Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Σημάτων, Ελέγχου
και Ρομποτικής

Cognitive and Cross-Topic Methods for Natural Language Representations

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΝΙΚΟΣ ΑΘΑΝΑΣΙΟΥ

Επιβλέπων : Αλέξανδρος Ποταμιάνος
Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Σημάτων, Ελέγχου
και Ρομποτικής

Cognitive and Cross-Topic Methods for Natural Language Representations

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΝΙΚΟΣ ΑΘΑΝΑΣΙΟΥ

Επιβλέπων : Αλέξανδρος Ποταμιάνος
Αναπληρωτής Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Ιουλίου 2019.

Alexandros Potamianos Associate Professor NTUA	Konstantinos Tzafestas Associate Professor NTUA	Andreas-Georgios Stafylopatis Professor NTUA

Αθήνα, Ιούλιος 2019

Νίκος Αθανασίου

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

Dedicated to my grandmother Marianthi who planted the ultimate principles of human generosity, patience, kindness and sympathy in me, by her presence alone. I hope to preserve them untouched no matter how much they are appreciated.

Acknowledgements

Throughout this thesis, I had the chance to be supervised by Prof. Alexandros Potamianos. I improved myself by listening to his advice and was able to publish my work. I want to thank him for his guidance and helpful comments. I would also like to express my kind regards to Elias Iosif for assisting me in the first part of my work. I would also like to thank PhD candidates Christos Baziotis and Giorgos Paraskevopoulos for the fruitful discussions which helped me decompress in crucial moments. Special thanks to my colleagues in NTUA-SLP lab for assisting me and for making our long hours of work, especially before deadlines.

By presenting this thesis, I am officially finishing with an important part of my life and starting a new one. I have to thank and show my gratitude to my parents Maria Ioannidou and Ioannis Athanasiou, for supporting me and helping me in all my decisions. I would also like to thank them both for believing in me and for motivating me to pursue my goals, as much as they could.

Moreover, I want to thank my high school physicist Ilias Kastrinellis for his patience and his intuitiveness which brightened my mind. I would also like to thank my mathematician Panagiotis Mitarellis for his endless thirst for knowledge that he altruistically transmitted to me and for entrenching my sense of persistence for the best.

Finally, I would like to thank my friends, T., B., S., P. and C. who some throughout, some in part of this 7 year challenge assisted me emotionally, psychologically or physically. I would also like to thank all the people that have been lost throughout all those years —as it commonly happens—, but encouraged me for periods longer than they disappointed me.

Nikos Athanasiou,

Athens, July 5, 2019

Abstract

In this work [1]we investigate Natural Language Representations by two different points of view cognitive neuroscience and topic modelling. For the evaluation of each approach, we use multiple datasets and experimental setups which follow literature's guidelines. Moreover, we evaluate our work both quantitatively and qualitatively providing useful insights and visualizations in order to make our results interpretable.

First, from the angle of cognitive neuroscience we explore how brain representations can help us improve current corpus-based language representations. Neural activation models that have been proposed in the literature use a set of example words for which fMRI measurements are available in order to find a mapping between word semantics and localized neural activations. Successful mappings let us expand to the full lexicon of concrete nouns using the assumption that similarity of meaning implies similar neural activation patterns. In this paper, we propose a computational model that estimates semantic similarity in the neural activation space and investigates the relative performance of this model for various natural language processing tasks. Despite the simplicity of the proposed model and the very small number of example words used to bootstrap it, the neural activation semantic model performs surprisingly well compared to state-of-the-art word embeddings. Specifically, the neural activation semantic model performs better than the state-of-the-art for the task of semantic similarity estimation between very similar or very dissimilar words, while performing well on other tasks such as entailment and word categorization. These are strong indications that neural activation semantic models can not only shed some light into human cognition but also contribute to computation models for certain tasks.

In the second part, we investigate how topic modelling can help us produce multi-prototype word embeddings and compare their performance with single-prototype models. In traditional Distributional Semantic Models (DSMs) the multiple senses of a polysemous word are conflated into a single vector space representation. In this work, we propose a DSM that learns multiple distributional representations of a word based on different topics. First, a separate DSM is trained for each topic and then each of the topic-based DSMs is aligned to a common vector space. Our unsupervised mapping approach is motivated by the hypothesis that words preserving their relative distances in different topic semantic sub-spaces constitute robust *semantic anchors* that define the mappings between them. Aligned cross-topic representations achieve state-of-the-art results for the task of contextual word similarity. Furthermore, evaluation on NLP downstream tasks shows that multiple topic-based embeddings outperform single-prototype models.

Key words

Computational Neuroscience, Deep Learning, Machine Learning, Word Embeddings, Multiple Word Embeddings, Natural Language Processing,Topic Modelling, Cognition & Natural Language

[1] Papers: [1], [2] have been conducted under the development of this thesis.

Εκτεταμένη Περίληψη

Στην παρούσα εργασία [2] επιχειρούμε να εξετάσουμε Αναπαραστάσεις Φυσικής Γλώωσσας από δύο διαφορετικές οπτικές γωνίες, αυτή της Γνωστικής Νευροεπιστήμης και αυτή της Θεματικής Μοντελοποίησης. Για την αξιολόγηση των αποτελεσμάτων μας, χρησιμοποιήσαμε πολλάπλα διαφορετικές βάσεις δεδομένων και πειραματικές διαδικασίες, σύμφωνα με την εκάστοτε βιβλιογραφία. Επιπλέον, αξιολογήσαμε τα αποτελέσματα της έρευνας μας τόσο ποιοτικά όσο και ποσοτικά our παρέχοντας χρήσιμες οπτικοποιήσεις και αποτελέσματα που δίνουν στον αναγνώστη βαθύτερη κατανόηση πίσω από τις ιδέες που χρησιμοποίηθηκαν και τα συμπεράσματα πυ εξάχθηκαν.

Αναπαραστάσεις Φυσικής Γλώσσας

Η μοντελοποίηση της Φυσικής Γλώσσας είναι διαχρονικά η διαδικασία πρόβλεψης της επόμενης λέξης σε ένα κομμάτι κειμένου δεδομένων των προηγούμενων λέξεων. Είναι ένα από τα πιο απλά προβλήματα του τομέα της Επεξεργασίας Φυσικής Γλώσσας με απτές πρακτικές εφαρμογές όπως η έξυπνη συμπλήρωση πληκτρολογούμενου κειμένου, πρόταση απαντήσεων σε μυνήματα ηλεκτρονικού ταχυδρομείου [3], διόρθωση συντακτικών λαθών κ.α. Επομένως, όπως είναι λογικό έχει ερευνηθεί διαχρονικά σε μεγάλο βαθμό. Οι κλασσικές προσεγγίσεις βασίζονται στη μελέτη των ν-γράμμων(n-grams). Το n-gram είναι η ακολουθία ν κομματιών κειμένου όπως συλλαβές, γράμματα ή λέξεις. Με τη χρήση Μαρκοβιανών Αλυσίδων, ο χωρισμός του κειμένου σε n-grams βοήθησε ώστε με διαδικασίες μάθησης να εξαχθούν στοιχεία συντακτικής ή σημασιολογικής μορφολογίας του κειμένου [4]. Πρόσφατα, τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης αντικαταστάθηκαν από νευρωνικά δίκτυα με ανατροφοδότηση (RNNs) [5].

Πολυσημία

Από υπολογιστικής πλευράς, οι αναπαραστάσεις των λέξεων στοχεύουν στο να διευκολύνουν τους υπολογιστές να εντοπίσουν νοηματικά στοιχεία της γλώσσας, αλλά και να κωδικοποιήσουν αυτή την πληροφορία με ένα φορμαλιστικό τρόπο που να τους επιτρέπει να την χρησιμοποιήσουν προς επίλυση σχετικών προβλημάτων, όπου η κατανόηση της γλώσσας παίζει σημαντικό ρόλο. Η σημαντικότητα της κατανόησης της σημασιολογίας διαφόρων λεξικολογικών μονάδων παίζει καθοριστικό ρόλο στην κατανόηση και την επίκτηση της γλώσσας. Αυτό συμβαίνει γιατί οι λέξεις, οι συλλαβές και τα υπόλοιπες γλωσσολογικές μονάδες αποτελούν τα βασικά συστατικά της Φυσικής Γλώσσας. Η πολυσημία είναι ένα γλωσσικό φαινόμενο που συναντάται συνήθως σε μονάδες που παίζουν σημαντικό ρόλο στην κατανόηση της γλώσσας, τις λέξεις. Συγκεκριμένα, η πολυσημία μιας λέξης (π.χ. *πλευρά*) θα μπορούσε να επηρεάσει δραστικά την αναπαράσταση της, ανάλογα με το γλωσσικό πλαίσιο στο οποίο την συναντάμε, δηλαδή τις λέξεις με τις οποίες συνυπάρχει σε ένα συγκεκριμένο κομμάτι κειμένου.

Φυσική Γλώσσα & Εγκέφαλος

Επιπλέον, προσφάτως, έρευνες πάνω σε γνωστικά πειράματα τα οποία διερευνούν την αλληλεπίδραση φυσικής γλώσσας και εγκεφάλου, φαίνεται να υποδηλώνουν ότι η κατανόηση της σημασιολογίας μπορεί να ενισχυθεί από τις θεμελιώδεις γνωστικές σχέσεις μεταξύ των λέξεων [6]. Επίσης, πρόσφατες εργασίες από υπολογιστικούς νευροεπιστήμονες και γλωσσολόγους, έχουν δείξει ότι είναι εφικτή η χαρτογράφηση μεταξύ γνωστικού και σημασιολογικού χώρου [7, 8, 9, 10]. Μια απλή συνέπεια αυτής της παρατήρησης είναι ότι η έννοια μιας λέξης εξαρτάται σε μεγάλο βαθμό από τις σημασιολογικές σχέσεις που μοιράζεται με άλλες λέξεις. Όπως έχουν πρόσφατα αναφερθεί από πολλούς ερευνητές, οι παραδοσιακές προσεγγίσεις μηχανικής μάθησης έφεραν ταχείες και σημαντικές βελτιώσεις σε διαφορετικά καθήκοντα επεξεργασίας φυσικών γλωσσών, αλλά ο τομέας ενδέχεται να αντιμετωπίσει από τώρα και στο εξής δύσκολα προβλήματα (π.χ. σχεδιασμός λόγου, επιχειρημα-

[2] Οι δημοσιεύσεις: [1], [2] εκπονηθηκαν κατά τη διάρκεια της παρούσας διπλωματικής εργασίας.

τολογική ανάλυση κ.λπ.) που θα επωφεληθούν από την καλύτερη κατανόηση των διαδικασιών που εμπλέκονται στον εγκέφαλο. Επιπλέον, οι ερευνητές ενδιαφέρονται και πάλι να αξιολογήσουν τη συνάφεια των μοντέλων τους σύμφωνα με μια γνωστική διάσταση. Τέλος, η γνωστική επιστήμη ωφελεί και μερικές φορές παίρνει έμπνευση από υπολογιστικά μοντέλα.

0.1 Γνωστικά Σημασιολογικά Μοντέλα

Σε αυτό το σημείο, θα παρουσιαστεί συνοπτικά η δουλειά μας στην δημοσίευση [1].

Η μελέτη της σημασιολογίας στον εγκέφαλο είναι ένας κλάδος της ψυχογλωσσολογίας που ενσωματώνει την κατανόηση της σημασιολογίας και των νευρολογικών δομών που εμπλέκονται. Προσπαθεί να απαντήσει στο αναπάντητο ερώτημα του πώς τα αντικείμενα και οι έννοιες αντιπροσωπεύονται και επεξεργάζονται στον ανθρώπινο εγκέφαλο [11]. Έχουν διεξαχθεί διάφορες μελέτες για να διερευνηθούν οι μηχανισμοί κωδικοποίησης και αποκωδικοποίησης του εγκεφάλου όταν υπάρχει ένα ερέθισμα, όπως αναλύεται στη συνέχεια. Για οπτικά ερεθίσματα, μελέτες έχουν δείξει ότι είναι εφικτό να γίνεται διάκριση και αναδημιουργία εικόνων με τη χρήση προτύπων νευρικής δραστηριότητας, κυρίως στο οπτικό φλοιό [12, 13, 14, 15, 16], το τμήμα του εγκεφάλου που είναι υπεύθυνο για την οπτική επεξεργασία πληροφοριών. Άλλες μελέτες έχουν καταδείξει τη σχέση μεταξύ της γνωστικής αντίληψης και της ομιλίας [17, 18]. Οι λεξικές σημασιολογίες βασίζονται στην υπόθεση ότι παρόμοιες λέξεις εμφανίζονται σε παρόμοια περιβάλλοντα [19]. Με βάση αυτή την υπόθεση, έχουν προταθεί δύο διαφορετικές προσεγγίσεις για τη δημιουργία σημασιολογικών μοντέλων. Η πρώτη προσέγγιση είναι η κωδικοποίηση της σημασιολογίας μιας λέξης, εφαρμόζοντας τη μείωση των διαστάσεων της μήτρας συνύπαρξης των λέξεων που υπολογίστηκε με τη χρήση μεγάλων κειμένων [20, 21]. Η δεύτερη προσέγγιση αντικαθιστά αυτή την «μέτρηση» με μοντέλα [22] με βάση τα νευρωνικά δίκτυα [23, 24, 25, 26, 27].

Αυτή η διατριβή επιχειρεί να απαντήσει εάν οι αναπαραστάσεις της σημασιολογίας με βάση τον εγκέφαλο είναι συμπληρωματικές σε σχέση με τις λεξικολογικές αναπαραστάσεις. Οι απαντήσεις σε μια τέτοια ερώτηση φέρνουν νέες γνώσεις για το ρόλο που μπορούν να διαδραματίσουν τα δεδομένα του εγκεφάλου στη μελέτη της σημασιολογίας.

