when we incorporate these neural representations in the training process.

The BERT experiments for this task confirm that the cognitive embeddings play a negative role for the model's attempt to understand if a sentence is grammatically correct. This becomes more clear when we realize that the less the cognitive embeddings take part in the training process the better the results we achieve. That's the reason why the models where we add cognitive embeddings alongside the actual BERT embeddings manage to achieve slightly better results.

For the SST-2 task, all our models show about as good performance as the base models. The trained with cognitive embeddings SHA-RNN model seems to be affected the most from noise existing in the neural representations. Some of the BERT models achieve approximately the same accuracy as the standard BERT, but we suspect that this results are due to the fact that the base models are extremely good at predicting the sentiment of a given sentence on their own. On the other hand, the almost good results of the cognitive models on this task may come from the existence of sentiment in the human brain and therefore in the neural representations as well.

In the same context as before, for the Microsoft Research Paraphrase Corpus, the base models set great scores as a baseline. Once again the models where the fMRI data take a bigger role in their training process seem to lose their robustness and efficiency. Even the models with the cognitive embeddings at only one attention layer each time appear to drop their performance the same.

Furthermore, our models, for this STS-B, need to predict the right score from 1 to 5 with the cognitive embeddings appearing to drop their results even below random, the more they participate in the whole structure. All the models where we add the cognitive embeddings with the BERT embeddings manage to achieve approximately good results.

In addition, the QNLI results are very close for the majority of our models. This may be due to the congnitive nature of this task since Question Answering is one of the most familiar tasks of the human brain.

Moreover, one of the tasks where cognitive embeddings gets completely destroyed by the base models is the Recognizing Textual Entailment (RTE). All the cognitive models reach accuracy close to random with even the models where the cognitive embeddings were added at only one layer to show equally bad results.

Finally, because of the bad performance of the base BERT model in WNLI task, we encounter a situation where some of our cognitive models seem to achieve better results over all. Although, all the scores are close and even below random accuracy so we can not judge from this slight improvement to decide if our cognitive embeddings improve the model over all.

Chapter 5

Conclusions

In this Diploma Thesis, we investigated the potential of fMRI data in Natural Language Processing. From our work, presented in the Chapters before, we draw some conclusions that can be devided into two main categories corresponding to Sections 4.4 and 4.5 respectively.

5.1 Discussion

5.1.1 Experiments with Cognitive Embeddings

In section 4.4, we extend the work from [6, 7] by utilizing the dataset from Mitchell [6] and Glove [25] embeddings for stimuli representation. We adopt the same pipeline for the voxel selection and compare the performance of cognitive embeddings and lexical embeddings in downstream tasks. First, we test the encoding model with ridge regression directly from glove embeddings instead of an intermediate semantic feature model [7], and evaluate it with a leave-out 2 procedure. Then, we compare this encoding model, in several variations, with traditional word embeddings on the MEN dataset.

We conclude that, the performance of the encoding model, is not affected overall from the semantic or glove space. Furthermore, for the similarity task, all the versions of the model achieve similar results with the combination of Word2vec and the Shared Response Model representations to attain slightly better scores. Generally, the cognitive embeddings are not clearly the best candidates as features for computational tasks, based on the aforementioned results. These experiments, although they are very promising, can not fully exploit the potential of cognitive embeddings.

5.1.2 Modifying Language Models with cognitive embeddings

First, after reviewing cases where cognitive embeddings have been used to enhance or improve task-based models, we focused on investigating how these representations could affect a language model and which modifications are a better fit in order for them to reveal their potential. BERT [5] was the focal point of our research. We based the majority of our experiments on the assumption that a good way to incorporate cognitive embeddings, into a language model's architecture, is by adding them as queries in the attention layer. This way, we could induce the cognitive bias of these embeddings into the training process and also mitigate the effect of the poor brain representations. We first test our approach on a smaller LSTM model where, by utilizing the experiment setup from [11], we find that its ability to predict brain recordings improves and that may leads to an improvement of the models performance at NLP tasks.