Στο πλαίσιο του εμπλουτισμού αυτών των λεξικών σημασιολογικών μοντέλων με γνωσιακές πληροφορίες, καθώς και την ανακάλυψη της γνωσιακής αναπαράστασης της σημασιολογικής σημασιολογίας, αρκετές μελέτες έχουν επιχειρήσει να εξετάσουν τη χαρτογράφηση μεταξύ της σημασιολογικής αναπαράστασης υπολογιστικών και γνωστικών μοντέλων. Σε προηγούμενη εργασία, έχει αποδειχθεί ότι η σημασιολογία των λέξεων σχετίζεται με δυναμικά ενεργοποίησης σε περιοχές του εγκεφάλου και ότι είναι δυνατή η αποκωδικοποίηση μεταξύ νευρωνικών ενεργοποιήσεων και σημασιολογικού περιεχομένου των λέξεων [7, 28, 29, 30, 31]. Επιπλέον, οι νευρικές ενεργοποιήσεις δείχνουν ότι έχουν προγνωστική ισχύ σε σχέση με τη σημασιολογία μιας λέξης [7, 8] και πρότασης [32, 33]. Οι υπολογιστικές μελέτες που στοχεύουν στη διερεύνηση της επίδρασης των νευρωνικών ενεργοποιήσεων σε εκφράσεις λέξεων έχουν δείξει ότι με την έμμεση ενσωμάτωση νευρικών ενεργοποιήσεων κατά την εκπαίδευση των λεξικών σημασιολογικών μοντέλων μπορεί να βελτιώσει την ικανότητα γενίκευσης τους παρά την περιορισμένη ποσότητα εγκεφαλικών δεδομένων νευρικής ενεργοποίησης που χρησιμοποιούνται [34, 35].

Αυτές οι δημοσιεύσεις καταδεικνύουν την ισχυρή ύπαρξη σχέσης μεταξύ υπολογιστικών σημασιολογικών μοντέλων και νευρωνικών αναπαραστάσεων. Ωστόσο, παραμένει να δούμε πώς οι γνωσιακές σημασιολογικές αναπαραστάσεις μπορούν να συμβάλουν στη βελτίωση της απόδοσης των υπολογιστικών σημασιολογικών μοντέλων, ειδικά για περίπλοκα προβλήματα ταξινόμησης και αναγνώρισης.

Με βάση τις προαναφερθείσες μελέτες που δείχνουν συσχετισμό μεταξύ τοπικών νευρονικών δραστηριοτήτων και λεξικολογικών αναπαραστάσεων, προτείνουμε ένα υπολογιστικό μοντέλο για σημασιολογική ομοιότητα που χρησιμοποιεί προβλεπόμενες νευρωνικές ενεργοποιήσεις που αποκτήθηκαν από ένα μικρό σύνολο ουσιαστικών. Το προτεινόμενο μοντέλο εφαρμόζεται σε διάφορα προβλήματα επεξεργασίας φυσικής γλώσσας. Το μοντέλο πρόβλεψης νευρωνικών ενεργοποίησεων

που χρησιμοποιούμε ώστε να εξάγουμε νεωρωνικές ενεργοποίησεις για λέξεις εκτός αυτών που υπάρχουν στα εγκεφαλικά δεδομένα που έχουμε στη διάθεση μας είναι αυτό που προτείνεται στο [7]. Στη λίστα με τα πειράματά μας, πρώτα συγκρίνουμε την απόδοση του προτεινόμενου μοντέλου για το πρόβλημα της σημασιολογικής ομοιότητας και δείχνουμε ότι για ορισμένα ζεύγη λέξεων αποδίδει αποτελέσματα καλύτερα από λεξικολογικές αναπαράστασεις. Στη συνέχεια, αξιολογούμε την απόδοση του μοντέλου για ταξινόμηση λέξεων, ταξινόμηση λέξεων ανάλογα με τη συσχέτιση του ς με τις αισθήσεις και την συνεπαγωγή κειμένου. Ο συνδυασμός των νευρωνικών και των παραδοσιακών λεξικολογικών αναπαραστάσεων ξεπερνά—σε κάποιες περιπτώσεις—τα καλύτερα μέχρι τώρα αποτελέσματα σύμφωνα με τη βιβλιογραφία.

0.1.1 Μοντέλο Σημασιολογικών Νευρωνικών Αναπαραστάσεων Λέξεων

0.1.2 Σημασιολογική ομοιότητα

Με βάση την υπόθεση ότι παρόμοιες λέξεις έχουν παρόμοιες νευρικές ενεργοποιήσεις, προτείνουμε ένα μοντέλο για την εκτίμηση των ομοιόμορφων λέξεων που βασίζονται σε νευρικές ενεργοποιήσεις που προβλέπονται με τη χρήση του προβλέπτη νευρωνικών ενεργοποιήσεων που υλοποιήθηκε [7]. Εδώ να σημειωθεί ότι voxel ονομάζουμε το 3d εικονοστοιχείο το οποίο εμπεριέχει μια τιμή ενεργοποίησης για μια στοιχειώδη εγκεφαλική περιοχή. Στην παρούσα δημοσίευση χρησιμοποιούμε τα 500 voxel(V) που μας παρέχουν την χρησιμότερη πληροφορία. Αξιολογήσαμε διάφορες μετρικές για τον υπολογισμό της σημασιολογικής ομοιότητας από τις νευρωνικές ενεργοποιήσεις. Παρουσιάζουμε μια μετρική, που έχει διαμορφωθεί ως η σταθμισμένη τετραγωνική απόσταση, και έδωσε τα καλύτερα αποτελέσματα:

$$S(w_1, w_2) = \sum_{v=1}^{V} b_v \ (y_v(w_1) - y_v(w_2))^2, \tag{0.1}$$

όπου $S(w_1, w_2)$ είναι η σημασιολογική ομοιότητα μεταξύ των λέξεων w_1 και w_2, Το V αντιπροσωπεύει τον αριθμό των voxels που χρησιμοποιούνται στην προβλεπόμενη εγκεφαλική αναπαράσταση, $y_v(w)$ είναι η ενεργοποίηση ενός voxel για τη λέξη w, και b_v είναι ένα βάρος της συμβολής ενός συγκεκριμένου voxel στη μετρική ομοιότητας, το οποίο το μαθαίνουμε μέσω εκπαίδευσης.

0.1.3 Πρόβλημα Ταξινόμησης

Η απόδοση της σημασιολογικής ομοιότητας που υπολογίστηκαν από την εξίσωση 0.1 αξιολογήθηκαν επίσης στο πρόβλημα δημιουργίας ταξονομίας στο σύνολο δεδομένων ESSLLI [36]. Η δημιουργία ταξονομίας γίνεται χρησιμοποιώντας τους φορείς νευρικής ενεργοποίησης $\vec{y}(w)$ που υπολογίζονται από την Εξίσωση 4.2 και τους φορείς συντελεστή $\vec{b}$ που ορίζονται στην Εξίσωση 0.1 που εκπαιδεύεται χρησιμοποιώντας γραμμική παλινδρόμηση σε όλο το σύνολο δεδομένων MEN. Στη συνέχεια, ο πίνακας ομοιότητας $S(w_i, w_j)$ υπολογίζεται για όλα τα ζεύγη στο σύνολο δεδομένων χρησιμοποιώντας την εξίσωση 0.1 και στη συνέχεια εφαρμόζεται ο αλγόριθμος φασματικής ομαδοποίησης που προτείνεται στο [37] για να ληφθούν οι λεξικές κλάσεις. Σε αυτή την εργασία, η νεωρωνική συγχώνευση αναφέρεται στην πρώιμη συνένωση των λεξικολογικών και νευρωνικών αναπαραστάσεων. Χρησιμοποιήσαμε τη μετρική της καθαρότητας των κλάσεων για την αξιολόγηση της ποιότητας των αυτόματα δημιουργημένων κλάσεων[38].

0.1.4 Ταξινόμηση Λέξεων ανάλογα με τη συσχέτιση τους με τις Αισθήσεις

Για το παρόν πρόβλημα χρησιμοποιούμε το σύνολο δεδομένων Sensicon. Εξ ορισμού όλα τα ουσιαστικά στο Sensicon συνδέονται με ένα πραγματικό αισθητήριο ερέθισμα. Η ταξινόμηση πραγματοποιείται όπως περιγράφεται στο υποκεφάλαιο 0.1.3, δηλ. Η μήτρα ομοιότητας στην Εξίσωση 0.1 υπολογίζεται με χρήση του φορέα βάρους $\vec{b}$ που εκπαιδεύεται στο σύνολο δεδομένων MEN και στη

συνέχεια εφαρμόζεται η φασματική ομαδοποίηση [37] για τις πέντε κατηγορίες αίσθησης. Οι προκύπτουσες ομάδες χρησιμοποιούνται για την ταξινόμηση νοημάτων είτε μεταξύ δύο αισθήσεων, ενός έναντι όλων ή μεταξύ των πέντε αισθήσεων.

0.2 Πειραματική Αξιολόγηση

MEN: Για πρόβλημα σημασιολογικής ομοιότητας, εκπαιδεύουμε και αξιολογούμε το μοντέλο μας στο σύνολο δεδομένων MEN [39] το οποίο αποτελείται από 3000 ζεύγη λέξεων (2000 για σύνολο εκπαίδευσης και 1000 ζευγάρια για σύνολο αξιολόγησης). Κάθε ζεύγος λέξεων συνδέεται με μια βαθμολογία ομοιότητας. Δημιουργήσαμε επίσης 2 υποσύνολα MEN από 39 πολύ παρόμοια και 79 εντελώς διαφορετικά ζεύγη λέξεων χρησιμοποιώντας μια τεχνική κατωφλίου, όπου τα ζευγάρια με βαθμολογία ομοιότητας πάνω από 0,85 και κάτω από 0,1 ανήκουν στο πρώτο και το δεύτερο υποσύνολο αντίστοιχα.

ESSLLI: Για την εργασία δημιουργίας ταξονομίας, αξιολογούμε το μοντέλο μας στο σύνολο δεδομένων ESSLLI [36]. Αποτελείται από μια ιεραρχία τριών επιπέδων (2-3-6 τάξεις). Το χαμηλότερο επίπεδο ιεραρχίας περιλαμβάνει 6 τάξεις ουσιαστικών, το μεσαίο 3 τάξεις, ενώ η ανώτερη τάξη διακρίνεται μεταξύ ζωντανών όντων και αντικειμένων.

Sensicon: Για ταξινόμηση νοήματος, χρησιμοποιούμε το σύνολο δεδομένων Sensicon [40]. Το Sensicon είναι ένα λεξικό που περιέχει 22684 αγγλικές λέξεις και συνδέει κάθε λέξη με 5 αριθμητικές βαθμολογίες. Οι βαθμολογίες αντιστοιχούν στη συνάφεια της λέξης με κάθε μία από τις 5 αισθήσεις, δηλαδή την όραση, την ακοή, τη γεύση, την οσμή και την αφή. Για να χρησιμοποιήσουμε αυτές τις βαθμολογίες για την εργασία αίσθησης ταξινόμησης, επιλέξαμε ουσιαστικά που έχουν μη μηδενικά αποτελέσματα με μία μόνο έννοια και έχουν ως αποτέλεσμα 1011 λέξεις.

SNLI dataset: Για το πρόβλημα της συνεπαγωγής χρησιμοποιήσαμε το σύνολο δεδομένων Stanford Natural Language Inference (SNLI) [41] το οποίο περιέχει περίπου 570 χιλιάδες ζεύγη προτάσεων με τρεις ετικέτες: συνεπαγωγή, αντίφαση και ουδέτερο. Η προεπεξεργασία σου για την παρουσία αρκετών ουσιαστικών, είχε ως αποτέλεσμα τον σχηματισμό 30.498 και 592 δειγμάτων εκπαίδευσης και αξιολόγησης για την περίπτωση τριών τουλάχιστον ουσιαστικών και 171.528 εκπαίδευσης και 3201 δειγμάτων αξιολόγησης για την περίπτωση τουλάχιστον δύο κοινών λέξεων με το MEN.

0.3 Πειραματικά αποτελέσματα

0.3.1 Σημασιολογική ομοιότητα

Για το πρόβλημα της σημασιολογικής ομοιότητας, εφαρμόσαμε την εξίσωση 0.1 για τα ζεύγη λέξεων του συνόλου δεδομένων MEN. Τα $y_v(.)$ της εξίσωσης 0.1 υπολογίστηκαν χρησιμοποιώντας την εξίσωση 4.2. Χρησιμοποιήσαμε το υποσύνολο εκπαίδευσης του MEN για την εκμάθηση των βαρών b της εξίσωσης 0.1 χρησιμοποιώντας γραμμική παλινδρόμηση. Αυτά τα βάρη χρησιμοποιήθηκαν για την εκτίμηση των ομοιοτήτων για το υποσύνολο δοκιμής του MEN. Ο συντελεστής συσχέτισης Spearman ανάμεσα στις βαθμολογίες ανθρώπινης ομοιότητας και τα αποτελέσματα ομοιότητας που υπολογίστηκαν από την εξίσωση 0.1 χρησιμοποιήθηκε ως μετρική αξιολόγησης. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 0.1, όπου συγκρίνουμε την απόδοση του προτεινόμενου μοντέλου σε σχέση με την απόδοση του λεξικολογικού μονέλου w2vec [25] εκπαιδευμένες στο κείμενο GoogleNews.

Συνολικά, το μοντέλο w2vec ξεπερνά το δικό μας μοντέλο επιτυγχάνοντας συσχέτιση 0.76 σε όλα τα ουσιαστικά. Το νευρωνικό σημασιολογικό μοντέλο αυξάνει την απόδοση του καθώς εκμεταλλεύομαστε περισσότερα voxels φθάνοντας στο 0.48 συσχέτιση για τουλάχιστον 150 voxels. Στον πίνακα 0.1, εμφανίζεται επίσης η απόδοση για τρία υποσύνολα του δοκιμαστικού συνόλου MEN, δηλαδή "Τα περισσότερα & Λιγότερο όμοια", "Λιγότερο όμοια" και "Τα περισσότερο όμοια" ουσιαστικά.

Υποσύνολο Ουσιαστικών	Αριθμός Voxel	Νευρωνικό Σημασιολ. μοντέλο	w2vec
All Concrete nouns	50	0.43	0.76
	100	0.47	0.76
	150	0.48	0.76
	200	0.48	**0.76**
Περισσότερο & Λιγότερο όμοιες	50	0.58	0.73
	100	0.82	0.73
	150	0.82	0.73
	200	**0.88**	0.73
Λιγότερο όμοιες	50	0.43	0.43
	100	0.44	0.43
	150	0.47	0.43
	200	**0.63**	0.43
Περισσότερο όμοιες	50	0.28	0.14
	100	0.63	0.14
	150	0.68	0.14
	200	**0.83**	0.14

Πίνακας 0.1: Αποτελέσματα αξιολόγησης στο υποσύνολο συγκεκριμένων ουσιαστικών του δοκιμαστικού συνόλου του MEN και στα περισσότερα και λιγότερο παρόμοια υποσύνολα συγκεκριμένων λέξεων.

Η βελτίωση της απόδοσης γίνεται πιο έντονη καθώς ο αριθμός των voxels αυξάνεται. Η καλύτερη βαθμολογία συσχέτισης που επιτυγχάνεται είναι 0.88 για την περίπτωση των "Περισσότερο & Λιγότερο όμοιων " για 200 voxels, υπερβαίνοντας το μοντέλο w2vec (συσχέτιση 0.73). Ειδικά για την περίπτωση του υποσυνόλου αξιολόγησης "Τα περισσότερ όμοια" παρατηρούμε μια αξιοσημείωτη διαφορά μεταξύ των δύο μοντέλων, δηλ. 0.83 έναντι 0.14.

0.3.2 Προβλήματα Φυσικής Γλώσσας

Στη συνέχεια παρουσιάζουμε την απόδοση του σημασιολογικού μοντέλου νευρικών ενεργοποιήσεων για τη δημιουργία ταξονομίας, την ταξινόμηση των αίσθησεων και τη νοηματική συνεπαγωγή.

Δεδομένα(ESSLI)	Νευρωνικό Σημασιολογικό Μοντέλο	w2vec	Συνδυασμός w2vec & Νευρωνικών αναπαραστάσεων
6 κλάσεις	0.61	0.70	**0.71**
3 κλάσεις	0.77	0.95	**0.95**
2 κλάσεις	0.66	**0.77**	0.72

Πίνακας 0.2: Αποτελέσματα αξιολόγησης για τη δημιουργία ταξινόμησης.

Τα αποτελέσματα της αξιολόγησης παρουσιάζονται στον Πίνακα 0.2. Όλα τα αποτελέσματα που δείχνονται υπολογίζονται σε voxels $V = 250$. Το μοντέλο μας συμπεριφέρεται χειρότερα από ότι το μοντέλο w2vec και στις τρεις τάξεις (6, 3 ή 2), ωστόσο, ο συνδυασμός των δύο επιτυγχάνει τις καλύτερα αποτελέσματα για 6 και 3 τάξεις, σε 0.71 και 0.95 καθαρότητα, αντίστοιχα. Για την εργασία ταξινόμησης των αισθήσεων χρησιμοποιούμε το σύνολο δεδομένων Sensicon για να αξιολογήσουμε την απόδοση του μοντέλου μας.

Τα αποτελέσματα της αξιολόγησης παρουσιάζονται στον Πίνακα 0.3. Τα δυο μοντέλα επιτυγχάνουν παρόμοια αποτελέσματα, με το μοντέλο μας να αποδίδει καλύτερα 0.37 έναντι 0.33 για ταξινομήση και των 5 αισθήσεων. Ο συνδυασμός των δύο μοοντέλων ξεπερνά τα επιμέρους αποτελέσματα

Κλάσεις	Νευρωνικό Σημασιολογικό μοντέλο	w2vec	Συνδυασμός w2vec & Νευρωνικών αναπ.
Όραση, Ακοή	0.55	0.55	**0.57**
Όραση, Αφή	0.68	0.66	**0.69**
Όραση, Γεύση	0.60	0.60	**0.61**
Ακοή, Αφή	0.59	0.58	**0.59**
Ακοή, Γεύση	0.57	0.55	**0.57**
Γεύση, Αφή	0.54	0.54	0.54
Όραση, Υπόλοιπες	0.68	0.68	0.68
Ακοή, Υπόλοιπες	0.74	0.74	0.74
Αφή, Υπόλοιπες	0.81	0.81	0.81
Γεύση, Υπόλοιπες	0.78	0.78	**0.79**
Ακοή, Όραση, Όσφρηση, Αφή, Γεύση	0.37	0.33	**0.39**

Πίνακας 0.3: Αποτελέσματα αξιολόγησης για την ταξινόμηση αισθήσεων των λέξεων.

για την πλειονότητα των ταξινομήσεων ανάμεσα σε δυάδες αισθήσεων και επιτυγχάνει επίσης την καλύτερη επίδοση για την ταξινόμηση ανάμεσα και στις πέντε αισθήσεις 0.39. Αυτά τα αποτελέσματα συμφωνούν επίσης με τη νευροεπιστημονική έρευνα [42, 43, 44, 45, 46, 47]. Παρατηρούμε ότι η κο-

Δεδομένα(SNLI)	Διαστάσεις (GloVe, Νευρωνικές ενεργ.)	GloVe	Συνδυασμος Νευρωνικών & GloVe Αναπραστάσεων
3-κοινά	(300,250)	68.2	**68.7**
2-κοινά	(300,250)	76.6	**77.7**

Πίνακας 0.4: Αποτελέσματα ακρίβειας για το πρόβλημα της προτασιακής συνεπαγωγής.

ρυφαία ακρίβεια επιτυγχάνεται με με το συνδυασμό των 2 μοντέλων αναπαραστάσεων τόσο για τα εξεταζόμενα υποσύνολα ($68.7 \pm 0.9\%$ και $77.7 \pm 0.9\%$). Σημειώστε ότι εδώ επιλέξαμε να συγκρίνουμε τις νευρικές ενεργοποιήσεις με διαφορετικές λεξικολογικές αναπραστάσεις για να επεκτείνουμε την αξιολόγησή μας σε άλλες λεξικολογικές αναπαράστασεις που εξάχθηκαν χρήση με διαφορετικού μοντέλου.

0.4 Πειραματικά Συμπεράσματα

Η ανάλυση της απόδοσης του μοντέλου μας έδειξε μπορεί να διαχωρίσει τα πολύ όμοια και πολύ ανόμοια ουσιαστικά καλύτερα από τα σημερινά μοντέλα λέξεων, ενώ παράλληλα έχει χειρότερη επίδοση συνολικά για το πρόβλημα σημασιολογικής ομοιότητας λέξεων. Αυτή είναι μια ισχυρή ένδειξη ότι η σημασιολογική διακριτότητα των νευρωνικών αναπαραστάσεων έχει διαφορετικό σημασιολογικό περιεχόμενο από αυτή των παραδοσιακών μοντέλων αναπαραστάσεων λέξεων και έτσι οι νευρωνικές ενεργοποιήσεις μπορούν να χρησιμοποιηθούν για την βελτίωση των σημερινών σημασιολογικών παραστάσεων. Τα αποτελέσματα σχετικά με την λεξικολογική ταξονομία, την ταξινόμηση αισθήσεων και το πρόβλημα της συνεπαγωγής πράγματι επιβεβαιώνουν τη σημασιολογική πληροφορία των νευρωνικών αναπαραστάσεων. Συνολικά, οι νευρωνικές ενεργοποιήσεις μπορούν επίσης να χρησιμοποιηθούν σε συνδυασμό με άλλες σημασιολογικές παραστάσεις και βαθιές αρχιτεκτονικές για τη βελτίωση των αποτελεσμάτων σε δύσκολα προβλήματα Φυσικής Γλώσσας, όπως η προτασιακή συνεπαγωγή.

0.5 Διαθεματικές Κατανεμημένες Αναπαραστάσεις

Σε αυτό το σημείο, θα παρουσιαστεί συνοπτικά η δουλειά μας που έγινε σε συνεργασία με πρώην μέλος της ομάδας Ελευθερία Μπριάκου στη δημοσίευση [2].

Στα παραδοσιακά Κατανεμημένα Σημασιολογικά Μοντέλα(DSM) οι πολλαπλές αισθήσεις μιας πολύσημης λέξης αναπαραριστόνται με ένα σημείο εντός διανυσματικού χώρου. Σε αυτή την εργασία, προτείνουμε ένα DSM που μαθαίνει πολλαπλές κατανεμημένες αναπαραστάσεις μιας λέξης που βασίζεται σε διαφορετικά θέματα. Τα τρέχοντα μαθησιακά μοντέλα αναπαράστασης λέξεων κωδικοποιούν τις σημασιολογικές και συντακτικές πληροφορίες των λέξεων υιοθετώντας την υπόθεση της κατανεμημένης έννοιας των λέξεων [19]. Οι αλγόριθμοι εξαγωγής αναπαραστάσεων λέξεων κωδικοποιούν τα συμφραζόμενα των λέξεων σε διανύσματα χαρακτηριστικών (embeddings). Ωστόσο, τέτοια μοντέλα (w2vec, Glove, fasttext) μαθαίνουν αναπαραστάσεις ενός σημείου, οι οποίες δεν μπορούν να καταγράψουν τις ξεχωριστές έννοιες πολύσημων λέξεων (π.χ. *βάρος* ή *κλείνω*), αφού δε λαμβάνουν τα διαφορετικά σημασιολογικά πλαίσια που αυτές μπορεί να βρεθούν εντός ενός κειμένου. Έτσι, η δημιουργία πολλαπλών διανυσμάτων, οι οποίες κωδικοποιούν διαφορετικές σημασίες λέξεων στον σημασιολογικό χώρο, μπορεί να μας βοηθήσει να βελτιώσουμε την κατανόηση της φυσικής γλώσσας.

Οι μέθοδοι που παράγουν πολλαπλές κατανεμημένες αναπαραστάσεις ανά λέξη μπορούν να ομαδοποιηθούν σε δύο ευρείες κατηγορίες. Οι μέθοδοι χωρίς επίβλεψη παράγουν αναπαραστάσεις πολλαπλών διανυσμάτων χωρίς τη χρήση σημασιολογικών λεξικών πόρων. Στο μοντέλο [48], τα κεντροειδή των κλάσεων τα οποία εξαρτώνται από τα διαφορετικά συμφραζόμενα που εμφανίζεται μια λέξη, χρησιμοποιήθηκαν για να δημιουργήσουν ένα σύνολο «ειδικών για κάθε διαφορετικό νόημα» αναπαραστάσεων για κάθε λέξη. Επόμενες έρευνες βασίστηκαν σε παρόμοιες προσεγγίσεις προσθέτοντας τη χρήση αρχιτεκτονικών νευρωνικών δικτύων που ενσωματώνουν τόσο το τοπικό όσο και το καθολικό πλαίσιο συμφραζομένων κατά τη διάρκεια της εκπαίδευσης [49, 50, 51]. Μια πιθανοτική προσέγγιση εισήχθη από το [52], όπου το μοντέλο Skip-Gram του Word2Vec τροποποιήθηκε για να μάθει πολλαπλές αναπαραστάσεις. Επιπλέον, ενσωματώθηκαν έμμεσα θέματα στο μοντέλο Skip-Gram, με αποτέλεσμα την δημιουργία διανυσμάτων που μοντελοποίησαν τη σημασιολογία μιας λέξης κάτω από διαφορετικά πλαίσια συμφραζομένων [53, 54, 55].Οι προσεγγίσεις με τη χρήση επίβλεψης, βασισμένες σε πρότερη γνώση που αποκτήθηκε από λεξικά (π.χ. WordNet) μαζί με αλγορίθμους διαχωρισμού των διαφορετικών νοημάτων μιας λέξης, εισήχθησαν επίσης για την εξαγωγή [56, 57] πολλαπλών αναπαραστάσεων.

0.5.1 Ενοποιημένο Διαθεματικό Μοντέλο

Το σύστημά μας ακολουθεί μια προσέγγιση τεσσάρων βημάτων, η οποία μπορεί να απεικονιστεί στο σχήμα 0.1:

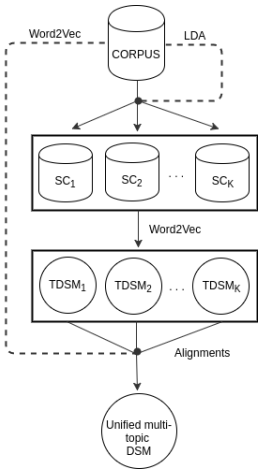
1. **Ενιαίο Κατανεμημένο Σημασιολογικό Μοντέλο.** Δοθέντος ενός συνόλου δεδομένων από πολλά κείμενα εκπαιδεύουμε ένα DSM που κωδικοποιεί τη σημασιολογία των διαφορετικών λέξεων σε αναπαραστάσεις που αποτελούνται από ένα διάνυσμα για την καθεμία. Στο Καθολικό Κατανεμημένο Σημασιολογικό Μοντέλο θα αναφερόμαστε ως Global-DSM.

2. **Διαθεματικά Κατανεμημένα Σημασιολογικά Μοντέλα.** Στη συνέχεια, ένα διαθεματικό μοντέλο εκπαιδεύεται χρησιμοποιώντας την ίδια συλλογή δεδομένων. Το μοντέλο αυτό χωρίζει τα δεδομένα σε K (πιθανώς αλληλεπικαλυπτόμενα) υποκείμενα. Έπειτα, ένα DSM εκπαιδεύεται σε κάθε υποκείμενο με αποτέλεσμα τα K DSMs το καθένα βασισμένο σε διαφορετικό θέμα (TDSMs). Η θεματική προσαρμογή του σημασιολογικού χώρου λαμβάνει υπόψη τις παραλλαγές που παρουσιάζει μια λέξη όταν συναντάται σε διαφορετικούς θεματικούς τομείς και ως εκ τούτου οδηγεί στη δημιουργία ειδικών θεματικών διανυσμάτων (topic embeddings).