Nonetheless, our results at NLP tasks indicate that, even the complex BERT architecture is negatively affected by the noisy neural representations. Our experiments on smaller models show that for the tasks where the base model already achieves good results, its performance is maintained even with the cognitive embeddings in the attention layer. We show this in more detail in Section 4.7, where our proposed cognitive BERT models completely lose their efficiency for some tasks due to the poor brain representations. One last note for our work is that when we tested the cognitive embeddings modification on different layers of the BERT model we demonstrated that the mid layers better distribute the noisy brain information in comparison with the earlier and the former layers, exactly as [11] had mentioned. Overall, our setup seems to lack the ingredients that the neural representations need in order to provide competitive results in the frame of modern natural language processing techniques.

5.2 Future Work

On the way to discover the potential of cognitive data in Natural Language Processing, we came across interesting future directions, that may hide an unexplored potential. These are briefly described below:

- Incorporate cognitive embeddings into more natural language processing models with different architectures.

- Extend the work on BERT by trying more complex setups, such as the Multimodal Adaptation Gate they proposed in [85] that allows BERT to accept multimodal nonverbal data during fine-tuning.

- Explore an evaluation of the cognitive models on tasks that do not require fine-tuning beyond pretraining to ensure that there is an opportunity to transfer the insight from the brain interpretations of the pretrained BERT model. In [11], they used a range of syntactic tasks proposed by [86], in order to quantify BERT's syntactic capabilities.

- Develop a pipeline for extracting cognitive embeddings from different fMRI datasets that present some good results, like the Harry dataset [10].

- Use sentence fMRI's, such as in [10, 40], in order to extract cognitive embeddings. This way, it could be possible to obtain embeddings with natural language context flavor.

Bibliography

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv:1301.3781 [cs]*, Sept. 2013. arXiv: 1301.3781.

[2] “Simple Anatomy of the Retina by Helga Kolb – Webvision.”

[3] J. Alammar, “The Illustrated Transformer.”

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019.

[6] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, “Predicting human brain activity associated with the meanings of nouns,” *Science*, vol. 320, pp. 1191–1195, May 2008.

[7] N. Athanasiou, E. Iosif, and A. Potamianos, “Neural activation semantic models: Computational lexical semantic models of localized neural activations,” in *Proceedings of the 27th International Conference on Computational Linguistics*, Aug. 2018.

[8] P.-H. C. Chen, J. Chen, Y. Yeshurun, U. Hasson, J. Haxby, and P. J. Ramadge, “A reduced-dimension fmri shared response model,” in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 460–468, Curran Associates, Inc., 2015.

[9] D. Schwartz, M. Toneva, and L. Wehbe, “Inducing brain-relevant bias in natural language processing models,” in *Advances in Neural Information Processing Systems 32*, p. 14123–14133, Curran Associates, Inc., 2019.

[10] L. Wehbe, B. Murphy, P. Talukdar, A. Fyshe, A. Ramdas, and T. Mitchell, “Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses,” *PLoS ONE*, vol. 9, p. e112575, Nov. 2014.

[11] M. Toneva and L. Wehbe, "Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)," in *NeurIPS*, 2019.

[12] J. Bingel, M. Barrett, and A. Søgaard, "Extracting token-level signals of syntactic processing from fMRI - with an application to PoS induction," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 747–755, Association for Computational Linguistics, Aug. 2016.

[13] A. Søgaard, "Evaluating word embeddings with fMRI and eye-tracking," in *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, (Berlin, Germany), pp. 116–121, Association for Computational Linguistics, Aug. 2016.

[14] N. Hollenstein, A. de la Torre, N. Langer, and C. Zhang, "CogniVal: A Framework for Cognitive Word Embedding Evaluation," *arXiv:1909.09001 [cs]*, Oct. 2019. arXiv: 1909.09001.

[15] S. Jain and A. G. Huth, "Incorporating Context into Language Encoding Models for fMRI," p. 6628–6637, 2018.

[16] F. Pereira, B. Lou, B. Pritchett, S. Ritter, S. J. Gershman, N. Kanwisher, M. Botvinick, and E. Fedorenko, "Toward a universal decoder of linguistic meaning from brain activation," Mar. 2018.