3. **Αντιστοίχιση θεματικών διανυσμάτων.** Στη συνέχεια, προβάλλουμε το διανυσματικό χώρο κάθε DSM στον κοινό χώρο του Global-DSM, χρησιμοποιώντας μια λίστα λέξεων που αποτελούν άγκυρες —παρουσιάζονται σε ίδιες σχετικές θέσεις— μεταξύ των δύο αυτών χώρων, οι

οποίες επιλέγονται μέσω ενός ανεξάρτητου συστήματος μάθησης χωρίς επίβλεψη. Στο ενοποιημένο σημασιολογικό χώρο, κάθε λέξη αντιπροσωπεύεται από ένα σύνολο θεματικών διανυσμάτων που προηγουμένως απομονώθηκαν σε ξεχωριστούς διανυσματικούς χώρους, δημιουργώντας έτσι ένα ενοποιημένο πολυθεματικό DSM (UTDSM).

4. **Εξομάλυνση πολλαπλών διαθεματικών διανυσμάτων.** Ως επόμενο βήμα, προβαίνουμε σε μια τεχνική εξομάλυνσης για να ομαδοποιήσουμε τα θεματικά διανύσματα κάθε λέξης σε N Κανονικές κατανομές μέσω ενός Μοντέλου Μίξης Κανονικών Κατανομών (GMM). Αυτό το βήμα μειώνει το θόρυβο που εισάγεται μέσω των αντιστοιχήσεων των 2 χώρων και των αραιών δεδομένων εκπαίδευσης και συμβάλει στην προσέγγιση των διαφορετικών νοημάτων μέσω ενοποίησης διανυσμάτων από διάφορα θέματα.



Σχήμα 0.1: Ξεκινώντας από ένα αρχικό κείμενο, δημιουργούνται K θέματοκεντρικά subcorpora (SC_i) και στη συνέχεια δημιουργούνται K χώροι θεματικής σημασιολογίας ($TDSM_i$) οι οποίοι στη συνέχεια προβάλλονται σε έναν ενιαίο χώρο.

Το πρώτο βήμα προς τη θεματική προσαρμογή του σημασιολογικού χώρου είναι εξαγωγή των θεμάτων, χρησιμοποιώντας τον Αλγόριθμο Latent Dirichlet (LDA) [58]. Η βασική του ιδέα είναι ότι τα έγγραφα αντιπροσωπεύονται ως τυχαία μείγματα πάνω σε θέματα, όπου κάθε θέμα ορίζεται ως κατανομή πιθανοτήτων σε μια συλλογή λέξεων. Τα προκύπτοντα θέματα χρησιμοποιούνται στη συνέχεια για την εκπαίδευση των θεματικών DSM. Σε αυτή την εργασία, επιλέγουμε το μοντέλο Word2Vec [25]. Ο εγγενής μη-ντετερμινισμός του αλγορίθμου Word2Vec οδηγεί στη δημιουργία συνεχών διανυσματικών χώρων που δεν είναι φυσικά ευθυγραμμισμένα σε ένα ενιαίο σημασιολογικό σύστημα αναφοράς, αποκλείοντας τη σύγκριση μεταξύ των αναπαραστάσεων τνω λέξεων διαφορετικών θεματικών πεδίων. Για να παρακάμψουμε αυτόν τον περιορισμό, πρέπει να προβάλλουμε τις εκφράσεις λέξεων των TDSM σε ένα κοινό διανυσματικό χώρο. Συγκεκριμένα, υποθέτουμε ότι τα TDSMs επιδεικνύουν σημαντικές διακυμάνσεις στη αναπαραστάσεις των πολύσημων λέξεων, ενώ διατηρείται η σχετική σημασιολογική απόσταση μεταξύ μονόσημων λέξεων. Αυτή η υπόθεση μας ώθησε να σκεφτούμε τις μονοσήμαντες λέξεις ως *άγκυρες* μεταξύ σημασιολογικών χώρων. Ένας τρόπος για να ανακτήσει κανείς τη λίστα των αγκυρών είναι να την εξάγει από λεξιλογικούς πόρους όπως το WordNet [59]. Ωστόσο, αυτή η μέθοδος περιορίζεται στις γλώσσες όπου υπάρχουν τέτοιοι λεξικοί πόροι και εξαρτάται από τη λεξική κάλυψη και την ποιότητα αυτών των πόρων. Για να ξεπεραστούν οι παραπάνω περιορισμοί, προτείνουμε μια πλήρως αυτόματη μέθοδο. Παρόλο που οι διαφορετικοί διανυσματικοί χώροι δεν είναι ευθυγραμμισμένοι ως προς κοινούς άξονες, οι αντίστοιχες μήτρες ομοιότητας (όταν κανονικοποιηθούν) είναι. Με βάση αυτή την παρατήρηση, υπολογίζουμε την ομοιότητα μεταξύ μιας δεδομένης λέξης και κάθε άλλης λέξης στο λεξιλόγιο (κατανομή ομοιότητας) για το διαφορετικά θέματα και των καθολικό χώρο. Στη συνέχεια, υποθέτουμε ότι οι καλές σημασιολογικές άγκυρες θα πρέπει να έχουν παρόμοιες κατανομές ομοιότητας ανάμεσα στους δύο

χώρους, όπως απεικονίζεται στο Σχήμα 5.3. Έστω το V να είναι η τομή των λεξιλογίων των TDSMs και Global-DSM, K τα διαφορετικά θέματα και d η διάσταση των διανυσματικών αναπαραστάσεων. Στη συνέχεια ορίζουμε το $X_k \in \mathbb{R}^{|V|\times d}$ ως μήτρα διανυσμάτων του TDSM k και το $Y \in \mathbb{R}^{|V|\times d}$ ως μήτρα διανυσμάτων του καθολικού DSM, όπου η σειρά i κάθε μήτρας αντιστοιχεί στην κανονικοποιημένη αναπαράσταση μιας λέξης στο V. Στη συνέχεια, ορίζουμε τα $S_k = X_k X_k^T$, $S_g = YY^T$ $\in \mathbb{R}^{|V|\times|V|}$ να είναι οι πίνακες κατανομής ομοιότητας για το TDSM k και το global-DSM, αντίστοιχα. Στόχος μας είναι να εξαγάγουμε μια λίστα σημασιολογικών αγκυρών A που ελαχιστοποιεί την ευκλείδεια απόσταση μεταξύ των δύο διαφορετικών κατανομών ομοιότητας. Συγκεκριμένα, για κάθε λέξη i υπολογίζουμε τη μέση σημασιολογική κατανομή σε όλα τα θέματα:

$$< s_k^i >_k = \frac{1}{K} \sum_{k=1}^{K} s_k^i \tag{0.2}$$

$$\| < s_k^i >_k \ - s_g^i \|_2 \ , \quad \forall \, i = 1, \ldots, |V| \tag{0.3}$$

όπου s_g^i, s_k^i είναι η γραμμή i της μήτρας ομοιότητας S_g και S_k, αντίστοιχα, που αντιπροσωπεύει την κατανομή ομοιότητας μεταξύ της λέξης i και κάθε άλλης λέξης στο λεξιλόγιο V. Στη συνέχεια επιλέγουμε τις άγκυρες $|A|$ ως τις λέξεις με τις μικρότερες τιμές σύμφωνα με το κριτήριο 0.3. Επιπλέον, υποθέτουμε ότι υπάρχει ένας ορθογώνιος πίνακας μετασχηματισμού μεταξύ των θεματικών αναπαραστάσεων των εξαγόμενων σημασιολογικών αγκυρών κάθε TDSM(χώρος πηγής) και των αντίστοιχων αναπαραστάσεων του καθολικού DSM(χώρος προορισμού) [60, 61, 62]. Υποθέτουμε ότι το $\alpha_k^j \in \mathbb{R}^d$ είναι το διάνυσμα της λέξης j-th *άγκυρας* στον χώρο πηγής και $\alpha_g^j \in \mathbb{R}^d$ είναι η αντίστοιχη αναπαράσταση του φορέα στο χώρο προορισμού. Ο πίνακας μετασχηματισμού $M_k \in \mathbb{R}^{d\times d}$ που προβάλλει τον πρώτο χώρο στο τελευταίο μάθει μέσω της επίλυσης του ακόλουθου προβλήματος βελτιστοποίησης περιορισμού[63]:

$$\min_{M_k} \sum_{j=1}^{|A|} \|M_k \alpha_k^j - \alpha_g^j\|_2^2, \ \text{ s.t. } \ M_k M_k^T = \mathbb{I} \tag{0.4}$$

Η προβολή των διανυσμάτων των διαφορετικών χώρων στον ενοποιημένο χώρο επιτυγχάνεται μέσω της εφαρμογής της εξίσωσης 0.4 σε κάθε TDSM. Συγκεκριμένα, δεδομένης μιας λέξης και k-οστής θεματικής κατανομής $x_k \in \mathbb{R}^d$, υπολογίζουμε την προβαλλόμενη αναπαράσταση $x'_k \in \mathbb{R}^d$ ως εξής:

$$x'_k = M_k x_k \tag{0.5}$$

Ξεκινώντας από το σύνολο ευθυγραμμισμένων θεματικών διανυσμάτων $\{x'_k\}_{k=1}^K$ για κάθε λέξη, μαθαίνουμε ένα μοντέλο μίξης κανονικών κατανομών με N συνιστώσες. Αυτό το βήμα λειτουργεί ως έμμεσος τρόπος κατάτμησης του χώρου των θεματικών αναπαραστάσεων για κάθε λέξη προκειμένου να καταγραφούν πιο χρήσιμα υπερ-θέματα —ένωση θεμάτων— τα οποία αντιπροσωπεύουν καλύτερα τις διαφορετικές σημασίες τους. Υποθέτουμε ότι κάθε κανονική κατανομή σχηματίζει μια σημασιολογικά συνεκτική μονάδα που αντιστοιχεί σε ένα διαφορετικό νόημα της εκάστοτε λέξης. Στη συνέχεια, το κεντροειδές κάθε κανονικής κατανομής χρησιμοποιείται ως αντιπρόσωπος κάθε συνιστώσας, οδηγώντας σε ένα νέο σύνολο *εξομαλυμένων* θεματικών διανυσμάτων $\{x_n^*\}_{n=1}^N$ για κάθε λέξη, όπου $x_n^* \in \mathbb{R}^d$.

0.5.2 Σύνολο δεδομένων ομοιότητας με βάση τα συμφραζόμενα

Για να εκτιμηθεί η σημασιολογική ομοιότητα μεταξύ ενός ζευγαριού λέξεων, χρησιμοποιούμε το καθιερωμένο σύνολο δεδομένων Stanford Contextual Word Similarity (SCWS) [49] το οποίο αποτελείται από ζεύγη λέξεων 2 003 με καθορισμένες σημασιολογικές ομοιότητες. Ακολουθώντας τις νόρμες αξιολόγησης που προτείνονται στη βιβλιογραφία, χρησιμοποιούμε τις μετρικές AvgSimC και MaxSimC, προτεινόμενες αρχικά στο [48].

0.5.3 Σύνολα δεδομένων για Προβλήματα Φυσικής Γλώσσας

Ταξινόμηση Κειμένου. Χρησιμοποιήσαμε το 20NewsGroup σύνολο δεδομένων, που αποτελείται από 20 000 έγγραφα. Ο στόχος μας είναι να ταξινομήσουμε κάθε έγγραφο σε μία από τις 20 διαφορετικές κλάσεις βασισμένοι στο περιεχόμενο του.
Αναγνώριση Παράφρασης. Σε αυτό το πρόβλημα στοχεύουμε στην εξακρίβωση της παράφρασης ανάμεσα σε ζεύγη προτάσεων μέσω του συνόλου δεδομένων της Microsoft [64].
Αναπαραστάσεις σε επίπεδο προτάσεων και εγγράφων.
Δεδομένου ενός εγγράφου ή μιας φράσης D, όπου w_d αντιστοιχεί στη d-ιοστή λέξη στο D, εξάγουμε την αναπαράστασή της με τρεις διαφορετικούς τρόπους:

$$\text{AvgC}_\text{D} = \frac{1}{|D|}\sum_{d=1}^{|D|}\sum_{k=1}^{K} \text{p}(k|D)x'_k(w_d), \tag{0.6}$$

$$\text{Avg}_\text{D} \quad = \frac{1}{|D|}\sum_{d=1}^{|D|}\sum_{k=1}^{K} \frac{1}{K}x'_k(w_d), \tag{0.7}$$

$$\begin{aligned} \text{MaxC}_\text{D} &= \frac{1}{|D|}\sum_{w=1}^{|D|} x'_m(w_d) \\ \text{s.t.} \quad m &= \underset{k=1,..,K}{\text{argmax}}\{\text{p}(k|D)\}, \end{aligned} \tag{0.8}$$

όπου $\text{p}(k|D)$ υποδηλώνει την πιθανότητα του θέματος k που επιστρέφεται από την LDA που δέχεται ως είσοδο την πρόταση/έγγραφο D και $x'_k(w_d)$ είναι η προβεβλημένη στον κοινό χώρο αναπαράσταση της λέξης w_d για το θέμα k. Για την περίπτωση της αναγνώρισης παραφράσεων, εξαγάγουμε ένα μόνο διάνυσμα χαρακτηριστικών για κάθε ζεύγος προτάσεων, συνδυάζοντας τα χαρακτηριστικά των μεμονωμένων προτάσεων.

Μετά την εξαγωγή χαρακτηριστικών, εκπαιδεύουμε έναν γραμμικό ταξινομητή με χρήση διανυσμάτων υποστήριξης (SVM) [65] χρησιμοποιώντας τα προτεινόμενα σύνολα δεδομένων εκπαίδευσης/δοκιμής και για τα δύο προβλήματα. Αναφέρουμε τα καλύτερα αποτελέσματα για κάθε πειραματική διαμόρφωση μετά από τη ρύθμιση της παραμέτρου ποινής του SVM του όρου σφάλματος χρησιμοποιώντας τις αναπαραστάσεις λέξεων 500-διαστάσεων.

0.6 Πειραματικά αποτελέσματα & Συζήτηση

0.6.1 Ομοιότητα με βάση τα συμφραζόμενα

Στον Πίνακα 0.5 συγκρίνουμε το μοντέλο μας (UTDSM) με τις αναπαραστάσεις στον καθολικό χώρο (Global-DSM) και τις καλύτερες από απόψη επίδοσης προσεγγίσεις πολλαπλών διανυσμάτων για το πρόβλημα σημασιολογικής ομοιότητας με βάση το πλαίσιο συμφραζομένων. Είναι σαφές ότι όλες οι διαφορετικές παραλλαγές του UTDSM έχουν καλύτερη απόδοση από το global-DSM και για τις δύο μετρικές ομοιότητας. Η χρήση μιας ενιαίας κανονικής κατανομής (UTDSM + GMM (1)) στο στάδιο εξομάλυνσης της μεθόδου παράγει παρόμοια αποτελέσματα με το global-DSM. Αυτό αναμένεται καθώς και οι δύο μέθοδοι παρέχουν μια κεντροειδής μονοδιανυσματική αντιπροσώπευση μιας λέξης. Όσον αφορά το MaxSimC, το μοντέλο αποδίδει σταθερά υψηλότερη απόδοση όταν η λίστα σημασιολογικών αγκυρών εξάγεται μέσω της μεθόδου μας, αντί να χρησιμοποιούμε τυχαία επιλεγμένες λέξεις άγκυρας (UTDSM Random). Παρατηρούμε επίσης ότι η τυχαία αγκύρωση εκτελεί ελαφρώς χειρότερη από την UTDSM σε σχέση με την μετρική AvgSimC. Αυτό το αποτέλεσμα επικυρώνει την υπόθεσή μας ότι οι αναπαραστάσεις λέξεων, οι οποίες μοιράζονται συνεπείς κατανομές ομοιό-

τητας μεταξύ διαφορετικών θεματικών χώρων, συνιστούν κατάλληλες *σημασιολογικές άγκυρες* που καθορίζουν τις αντιστοιχίσεις μεταξύ σημασιολογικών διανυσματικών χώρων.

Μέθοδος	**AvgSimC**	**MaxSimC**
Liu et. al(2015)[54]	67.3	68.1
Liu et. al(2015b)[53]	69.5	67.9
Amiri et. al(2016)[66]	70.9	-
Lee et. al(2017)[67]	68.7	67.9
Guo et. al(2018)[68]	69.3	68.2
300-διαστάσεις		
Global-DSM	67.1	67.1
UTDSM Random	69.1 ± 0.1	66.4 ± 0.2
UTDSM	**69.6**	67.1
UTDSM + GMM (1)	67.4	67.4
UTDSM + GMM (2)	68.4	**68.3**
UTDSM + GMM (3)	68.9	**68.3**
500-διαστάσεις		
Global-DSM	67.6	67.6
UTDSM Random	69.4 ± 0.1	66.5 ± 0.3
UTDSM	**70.2**	68.0
UTDSM + GMM (1)	67.6	67.6
UTDSM + GMM (2)	68.8	**68.6**
UTDSM + GMM (3)	69.0	68.5

Πίνακας 0.5: Σύγκριση απόδοσης μεταξύ των διαφορετικών προσεγγίσεων στο σύνολο δεδομένων SCWS, με χρήση της συσχέτισης Spearman. Το UTDSM αναφέρεται στην προβαλλόμενη διαθεματική αναπαράσταση, το UTDSM Random αναφέρεται στην περίπτωση που τυχαίες λέξεις χρησιμοποιούνται ως άγκυρες και το GMM (c) αντιστοιχεί στην εξομάλυνση μέσω GMM με c συνιστώσες.

Επιπλέον, παρατηρούμε ότι η εξομάλυνση μέσω GMM έχει διαφορετική επίδραση στις δύο μετρικές, MaxSimC και AvgSimC. Συγκεκριμένα, η AvgSimC αποδίδει με συνέπεια τα χαμηλότερα αποτελέσματα όταν η εξομάλυνση GMM εφαρμόζεται. Αποδίδουμε αυτή τη συμπεριφορά σε πιθανή απώλεια της πληροφορίας του μοντέλου —μείωση του αριθμού των θεματικών διανυσμάτων— που είναι ικανή να οδηγήσει σε απώλειας χρήσιμης σημασιολογικής πληροφορίας. Ταυτόχρονα, η τεχνική εξομάλυνσης βελτιώνει εξαιρετικά την απόδοση της MaxSimC σε όλες τις πιθανές περιπτώσεις. Δεδομένου ότι αυτή η μετρική είναι πιο ευαίσθητη στις θορυβώδεις αναπαραστάσεις λέξεων, αυτό το αποτέλεσμα δείχνει ότι η τεχνική μας μειώνει το θόρυβο που εισάγεται στο σύστημά μας, κρατώντας σημασιολογικά βελτιωμένες εκδόσεις των αναπαραστάσεων κάθε λέξης.

Συνολικά, η απόδοση του μοντέλου μας είναι συγκρίσιμη και συχνά καλύτερη σε σχέση με άλλα μοντέλα βάσει της AvgSimC, για τις διαθεματικές αναπαραστάσεις διάστασης 500. Επιτυγχάνουμε επίσης την καλύτερη επίδοση για τη μετρική MaxSimC, χρησιμοποιώντας εξομαλυμένες θεματικές αναπαραστάσεις διαστάσεων 300 ή 500 με 2 ή 3 συνιστώσες κανονικών κατανομών.

0.6.2 Προβλήματα Φυσικής Γλώσσας

Εκτός από το τυποποιημένο κριτήριο αξιολόγησης της ομοιότητας μεταξύ λέξεων, διερευνάμε επίσης την αποτελεσματικότητα του μοντέλου μας μας σε επίπεδο εγγράφων και προτάσεων: ταξινόμηση κειμένου και αναγνώριση παραφράσεων.

Τα αποτελέσματα της αξιολόγησης σχετικά με την ταξινόμηση κειμένου παρουσιάζονται στον Πίνακα 0.6. Παρατηρούμε ότι το μοντέλο μας λειτουργεί καλύτερα από τις μονοδιανυσματικές αναπραστάσεις σε όλες τις μετρικές και για τις δύο προσεγγίσεις (AvgC_D, Avg_D), ενώ η χρήση κυρίαρχων θεμάτων φαίνεται να έχει χαμηλότερη απόδοση (MaxC_D). Η διαφορά απόδοσης μεταξύ του κυρίαρχου και του μέσου θέματος προκύπτει επειδή τα θέματα που ανακαλύφθηκαν από τον αλγόριθμο LDA

Μέθοδος	Ακρίβεια	Ανάκληση	F1-σκορ	Ευστοχία
LDA	39.7	41.8	38.8	41.8
Global-DSM	62.9	63.3	62.9	63.3
MaxC_D	61.9	63.0	62.0	63.0
Avg_D	63.5	64.6	63.3	64.3
AvgC_D	**64.6**	**65.5**	**64.5**	**65.5**

Πίνακας 0.6: Αποτελέσματα αξιολόγησης ταξινόμησης κειμένου πολλαπλών κατηγοριών.

είναι πιθανώς διαφορετικά από τα διαφορετικά θέματα του συγκεκριμένου συνόλου δεδομένων. Ως εκ τούτου, ένα μέσο μίγμα διανυσμάτων επιτυγχάνει καλύτερα αποτελέσματα από την επιλογή διανυσμάτων με βάση την υψηλότερη πιθανότητα θεμάτων, αγνοώντας έτσι θέματα τα οποία μπορεί να έχουν παρόμοιες πιθανότητες και έχουν σημαντική συμβολή στο αποτέλεσμα.

Μέθοδος	Ακρίβεια	Ανάκληση	F1-σκορ	Ευστοχία
Global-DSM	68.6	69.2	62.0	69.2
MaxC_D	**69.0**	69.3	62.1	69.3
Avg_D	67.7	**69.4**	**64.0**	**69.4**
AvgC_D	68.8	69.4	62.6	69.4

Πίνακας 0.7: Αποτελέσματα αξιολόγησης σχετικά με την πρόβλημα ανίχνευσης παραφράσεων.

Τα αποτελέσματα για το πρόβλημα αναγνώρισης παραφράσεων παρουσιάζονται στον Πίνακα 0.7. Το Avg_D αποφέρει τα καλύτερα αποτελέσματα, ειδικά σε μετρήσεις F1 που δείχνουν ότι οι διαθεματικές αναπαραστάσεις είναι σημασιολογικά πλουσιότερες από τις μονοδιανυσματικές αναπαραστάσεις (Global-DSM). Παρόλο που εφαρμόζουμε τις κατανομές θέματος $\text{p}(k|D)$ που εξάγονται από LDA (μοντέλο επιπέδου εγγράφου) σε ένα πρόβλημα που αποτελείται από προτάσεις, οι βελτιώσεις σε σχέση με τη μονοδιανυσματικές αναπαραστάσεις εμφανίζονται επίσης στις περιπτώσεις AvgC_D και MaxC_D.

Συνολικά, το προτεινόμενο μοντέλο UTDSM ξεπερνά το μονοδιανυσματικό μοντέλο Global-DSM στο πρόβλημα της σημασιολογικής ομοιότητας με βάση τα συμφραζόμενα και προβλήματα επεξεργασίας φυσικής γλώσσας. Συγκεκριμένα όσον αφορά τη σημασιολογική ομοιότητα, η προσέγγισή μας εξομάλυνσης βελτιώνει τα αποτελέσματά μας στη MaxSimC, το οποίο εξαρτάται περισσότερο από το θόρυβο, ενώ επηρεάζει ελαφρώς τη AvgSimC. Στην αξιολόγηση για προβλήματα φυσικής γλώσσας, παρατηρούμε ότι στην περίπτωση της ταξινόμησης κειμένου οι μέσες μέθοδοι συνδυασμού των διανυσμάτων επιτυγχάνουν καλύτερα αποτελέσματα. Στην αναγνώριση παράφρασης, η επιλογή του διανύσματος με την υψηλότερη πιθανότητα βελτιώνει τα αποτελέσματα ακρίβειας, ενώ οι υπόλοιπες μέθοδοι λειτουργούν καλύτερα σε άλλες μετρήσεις. Σε όλες σχεδόν τις περιπτώσεις, εκτός από την χρήση της μεθόδου MaxC_D στην ταξινόμηση κειμένου, οι πολλαπλές αναπαραστάσεις μας έχουν καλύτερες επιδόσεις από το μοντέλο μονής αναπαράστασης.

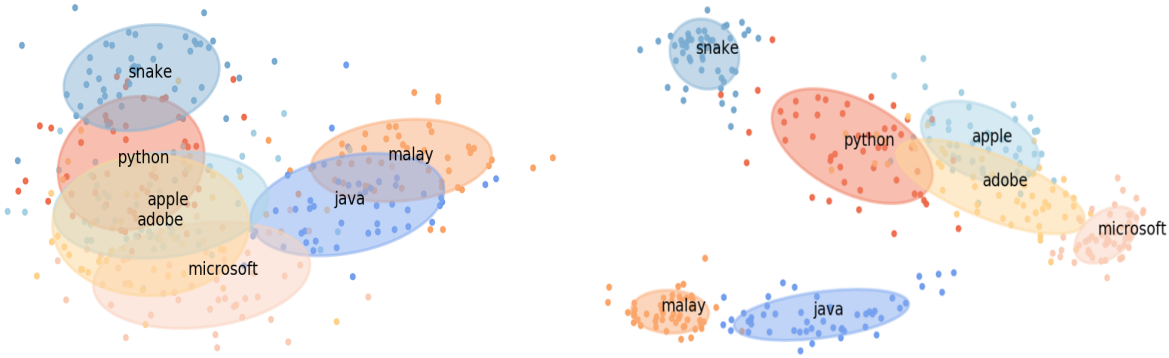
0.7 Ανάλυση Cross Domain

Σε αυτή την ενότητα θα εκτελέσουμε μια ποιοτική ανάλυση των αποτελεσμάτων μας, θα απεικονίσουμε τα αποτελέσματα των μοντέλων μας και θα εξετάσουμε το αντίκτυπο της προβολής σε έναν κοινό χώρο για συγκεκριμένες λέξεις.

0.7.1 Οπτικοποίηση της σημασιολογικής διακύμανσης

Στην Εικόνα 0.2 απεικονίζουμε τις θεματικές αναπαραστάσεις επτά λέξεων πριν και μετά την προβολή τους στον ενοποιημένο χώρο, χρησιμοποιώντας την μέθοδο της ανάλυσης πρωτεύουσων συνιστωσών. Παρουσιάζουμε επιπλέον τις κανονικές κατανομές που αντιστοιχούν στις πολλαπλές διαθεματικές αναπαραστάσεις κάθε λέξης οι οποίες αντανακλούν τη διακύμανση του νοήματος τους. Το κέντρο κάθε κατανομής καθορίζεται από τον μέσο της αντίστοιχης κατανομή και η επιφάνεια που

καταλαμβάνει από τον πίνακα συνδιακύμανσης. Αριστερά, απεικονίζουμε τη θέση των λέξεων πριν την προβολή τους στον κοινό χώρο. Στον χώρο αυτό, οι λέξεις επιδεικνύουν παρόμοια κάλυψη περιοχής ανεξάρτητα από την πολυσημία τους. Μετά από τις αντιστοιχίσεις, βλέπουμε στα δεξιά ότι η περιοχή κατανομής μιας λέξης είναι ενδεικτική του βαθμού πολυσημίας της. Συγκεκριμένα, παρατηρούμε ότι η διακύμανση των αναπαραστάσεων γίνεται μεγαλύτερη για τις περιπτώσεις πολύσημων λέξεων όπως “python”, “java”, “adobe”, προκειμένου να αποδοθούν πιθανότητες στις διαφορετικές έννοιές τους. Μονόσημες λέξεις όπως “snake”, “microsoft” και “malay”, έχουν σαφώς μικρότερες διακυμάνσεις. Συγκρίνοντας τις δύο διαφορετικές εικόνες, μπορούμε να δούμε ότι το σημασιολογικό εύρος τους μεταβάλλεται ανάλογα με την πολυσημία τους. Επιπλέον, παρατηρούμε ότι οι σημασιολογικές σχέσεις μεταξύ των λέξεων έχουν αποδοθεί πολύ καλύτερα μετά την προβολή τους στον κοινό διανυσματικό χώρο.



Σχήμα 0.2: Μια δισδιάστατη αναπαράστασεις —χρησιμοποιώντας τον PCA— που απεικονίζει τις αναπαραστάσεις 7 λέξεων πριν (αριστερά) και μετά (δεξιά) προβολή των TDSM στην κοινό διανυσματικό χώρο.