[17] M. Stephen, X. Caiming, B. James, and R. Socher

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[19] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 2020. https://d2l.ai.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[21] C. Sammut and G. I. Webb, eds., *Mean Squared Error*, pp. 653–653. Boston, MA: Springer US, 2010.

[22] C. Sammut and G. I. Webb, eds., *Mean Absolute Error*, pp. 652–652. Boston, MA: Springer US, 2010.

[23] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[24] Z. S. Harris, “Distributional Structure,” *WORD*, vol. 10, pp. 146–162, Aug. 1954. Publisher: Routledge _eprint: https://doi.org/10.1080/00437956.1954.11659520.

[25] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[26] S. Hochreiter and J. Schmidhuber, “LSTM CAN SOLVE HARD LO G TIME LAG PROBLEMS,” p. 8.

[27] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[28] A. Graves, “Generating Sequences With Recurrent Neural Networks,” *arXiv:1308.0850 [cs]*, June 2014. arXiv: 1308.0850.

[29] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” *arXiv:1409.0473 [cs, stat]*, May 2016. arXiv: 1409.0473.

[30] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” *arXiv:1409.3215 [cs]*, Dec. 2014. arXiv: 1409.3215.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep Contextualized Word Representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[32] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training,” p. 12.

[33] A. M. Dai and Q. V. Le, “Semi-supervised Sequence Learning,” *arXiv:1511.01432 [cs]*, Nov. 2015. arXiv: 1511.01432.

[34] J. Howard and S. Ruder, “Fine-tuned Language Models for Text Classification,” *ArXiv*, 2018.

[35] “Transformers.” `https://huggingface.co/transformers/index.html`.

[36] “BERT.” `https://huggingface.co/transformers/model_doc/model_doc/bert.html`.

[37] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *arXiv:1609.08144 [cs]*, Oct. 2016. arXiv: 1609.08144.

[38] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 328–339, Association for Computational Linguistics, July 2018.

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *arXiv:1409.0575 [cs]*, Jan. 2015. arXiv: 1409.0575.

[40] J.-M. Schoffelen, R. Oostenveld, N. H. L. Lam, J. Uddén, A. Hultén, and P. Hagoort, "A 204-subject multimodal neuroimaging dataset to study language processing," *Scientific Data*, 2019.

[41] N. Chang, J. Pyles, A. Marcus, A. Gupta, M. Tarr, and E. Aminoff, "BOLD5000," 1 2019. https://kilthub.cmu.edu/articles/BOLD5000/6459449.

[42] S. Bhattasali, J. Brennan, W.-M. Luh, B. Franzluebbers, and J. Hale, "The alice datasets: fMRI & EEG observations of natural language comprehension," in *Proceedings of The 12th Language Resources and Evaluation Conference*, (Marseille, France), pp. 120–125, European Language Resources Association, May 2020.

[43] M. Hanke, F. J. Baumgartner, P. Ibe, F. R. Kaule, S. Pollmann, O. Speck, W. Zinke, and J. Stadler, "A high-resolution 7-tesla fmri dataset from complex natural stimulation with an audio movie," *Scientific Data*, vol. 1, p. 140003, May 2014.

[44] X. Liu, Z. Zhen, A. Yang, H. Bai, and J. Liu, "A manually denoised audio-visual movie watching fmri dataset for the studyforrest project," *Scientific Data*, vol. 6, p. 295, Nov 2019.

[45] M. Hanke, N. Adelhöfer, D. Kottke, V. Iacovella, A. Sengupta, F. R. Kaule, R. Nigbur, A. Q. Waite, F. Baumgartner, and J. Stadler, "A studyforrest ex-

tension, simultaneous fmri and eye gaze recordings during prolonged natural stimulation," *Scientific Data*, vol. 3, p. 160092, Oct 2016.

[46] A. Sengupta, F. R. Kaule, J. S. Guntupalli, M. B. Hoffmann, C. Häusler, J. Stadler, and M. Hanke, "A studyforrest extension, retinotopic mapping and localization of higher visual areas," *Scientific Data*, vol. 3, p. 160093, Oct 2016.