0.8 Πειραματικά Συμπεράσματα

Συνολικά, η προτεινόμενη μέθοδος μας για τη δημιουργία μη πολλαπλών διαθεματικών αναπαραστάσεων πέτυχε τα καλύτερα αποτελέσματα για το βασικό πρόβλημα σύγκρισης μεθόδων πολλαπλών διανυσμάτων, σύμφωνα με τη βιβλιογραφία. Αν και τα θέματα σε σχέση με τα διαφορετικά νοήματα των λέξεων δεν είναι απόλυτα ταυτισμένα, η διαισθητική μας προσέγγιση εξομάλυνσης βελτίωσε τα αποτελέσματά μας. Ένα σχήμα προσαρμοστικού GMM που καθορίζει τον αριθμό των κανονικών κατανομών διαφορετικά ανά λέξη, θα μπορούσε να είναι πιο αποτελεσματικό καθώς ο αριθμός των νοημάτων διαφέρει για διαφορετικές λέξεις.

Τα πειράματα σε κλασσικά προβλήματα φυσικής γλώσσας έδειξαν ότι ένας απλός συνδυασμός των πολλαπλών διανυσμάτων μας βελτιώνει την επίδοση σε σύγκριση με μοντέλα μονής αναπαράστασης. Επιπλέον, η ποιοτική ανάλυση επιβεβαιώνει την ερμηνεία του μοντέλου μας και επικυρώνει τις διαφορές μεταξύ ευθυγραμμισμένων και μη ευθυγραμμισμένων χώρων. Τέλος, περιγράφει με σαφήνεια τις μελλοντικές κατευθύνσεις όπως η χρήση προσαρμοστικού αριθμού κανονικών κατανομών.

Λέξεις κλειδιά

Υπολογιστική Νευροεπιστήμη, Βαθειά Μάθηση, Μηχανική Μάθηση, Διανυσματικές Αναπαραστάσεις Λέξεων, Πολλαπλές Διανυσματικές Αναπαραστάσεις Λέξεων, Θεματική Μοντελοποίηση, Επεξεργασία Φυσικής Γλώσσας, Γνωστική λειτουργία & Φυσική Γλώσσα

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Natural Language Representations

Language modelling is the task of predicting the next word in a text given the previous words. It is probably the simplest language processing task with concrete practical applications such as intelligent keyboards, email response suggestion [3], spelling autocorrection etc. Unsurprisingly, language modelling has a rich history. Classic approaches are based on n-grams and employ smoothing to deal with unseen n-grams [4]. More recently, feed-forward neural networks have been replaced with recurrent neural networks (RNNs) [5] and long short-term memory networks (LSTMs) [70] for language modelling. Many other language models that extend the classic LSTM have been proposed in recent years. Probably the most remarkable aspect about language modelling is that despite its simplicity, it is core to many of the later advances in Natural Language Processing and Understanding field:

- **Word embeddings/representations**: The objective of word2vec is a simplified version of language modelling.
- **Sequence-to-sequence models**: Such models generate an output sequence by predicting one word at a time.
- **Pretrained language models**: These methods use representations previously extracted from language models for transfer learning.

This conversely means that many of the most important recent advances in NLP reduce to a form of language modelling. Multi-task learning is a general method for sharing parameters between models that are trained on multiple tasks. In neural networks, this can be done easily by tying the weights of different layers. Intuitively, multi-task learning encourages the models to learn representations that are useful for many tasks. This is particularly useful for learning general, low-level representations, to focus a model's attention or in settings with limited amounts of training data.

Multi-task learning is now used across a wide range of NLP tasks and leveraging existing or "artificial" tasks has become a useful tool in the NLP repertoire. Dense vector representations of words or word embeddings have been used as early as 2003 [23]. The main innovation that was proposed in [25] was to make the training of these word embeddings more efficient by removing the hidden layer and approximating the objective. While these changes were simple in nature, they enabled—together with the efficient word2vec implementation—large-scale training of word embeddings. Word2vec comes in two flavours continuous bag-of-words (CBOW) and skip-gram. They differ in their objective: one predicts the centre word based based on the surrounding words, while the other does the opposite. While these embeddings are no different conceptually than the ones learned with a feed-forward neural network, training on a very large corpus enables them to approximate certain relations between words such as gender, verb tense, and country-capital relations.

While the relations word2vec captured had an intuitive simplicity, later studies showed that there is nothing inherently special about word2vec: Word embeddings can also be learned via matrix factorization [74, 75] and with proper tuning, classic matrix factorization approaches like SVD and LSA achieve similar results[76] bridging the theoretical gap between count-based models([77, 38]) and predict-based (neural network) models. However, due to their performance and computational efficiency predict-based models [22], tend to be mostly used in literature.

From a computational perspective, word representations aim to facilitate computers to detect aspects of meaning in language, as well as to encode this information in a formalistic way that enables their interpretability by computers. The importance of understanding the semantics of lexical units is paramount to language comprehension and acquisition, as they constitute the basic components of human language. Furthermore, polysemy is a linguistic phenomenon commonly found in corpora that plays a major role in language comprehension. Specifically, the polysemic nature of a word eg. *python* could drastically affect its representation, depending on the context it belongs to.

Recently, cognitive experiments seem to indicate that the understanding of these semantics could be aided by the fundamental cognitive relationships between words [6]. One straightforward implication of this observation is that a word's meaning highly depends on the semantic relationships it shares with other words. Furthermore, recent work by computational neuroscientists and linguists, have shown that a mapping between cognitive and semantic space is feasible [7, 8, 9, 10].

Some of the applications of NLP that integrate the above information in their systems include: automatic translation of texts, information retrieval, automatically summarizing text, natural language generation, question answering, search engines and converting spoken speech into text.

1.2 Cognition & Natural Language

Neural language processing would benefit from a better understanding of human processes: as it has been said by several researchers recently, traditional machine learning approaches have brought rapid and important improvements in different natural language processing tasks but the field may have to face from now on more difficult problems (e.g. discourse planning, argumentative analysis, etc.) that would benefit from a better understanding of the processes involved in the brain. Additionally, researchers are again interested in evaluating the relevance of their models according to a cognitive dimension. Many of the existing computational models attempt to study language tasks under cognitively plausible criteria (such as memory and processing limitations) that humans face. New machine learning techniques, especially deep learning, bring back to the front scene a new version of neural networks that seems both more powerful and more sound, from a technical as well as a cognitive point of view. Last but not least, cognitive science also benefits and sometimes takes inspiration from computational models.

Language acquisition is another domain where computation models and cognitive sciences have a fruitful dialogue. To a certain extent, neuroimaging has renewed the study of language by making it possible to directly observe processes in the brain but for the rest, language is only known through direct production, i.e. language utterances. Therefore, the study of language acquisition by children is crucial, since it gives an overview on what vocabulary and structures are mastered first.

The study of people with *language pathologies* has also attracted a high interest in the last decades, see for one example among many others[78]. This field can be compared, to a certain extent, to the research done in language acquisition: the idea is to get an accurate description of the language production of people with languages pathologies so as to find what is deficient in their speech and then propose relevant treatments or relevant measures to help them overcome their difficulties. Additionally cognitive science has of course a long tradition of mapping language deficiencies with specific areas in the brain [79].

At first sight, *language evolution* can be seen more as a social process than as a cognitive one. However, language evolution has to take into account how a group of individuals master a language and transmit this knowledge to their infants. This is the core of social cognition, that aims at studying how individual knowledge interacts so as to give birth to social processes. Language evolution can thus be seen as one of the central topics of social cognition [80, 81, 82, 83].

1.2.1 Semantics and Brain

The study of semantics in the brain is a branch of psycholinguistics that incorporates the understanding of semantics and the neurological structures that are involved. It attempts to answer the unanswered

question of "how objects and concepts are represented and processed in the human brain"[11]. This field of study has received an enormous amount of research because in essence, semantics is what allows us to verbalize and express ourselves about the people, places, and things in our lives. It is essential to human communication and exists within all human beings across languages. Although widely studied, understanding the neurological side of semantics is highly controversial. Researchers agree that the inferior frontal, inferior parietal , and temporal cortex are all involved in processing semantic memory however the exact involvement of the specific areas is not necessarily agreed upon [84].

1.2.2 Motivation & Challenges

Various studies have been carried out to explore brain encoding and decoding mechanisms when a stimulus is present, as detailed next. For visual stimuli, studies have shown that is feasible to discriminate and reconstruct images using patterns of neural activity, mainly found in the visual cortex [12, 13, 14, 15, 16], the part of brain responsible for visual information processing. Other works have demonstrated the relationship between cognitive perception and speech [17, 18]. Regarding textual stimuli, researchers have shown distributed semantic maps of words are present in our brains [10, 29]. Lexical semantics are based on the assumption that similar words appear in similar contexts [19]. Based on that assumption, two different approaches for building semantic models have been proposed. The first approach is to encode word semantics, by applying dimensionality reduction of context-word occurrence matrix which was computed using large corpora [20, 21]. The second approach replaces thse "counting" by predictive models [22] based on neural networks [23, 24, 25, 26, 27]. Counting models calculate and weight context vectors, while predictive models learn word vectors by guessing the context in which these words tend to appear.

Linguistic resources have helped neuroscientists map semantics onto the brain. This thesis attempts to answer whether brain and corpus-based representations of semantics also complementary. The answers to such question bring new insights into the role that brain and corpus data can play in the study of semantics.

In pursuance of enriching such lexical semantic models with cognitive information, as well as discovering the cognitive representation of word semantics, a number of studies have attempted to examine the mapping between semantic representation of computational and cognitive models. In prior work, it has been shown that semantic of words are related to activation potentials in regions of the brain and that decoding between neural activations and semantic content [7, 28, 29, 30, 31] is possible. Furthermore, neural activations are shown to have predictive power with respect to semantics at the word [7, 8] and sentence [32, 33] level. Computational studies that aim to explore the influence of neural activations in word representations have shown that by incorporating neural activations when training lexical semantic models can improve their generalization ability despite the small amount of neural activation data used [34, 35]. These works show that a strong relationship exists between computational semantic models and neural representations. However, it remains to be seen how cognitive semantic representations, including localized neural activation patterns can help improve the performance of computational semantic models, especially for complicated classification and recognition tasks.

Motivated by the aforementioned studies that show correlation between localized neural activations and word semantics, we propose a computational model for semantic similarity that utilizes predicted neural activations learned from a small set of concrete nouns. The proposed model is applied to a variety of natural language processing tasks. The neural activation prediction model used here for lexical expansion is that proposed in [7]. In our list of experiments, we first compare the performance of the proposed neural activation model for a concrete noun semantic similarity task and show that for certain word pairs it outperforms the state-of-the-art. Then we evaluate the performance of neural activation vectors for a word classification, sensory modality (sense) classification and textual entailment task. The fusion of neural and traditional word embedding vectors are shown to outperform the state-of-the-art. To our knowledge, this is the first time brain imaging data are successfully used for the aforementioned tasks.

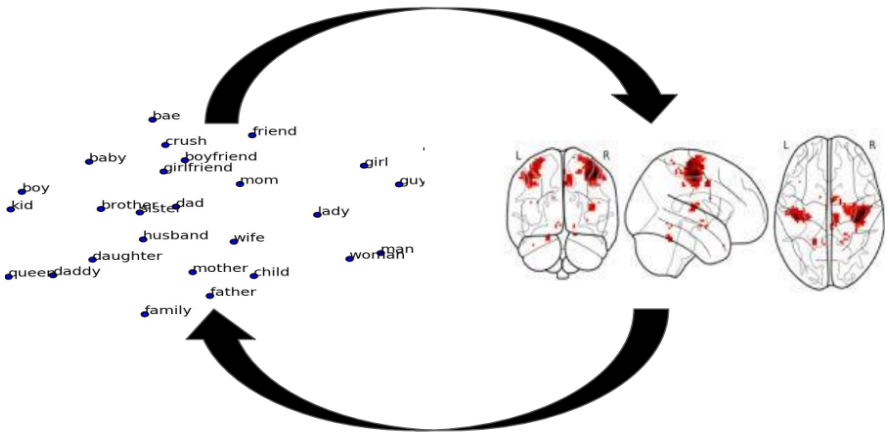


Figure 1.1: The two different spaces of neural and word representations that we are trying to bridge in the Chapter 4. Computational integration of brain information in word representations could help us encode word semantics better.

1.3 Multiple Representations

Most word embedding models typically represent each word using a single vector, which makes these models in discriminative for ubiquitous homonymy and polysemy. In order to enhance discriminativeness, we employ latent topic models to assign topics for each word in the text corpus, and learn topical word embeddings(TWE) based on both words and their topics. In this way, contextual word embeddings can be flexibly obtained to measure contextual word similarity. We can also build document representations, which are more expressive than some widely-used document models such as latent topic models. In the experiments, we evaλuate the TWE models on two tasks, contextual word similarity and text classification. The experimental results show that our models outperform typical word em-bedding models including the multi-prototype version on contextual word similarity, and also exceed latent topic models and other representative document models on text classification. Most word embedding methods assume each word preserves a single vector, which is problematic due to homonymy and polysemy. Multi-prototype vector space models [48] were proposed to cluster contexts of a word into groups, then generate a distinct prototype vector for each cluster. Following this idea, proposed multi-prototype word embed-dings based on neural language models [23]. Despite of their usefulness, multi-prototype word embed-dings face several challenges:

- These models generate multi-prototype vectors for each word in isolation, complicated correlations among words as well as their con-texts.

- In multi-prototype setting, contexts of a word are divided into clusters with no overlaps. In reality, a word's several senses may correlate with each other, and there is not clear semantic boundary between them

1.3.1 Cross Topic Semantics

Recent approaches that produce multiple distributed representations per word make use of topic modeling techniques as discussed in Chapter 2. Current word representation learning models encode the semantic and syntactic information of words adopting the distributional hypothesis [19]. Those approaches are ubiquitous in Natural Language Processing (NLP), achieving impressive results in tasks such as information retrieval [85], sentiment analysis [86] and machine translation [66, 87]. However, such models learn single point representations, which cannot capture the distinct meanings of polysemous words (e.g., *cancer* or *view*). This leads to conflated word representations of diverse contextual

semantics. Thus, the creation of multi-sense embeddings, which encode different word meanings in the semantic space can help us to improve natural language understanding.