[47] M. Dehghani, R. Boghrati, K. Man, J. Hoover, S. I. Gimbel, A. Vaswani, J. D. Zevin, M. H. Immordino-Yang, A. S. Gordon, A. Damasio, and J. T. Kaplan, "Decoding the neural representation of story meanings across languages: Decoding the neural representation," vol. 38, no. 12, pp. 6096–6106, 2017.

[48] S. Stehwien, L. Henke, J. Hale, J. Brennan, and L. Meyer, "The little prince in 26 languages: Towards a multilingual neuro-cognitive corpus," p. 7, 2020.

[49] Z. Hu, H. Yang, Y. Yang, S. Nishida, C. Madden-Lombardi, J. Ventre-Dominey, P. F. Dominey, and K. Ogawa, "Common neural system for sentence and picture comprehension across languages: A chinese–japanese bilingual study," vol. 13, p. 380, 2019.

[50] J. Brennan, E. Stabler, S. Wagenen, W.-M. Luh, and J. Hale, "Abstract linguistic structure correlates with temporal activity during naturalistic comprehension," *Brain and Language*, vol. 157-158, pp. 81–94, 06 2016.

[51] N. Kriegeskorte, R. Goebel, and P. Bandettini, "Information-based functional brain mapping," 3 2006.

[52] L. Beinborn, S. Abnar, and R. Choenni, "Robust evaluation of language-brain encoding experiments," *CoRR*, vol. abs/1904.02547, 2019.

[53] J. Gauthier and A. Ivanova, "Does the brain represent words? an evaluation of brain decoding studies of language understanding," Sept. 2018.

[54] E. Artemova, A. Bakarov, A. Artemov, E. Burnaev, and M. Sharaev, "Data-driven models and computational tools for neurolinguistics: a language technology perspective," 2020.

[55] J. V. Haxby, J. S. Guntupalli, A. C. Connolly, Y. O. Halchenko, B. R. Conroy, M. I. Gobbini, M. Hanke, and P. J. Ramadge, "A common, high-dimensional model of the representational space in human ventral temporal cortex," *Neuron*, vol. 72, pp. 404–416, Oct 2011. PMC3201764[pmcid].

[56] K. Vodrahalli, P.-H. Chen, Y. Liang, C. Baldassano, J. Chen, E. Yong, C. Honey, U. Hasson, P. Ramadge, K. A. Norman, and S. Arora, "Mapping

between fMRI responses to movies and their natural language annotations," *NeuroImage*, vol. 180, pp. 223–231, Oct. 2018.

[57] J. S. Turek, T. L. Willke, P.-H. Chen, and P. J. Ramadge, "A semi-supervised method for multi-subject FMRI functional alignment," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Mar. 2017.

[58] M. Yousefnezhad and D. Zhang, "Local discriminant hyperalignment for multi-subject fmri data alignment," 2016.

[59] M. Yousefnezhad, A. Selvitella, L. Han, and D. Zhang, "Supervised hyperalignment for multi-subject fmri data alignment," *IEEE Transactions on Cognitive and Developmental Systems*, p. 1–1, 2020.

[60] M. Yousefnezhad and D. Zhang, "Deep hyperalignment," 2017.

[61] B. Devereux, C. Kelly, and A. Korhonen, "Using fmri activation to conceptual stimuli to evaluate methods for extracting conceptual representations from corpora," in *Proceedings of the NAACL HLT 2010 First Workshop on Computational Neurolinguistics*, CN '10, (USA), p. 70–78, Association for Computational Linguistics, 2010.

[62] A. Jelodar, M. Alizadeh, and S. Khadivi, "Wordnet based features for predicting brain activity associated with meanings of nouns," pp. 18–26, 01 2010.

[63] J. António Rodrigues, R. Branco, J. Silva, C. Saedi, and A. Branco, "Predicting Brain Activation with WordNet Embeddings," in *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, (Melbourne), pp. 1–5, Association for Computational Linguistics, July 2018.

[64] S. Abnar, R. Ahmed, M. Mijnheer, and W. Zuidema, "Experiential, Distributional and Dependency-based Word Embeddings have Complementary Roles in Decoding Brain Activity," in *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, (Salt Lake City, Utah), pp. 57–66, Association for Computational Linguistics, Jan. 2018.