Employing multiple embeddings can improve performance in downstream NLP tasks, such as part-of-speech tagging, semantic relation identification [51] and machine translation [60]. Distributional Semantic Models (DSMs) with multiple representations per word have been proposed in the literature, based on clustering local contexts of individual words [48, 52, 50] and on usage of external lexical resources [88, 89]. Furthermore, alternative approaches were proposed which associated different word embeddings with different topics [90, 53, 54].

1.3.2 Motivation & Challenges

Unsupervised models for learning multiple embeddings rely on locally clustering the contexts for each individual word [48, 49, 50, 51, 52]. This locality assumption ignores complicated correlations among words, that is with their contexts. Furthermore, contexts are clustered without overlaps which is not so accurate as several meanings may correlate with each other. The topic based approaches do not model clearly topic and word interactions [53] or make static assumption for such relationships [54]. Other relevant approaches that utilize topic-based corpora, do not perform both the necessary alignment of word embedding models trained in different topic spaces and their sense adaptation yielding evaluation results not directly comparable to the standard evaluation methods in contextual similarity task [90]. Supervised approaches[56, 57, 88, 89], rely on external resources and lexicons. Consequently, they are restricted to languages where such lexical resources exist, depend on the lexical coverage and quality of such resource and cannot capture the semantic shift of words over time [91].

With the proposed research, we intend to explore methods for creating multiple representations of words in different topic spaces. In particular, we propose a novel approach for learning cross-topic word embeddings, by employing weakly supervised or unsupervised methods. As a second step, we aim to investigate methods for aligning cross-topic word vectors, by grouping them together into sense clusters, in order to obtain smoothed word sense representations, thus reducing the noise from the sparse training data. Finally, the aforementioned approach is totally unsupervised depending only on the quality and diversity of the initial corpus.

1.4 Thesis Organization

The remainder of this thesis is organized as follows. Note that each chapter can be considered self-contained in terms of notation but methods, experiments and conclusions may be drawn from the results of the former.

- Chapter 2 outlines the machine learning background theory to follow the methods and the content of the present thesis. Specifically, a general introduction in machine learning is presented explaining the learning process. Then, traditional classification models which are used in this work are explained. After the traditional models, neural networks and especially recurrent neural networks (RNNs), long short-term memory units (LSTMs) and attention mechanism are described. Finally, we describe two major dimensionality reduction methods.

- Chapter 3 presents the natural language processing background needed to understand this thesis. After briefly presenting the basic hypothesis between the vast majority of language representations models [19], the most popular single representation models are presented. Then, we briefly discuss about multiple representations models. Next, we extensively explain the topic modelling algorithm we used in this work. Moreover, we present some previous work on transformation between different spaces. Finally, the needed connection between brain and semantics is presented.

- Chapter 4 includes research work involved the computational examination of brain representations (fMRI) and whether they can be combined along with current word embeddings. We

used an already proposed approach to predict neural activations—intensity values of voxels of an fMRI image—for a given word. Next, our proposal of a similarity model which utilizes neural activations showed that neural activations differentiate from word embeddings especially in highly similar and dissimilar words. In addition, a simple fusion schema of neural activations along with word embeddings in various NLP tasks (taxonomy creation, human sense classification, textual entailment) increased performance in both of the cases of traditional and "modern" ML (clustering, neural networks) models.

- Chapter 5 Traditional DSMs conflate multiple word senses in one representation. On the other hand, TDSMs incorporate the assumption that the meaning of a word changes in different topic domains. We focus on alignment of word embeddings from different topic-specific spaces to a common space by hypothesizing that monosemous words preserve their relative distances. Preliminary contextual semantic similarity experimentation has shown that topic-based representations fusion improves current results.

- Chapter 6 includes conclusions inferred from the thesis and outlines future directions that could be followed.

Chapter 2

Machine Learning Background

2.1 Notation

We denote real, integer and natural numbers as $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$, respectively. Scalars are represented by no-boldface letters, vectors appear in boldface lowercase letters and matrices are indicated by boldface uppercase letters. All vectors are assumed to be column vectors unless they are explicitly defined as row vectors. For a vector $\mathbf{z} \in \mathbb{R}^n$, $\|\mathbf{z}\|_1 = \sum_{i=1}^{n} |z_i|$ is its ℓ_1 norm and $\|\mathbf{z}\|_2 = \sqrt{\sum_{i=1}^{n} z_i^2}$ is its ℓ_2 norm, where z_i is the ith element of $\mathbf{z}$. By $\mathbf{A} \in \mathbb{R}^{n \times m}$ we denote a real-valued matrix with n rows and m columns. Additionally, the jth column of the matrix $\mathbf{A}$ and its entry at ith row and jth column are referenced as $\mathbf{A}_j$ and A_{ij}, respectively. The trace of the matrix $\mathbf{A}$ appears as $tr(\mathbf{A})$ and its Frobenious norm as $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2}$. The square identity matrix with n rows is denoted as $\mathbf{I}_n \in \mathbb{R}^{n \times n}$. Finally, $\mathbf{X}^{(k)}$ refers to the estimate of a variable $\mathbf{X}$ at the kth iteration of an algorithm. We define the conditional probability of the event Ω_1 given that the event Ω_2 has already happened with: $p(\Omega|\Omega')$. We denote with $\mathbf{a}||\mathbf{b}$ the concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$. We define the element-wise multiplication for two vectors $\mathbf{a}$ and $\mathbf{b}$ with $\mathbf{a} \odot \mathbf{b}$.For the matrices $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^{n \times m}$ we indicate their Hadamard(element-wise) product as $A \otimes B$.

2.2 Machine Learning Introduction

Machine learning (ML) is the scientific field that can be seen as a subset of Artificial Intelligence(AI) and studies algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. Machine learning algorithms are mathematical algorithms that are based on sample data, known as ”training data”, in order to make predictions in a desired range of values. In general, they seek to provide knowledge to computers through data, observations and enable them to interact with the real world. That acquired knowledge allows computers to correctly generalize to new settings without human intervention.

ML tasks can be categorized in several broad categories. These categories are mainly discriminated by the way learning is achieved and how the system receives feedback during the learning procedure is given to the system developed. Two of the most widely adopted ML methods are *supervised* learning which uses labelled output data in order to refine the systems knowledge and *unsupervised* learning which exploits a set of data which contains only inputs and no desired output labels to learn their structure.

However, there are some other categories such as reinforcement learning, active learning –which are not part of the thesis–, semi-supervised and weakly supervised learning which make use of labelled output data in smaller parts of the learning process.

2.2.1 Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. A supervised learning algorithm learns a mapping function from the training data and the learned function allows for the algorithm to correctly determine the class

labels for unseen instances. Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. Supervised learning problems can be categorized into **regression** and **classification** problems.

Classification is when the output space is discrete i.e. the output variable is a category. Regression is when the output space is continuous i.e. the output variable is a real value.

2.2.2 Unsupervised Learning

In other machine learning problems, the training data consists of a set of input vectors without any corresponding target values. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. It could be contrasted with supervised learning by saying that whereas supervised learning intends to infer a conditional probability distribution conditioned on the labels of input data, unsupervised learning intends to infer an a priori probability distribution. One subset of unuspervised learning is *clustering*. A clustering problem is where you want to discover the inherent groupings in the data. Another class of unsupervised tasks is association problems. In *association* problems we want to discover rules is that describe large portions of your data without the existence of labels.

2.3 Traditional Classification Methods

Any classification method uses a set of features or parameters to characterize each object, where these features should be relevant to the task at hand. We consider here methods for supervised classification as we have defined it in section. There are two phases to constructing a classifier. In the training phase, the training set is used to decide how the parameters ought to be weighted and combined in order to separate the various classes of objects. In the testing phase, the weights determined in the training set are applied to a set of objects that do not have known classes in order to determine what their classes are likely to be.

If a problem has only a few (two or three) important parameters, then classification is usually an easy problem. For example, with two parameters one can often simply make a scatter-plot of the feature values and can determine graphically how to divide the plane into homogeneous regions where the objects are of the same classes. The classification problem becomes very hard, though, when there are many parameters to consider. Not only is the resulting high-dimensional space difficult to visualize, but there are so many different combinations of parameters that techniques based on exhaustive searches of the parameter space rapidly become computationally infeasible. Practical methods for classification always involve a heuristic approach intended to find a "good-enough" solution to the optimization problem.

2.3.1 Linear Regression (LR)

In statistics, linear regression is a linear approach to modelling the relationship between a dependent variable and one or more independent variables. The case of one dependent variable is called simple linear regression while if more than one dependent variables exist, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Simple linear regression estimates exactly how much the dependent variable y will change when the independent variable x changes by a certain amount. With regression, we are trying to predict the y variable from x using a linear relationship (i.e., a line):

$$\boldsymbol{y} = b_0 + b_1 \boldsymbol{x} \tag{2.1}$$

where b_0 is known as the intercept (or constant), and the b_1 as the slope for x. The machine learning community tends to use other terms, calling Y the target and X a feature vector. Specifically for the case of multiple linear regression let:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1k} \\ 1 & x_{21} & \dots & x_{2k} \\ 1 & x_{31} & \dots & x_{3k} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{nk} \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_k \end{bmatrix} \tag{2.2}$$

According to the above notation, τhe linear regression model can be written in the form:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} \tag{2.3}$$

and b coefficients can be estimated as

$$\hat{b} = \arg\min_{b} ||\boldsymbol{y} - \boldsymbol{b}\boldsymbol{X}||_2^2 \tag{2.4}$$

using that minimizes the Sum of Squared Errors (SSE). Tikhonov regularization, named for Andrey Tikhonov, is the most commonly used method of regularization of ill-posed problems. In statistics, the method is known as *Ridge* regression, in machine learning it is known as weight decay. Ridge Regression is a remedial measure taken to alleviate multicollinearity amonst regression predictor variables in a model. Often predictor variables used in a regression are highly correlated. When they are, the regression coefficient of any one variable depend on which other predictor variables are included in the model, and which ones are left out. Ridge regression adds a small bias factor to the variables in order to alleviate this problem. In that case the b coefficients can be estimated as:

$$\hat{b}^{ridge} = \arg\min_{b} ||\boldsymbol{y} - \boldsymbol{b}\boldsymbol{X}||_2^2 + \lambda||b||_2^2 \tag{2.5}$$

where $y \in \mathbb{R}^n$, $x \in \mathbb{R}^{n \times p}$. Here, $\lambda \geq 0$ is a tuning parameter for controlling the strength of the penalty. When $\lambda = 0$, we minimize only the loss which may lead to overfitting and when $\lambda = \infty$ the coefficients are zeroed which leads to underfitting.

2.3.2 Support Vector Machines (SVMs)

In many machine learning problems feature vectors of different classes may be not linearly separable in the original space they live in. Presumably, one cannot easily find a hyperplane of the input feature space serving as a classification boundary for data belonging to each class of the training set. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space [92].

SVMs are trying to find maximum-margin hyperplanes in order to create these classification boundaries between the vectors of each class as it is depicted in Figure. Concretely, let a training set comprises N input vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ with corresponding target values $\boldsymbol{y}_1, \dots, \boldsymbol{y}_N$ where $y_i \in \{-1, 1\}$.

We are given l training examples $x_i, y_i,\ i = 1, \cdots, l\ ,$ where each example $x_i \in \mathbb{R}^d$), and a class label with one of two values ($y_i \in \{-1, 1\}$). Now, all hyperplanes in $\mathbb{R}^d$ are parameterized by a vector (w), and a constant (b), expressed in the equation:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2.6}$$

Given such a hyperplane (w,b) that separates the data, this gives the function

$$f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{2.7}$$

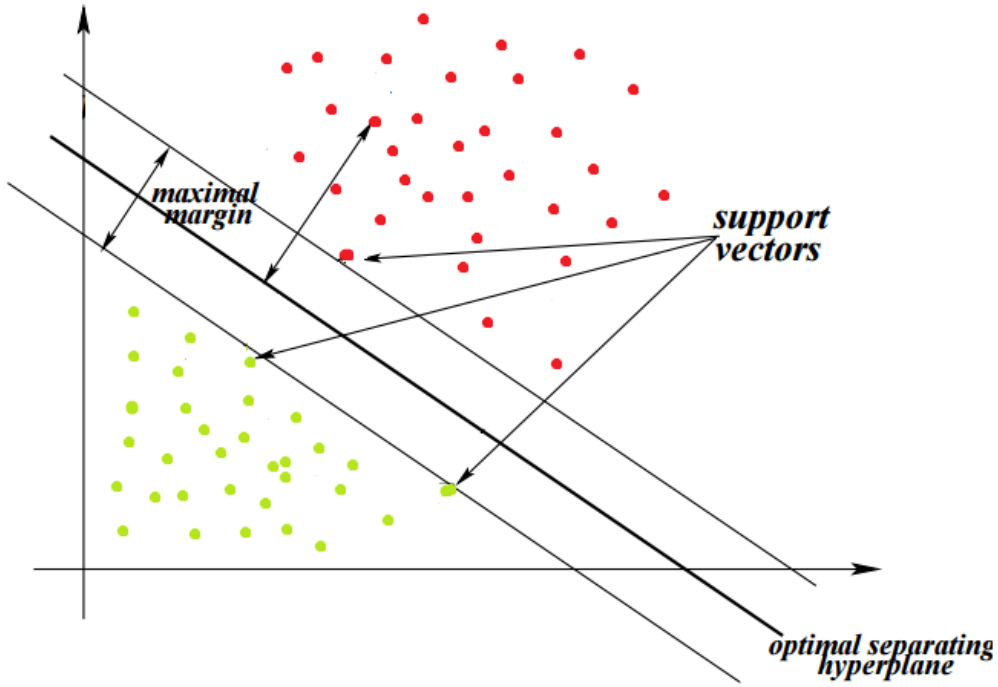


Figure 2.1: Example of binary classification of two linearly separable classes where each main parameter is explicitly indicated.

which correctly classifies the training data and other data it hasn't seen yet. So we define the *canonical hyperplane* to be that which separates the data from the hyperplane by a "distance" of at least[1] 1. That is, we consider those that satisfy:

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \geq 1 \quad \forall i \tag{2.8}$$

To obtain the geometric distance from the hyperplane to a data point, we must normalize by the magnitude of $\mathbf{w}$. This distance is simply:

$$d\Big((\mathbf{w}, b)\ ,\ \mathbf{x}_i\Big) = \frac{y_i(\mathbf{x_i} \cdot \mathbf{w} + b)}{\parallel \mathbf{w} \parallel} \geq \frac{1}{\parallel \mathbf{w} \parallel} \tag{2.9}$$

Intuitively, we want the hyperplane that maximizes the geometric distance to the closest data points.