[65] A. J. Anderson, J. R. Binder, L. Fernandino, C. J. Humphries, L. L. Conant, M. Aguilar, X. Wang, D. Doko, and R. D. S. Raizada, "Predicting Neural Activity Patterns Associated with Sentences Using a Neurobiologically Motivated Model of Semantic Representation," *Cerebral Cortex*, vol. 27, pp. 4379–4395, 08 2016.

[66] L. Bulat, S. Clark, and E. Shutova, “Speaking, Seeing, Understanding: Correlating semantic models with conceptual representation in the brain,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 1081–1091, Association for Computational Linguistics, Sept. 2017.

[67] A. G. Huth, W. A. de Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant, “Natural speech reveals the semantic maps that tile human cerebral cortex,” *Nature*, vol. 532, pp. 453–458, Apr. 2016.

[68] L. Cao and Y. Zhang, *Investigating Lexical and Semantic Cognition by Using Neural Network to Encode and Decode Brain Imaging*, pp. 84–100. 11 2019.

[69] S. Abnar, L. Beinborn, R. Choenni, and W. Zuidema, “Blackbox meets blackbox: Representational Similarity and Stability Analysis of Neural Language Models and Brains,” *arXiv:1906.01539 [cs, q-bio]*, June 2019. arXiv: 1906.01539.

[70] P. Qian, X. Qiu, and X. Huang, “Bridging LSTM Architecture and the Neural Dynamics during Reading,” *arXiv:1604.06635 [cs]*, Apr. 2016. arXiv: 1604.06635.

[71] S. Merity, “Single headed attention rnn: Stop thinking with your head,” 2019.

[72] A. Warstadt, A. Singh, and S. R. Bowman, “Neural Network Acceptability Judgments,” *arXiv:1805.12471 [cs]*, Oct. 2019. arXiv: 1805.12471.

[73] B. W. Matthews, “Comparison of the predicted and observed secondary structure of T4 phage lysozyme,” *Biochimica Et Biophysica Acta*, vol. 405, pp. 442–451, Oct. 1975.

[74] W. B. Dolan and C. Brockett, “Automatically Constructing a Corpus of Sentential Paraphrases,” in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[75] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, “SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (Vancouver, Canada), pp. 1–14, Association for Computational Linguistics, Aug. 2017.

[76] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, Nov. 2016.

[77] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," 2019. In the Proceedings of ICLR.

[78] A. S. White, P. Rastogi, K. Duh, and B. Van Durme, "Inference is Everything: Recasting Semantic Resources into a Unified Evaluation Framework," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Taipei, Taiwan), pp. 996–1005, Asian Federation of Natural Language Processing, Nov. 2017.

[79] D. Demszky, K. Guu, and P. Liang, "Transforming Question Answering Datasets Into Natural Language Inference Datasets," *arXiv:1809.02922 [cs]*, Sept. 2018. arXiv: 1809.02922.

[80] I. Dagan, O. Glickman, and B. Magnini, "The PASCAL Recognising Textual Entailment Challenge," in *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment* (J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d'Alché Buc, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 177–190, Springer, 2006.

[81] R. Bar Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor, "The second PASCAL recognising textual entailment challenge," 2006.

[82] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, "The third PASCAL recognizing textual entailment challenge," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, (USA), pp. 1–9, Association for Computational Linguistics, June 2007.

[83] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini, "The Fifth PASCAL Recognizing Textual Entailment Challenge," in *In Proc Text Analysis Conference (TAC'09*, 2009.

[84] H. Levesque, E. Davis, and L. Morgenstern, "The Winograd Schema Challenge," p. 10.

[85] W. Rahman, M. K. Hasan, S. Lee, A. Zadeh, C. Mao, L.-P. Morency, and E. Hoque, "Integrating Multimodal Information in Large Pretrained Transformers," *arXiv:1908.05787 [cs, stat]*, Nov. 2020. arXiv: 1908.05787.

[86] R. Marvin and T. Linzen, "Targeted syntactic evaluation of language models," 2018.