From the equation we see this is accomplished by minimizing $||\mathbf{w}||$ (subject to the distance constraints). The main method of doing this is with Lagrange multipliers. We can define the matrix $(H)_{ij} = y_i y_j(\mathbf{x}_i \cdot \mathbf{x}_j)$, and introduce more compact notation. The problem is eventually transformed into:

$$\text{minimize:} \quad W(\alpha) = -\alpha^T \mathbf{1} + \tfrac{1}{2}\alpha^T H \alpha \tag{2.10}$$

$$\text{subject to:} \quad \alpha^T y = 0 \tag{2.11}$$

$$\mathbf{0} \leq \alpha \leq C\mathbf{1} \tag{2.12}$$

where α is the vector of l non-negative Lagrange multipliers to be determined, and C is a regularization term for configuring the penalty term of wrongly classified instances. In addition, from the derivation

[1] In fact, we require that at least one example on both sides has a distance of *exactly* 1. Thus, for a given hyperplane, the scaling (the λ) is implicitly set.

of these equations, it was seen that the optimal hyperplane can be written as:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{2.13}$$

The solution of constrained equation system(2.10, 2.11, 2.12) is given by Lagrange multipliers [93].

When a data set is not linearly separable, doesn't mean there isn't some other concise way to separate the data. To do this, we define a mapping $\mathbf{z} = \phi(\mathbf{x})$ that transforms the d dimensional input vector $\mathbf{x}$ into a (usually higher) d' dimensional vector $\mathbf{z}$.

Given a mapping $\mathbf{z} = \phi(\mathbf{x})$, to set up our new optimization problem, we simply replace all occurences of $\mathbf{x}$ with $\phi(\mathbf{x})$. Our problem (eq. 2.10) becomes:

$$\text{minimize:} \quad W(\alpha) = -\alpha^T \mathbf{1} + \tfrac{1}{2}\alpha^T H \alpha$$

with $(H)_{ij} = y_i y_j (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j))$ Then eq. 2.13 would be

$$\mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)$$

Any time a $\phi(\mathbf{x}_a)$ appears, it is always in a dot product with some other $\phi(\mathbf{x_b})$. That is, if we knew the formula (*kernel*) for the dot product in the higher dimensional feature space,

$$K(\mathbf{x_a}, \mathbf{x_b}) = \phi(\mathbf{x_a}) \cdot \phi(\mathbf{x_b}) \tag{2.14}$$

The matrix in our optimization would simply be $(H)_{ij} = y_i y_j (K(\mathbf{x}_i, \mathbf{x}_j))$. And our classifier $f(\mathbf{x}) = \text{sign}\Big(\sum_i \alpha_i y_i (K(\mathbf{x}_i, \mathbf{x})) + b\Big)$. We can easily extend the previous formulation of binary decision SVMs in multi-class problems by simply training separate binary classifiers for all the classes available in the training data and choose the one with the highest confidence.

2.3.3 Clustering

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

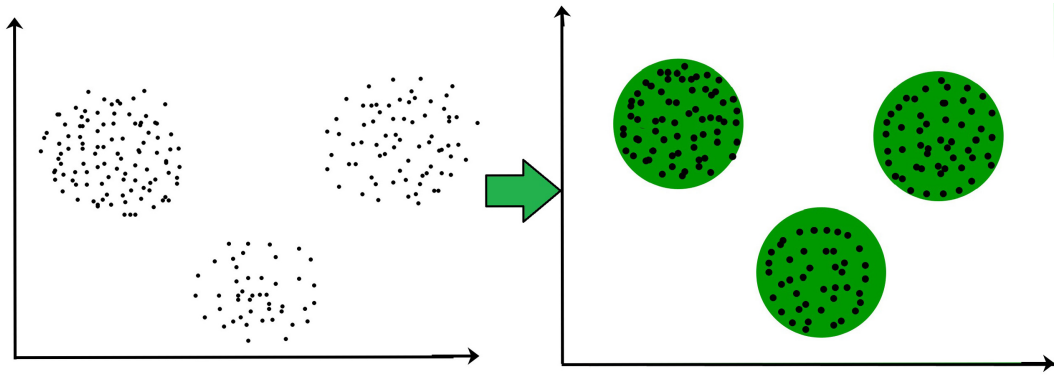


Figure 2.2: A clustering example.

Spectral Clustering

In spectral clustering, the data points are treated as nodes of a graph. Thus, clustering is treated as a graph partitioning problem. The nodes are then mapped to a low-dimensional space that can be easily segregated to form clusters. An important point to note is that no assumption is made about the shape/form of the clusters. Given an enumerated set of data points, the similarity matrix–which is symmetric– defined as $\boldsymbol{A}$, where $A_{ij} \geq 0$ represents a measure of the similarity between data points with indices i and j. The general approach to spectral clustering is to use a standard clustering method on relevant eigenvectors of a Laplacian matrix of A. There are many different ways to define a Laplacian which have different mathematical interpretations, and so the clustering will also have different interpretations. Next, we describe a widely adopted approach by [37]. Given a dataset of n points $x_1, ...x_n$ in $\mathbb{R}^d$ we form the affinity matrix $A \in \mathbb{R}^n \times n$ as:

$$A_{ij} = e^{-||s_i - s_j||/2\sigma^2} A_{ii} = 0 \tag{2.15}$$

Here, the scaling parameter σ^2 controls how rapidly the affinity A_{ij} decreases with the distance between x_i and x_j. Its value is determined we simply search over as the one that gives the tightest (smallest distortion) clusters as described in [37].

$$L^{\text{norm}} := I - D^{-1/2} A D^{-1/2}, D_{ii} = \sum_j A_{ij} \tag{2.16}$$

Then we find $e_1, e_2, ..., e_k$, the k largest eigenvectors of L, form the matrix $E = [e_1 e_2, ...e_k] \in \mathbb{R}_{nxk}$ where $e_i \in \mathbb{R}^n$ and normalize each row of E to have unit length. Treating each row of E as a point in $\mathbb{R}^k$, cluster them into k clusters via k-means or any other algorithm. As a final step we assign the original point x_i to cluster j if and only if row i of the matrix E was assigned to cluster j.

Gaussian Mixture Model (GMM)

GMM is a probabilistic model coming from the mathematical field of statistics [94]. In this approach we describe each cluster by its centroid (mean), covariance matrix, and the size of the cluster. Rather than identifying clusters by "nearest" centroids, we fit a set of Gaussians to the data. And we estimate Gaussian distribution parameters such as mean and Variance for each cluster and weight of a cluster. After learning the parameters for each data point we can calculate the probabilities of it belonging to each of the clusters.

So mathematically we can define Gaussian mixture model as mixture of Gaussian distributions. It is based on multi-variant normal distributions, which have n-dimensional random variables as described in 2.17.

$$f(X) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)} \tag{2.17}$$

Equation 2.17 use the random variable X, the expectation μ, the variance σ^2 and the variance matrix Σ respectively. A GMM itself is constructed from weighted sums of N gaussian densities as in 2.18.

$$p(x|\lambda) = \sum_{i=1}^{N} w_i g(x|\mu_i, \Sigma_i) \tag{2.18}$$

with

$x = (x_1, \ldots, x_n)$	d-dimensional data vector
$w = (w_1, \ldots, w_N)$	mixture weights
$g(x\|\mu_i, \Sigma_i); i, \ldots, N$	gaussian components

The mixture weights have to satisfy the constraint $\sum_{i=1}^{N} w_i = 1$. Each of the components is a multi-variant gaussian function as described in 2.19.

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\} \tag{2.19}$$

A complete GMM can be described by its parameters $\lambda = \{w, \mu, \Sigma\}$. Typically a GMM is trained by the Expectation-Maximization (EM) algorithm [95]. This algorithm is used to iteratively apply the Maximum-Likelihood estimation of the model parameters which tries to find the parameters, that maximize the likelihood of a GMM for a given training data with N datapoints. The GMM likelihood is typically described as (assuming an independence between the training vectors $X = \{x_i, \ldots, x_N\}$):

$$p(X|\lambda) = \prod_{j=1}^{N} p(x_j|\lambda) \tag{2.20}$$

As a direct maximization is not possible because of the non-linearity of the 2.20, the EM tries to iteratively estimate new model parameters $\bar{\lambda}$ based on a given model λ such that $p(X|\bar{\lambda}) \geq p(X|\lambda)$. One iteration step itself is typically split into the E- and M-step. In terms of GMM, during the E-step the probabilities of the generation of a datum x_i by the component i are computed. These probabilities are used to calculate the amount of datapoints assigned to a component by $n_i = \sum_{j=1}^{N} p_{ji}$. In the M-step the new parameters are calculated by

$$\mu_i = \sum_{j=1}^{N} \frac{p_{ij}x}{n_i} \tag{2.21}$$

$$\Sigma_i = \sum_{j=1}^{N} \frac{p_{ij}(x_j-\mu_i)(x_j-\mu_i)^T}{n_i} \tag{2.22}$$

$$w_i = \frac{n_i}{N} \tag{2.23}$$

These steps are repeated until a convergence threshold is reached.

2.4 Neural Networks

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. These techniques have enabled significant progress in the fields of sound and image processing, including facial recognition, speech recognition, computer vision, automated language processing, text classification. An Artificial Neural Network (ANN) is a biologically inspired computational model that is patterned after the network of neurons present in the human brain. The area of ANNs has originally been inspired by the goal of modeling biological neural systems, but has since diverged and become a matter of engineering and achieving good results in Machine Learning tasks. The basic computational unit of the brain is a *neuron*. Each neuron receives input signals from its *dendrites* and produces output signals along its (single) *axon*. Its axon connects via synapses to dendrites of other neurons. The dendrites carry the signal to the cell body where they all get summed. If the final sum is above a certain threshold, the neuron can fire, sending a spike along its axon. In the computational model, we assume that only the frequency of the firing communicates information. We thus model the firing rate of the neuron with an *activation function* f—usually sigmoid function σ in neural networks—, which represents the frequency of the spikes along the axon. In pursuance of learning complex non-linear functions, architectures that combine several artificial neurons have been designed and are called Multi-Layer Perceptrons (MLPs). Instead of MLPs, Feed-Forward Neural Networks (FFNNs) have been implemented, where each neuron connects with all neurons of the previous layer and there are no connections between the neurons of the same layer.

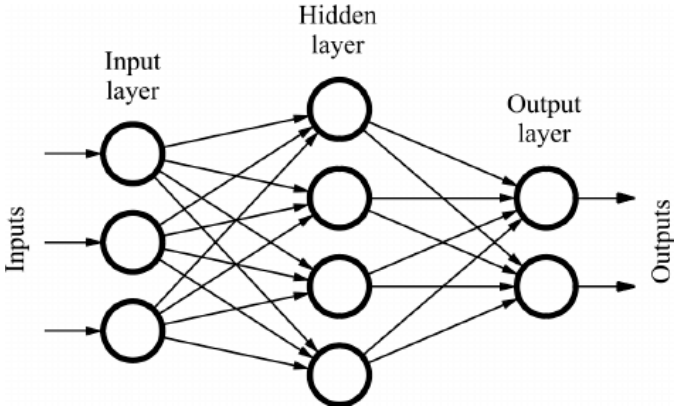


Figure 2.3: A three-layer Neural Network which was taken from [69]. It contains three inputs, one hidden layers of four neurons and one output layer with two outputs.

Each neural network is composed of an input layer, one or more hidden layer(s) and an output layer as depicted in Figure 2.3. Another basic component of neural networks is *activation function* which decides whether a neuron should be activated or not by introducing non-linearities to its output. Examples of such functions is sigmoid, tanh, ReLU and leaky-ReLU. The objective of a neural network is to minimize Equation 2.24.

However, the problem of *overfitting* arises when we have such an expressive model. We do not just fit the underlying function, but we in fact fit the noise as well. It naturally results in a model that does not generalize well on test data. *Regularization* is a way to mitigate this effect. To solve this problem we restrict the form of the solution. Specifically, the loss function takes the following form:

$$\hat{\Theta} = \arg\min_{\theta} \mathcal{L}(\theta) = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(f(\boldsymbol{x_i}; \theta), \boldsymbol{y_i}) + \lambda R(\theta). \tag{2.24}$$

λ is a value that has to be set manually, based on the classification performance on a development set (called *hyperparameter*). The regularizers *R* measure the norms of the parameter matrices and opt for solutions with low norms. Common norms are L_1 [96] and L_2. Another effective technique for preventing overfitting is *dropout* [97, 98].

Finally, to train the model, we need to solve the optimization problem in Equation 2.24. A common solution is to use Stochastic Gradient Descent (SGD) [99]. A number of alternate optimization algorithms have been introduced to ensure convergence. Currently, optimization techniques with automatic regulation of the learning rate are used such as Adagrad [100], Adadelta [101] and Adam [102]. To minimize the cost function of a given neural network using the optimal set of values for θ , we need to compute the gradient. Efficient gradient calculation was introduced with *backpropagation* algorithm [103, 104]. Backpropagation methodically computes the derivatives of a complex expression using the chain-rule, while caching intermediary results. The gradients are indicative of reaching the minimum of the loss which is neural networks training objective.

2.4.1 Recurrent Neural Networks (RNNs)

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence. The core idea behind RNNs is to process information sequentially. In a traditional neural network we assume inputs and outputs are independent of each other. They are called recurrent because they perform the same task for every element of a sequence, depending the output on previous inputs. Thus, they demonstrate the ability to have a "memory" which captures the information calculated so far. They are particularly useful when modeling audio and text modalities where the underlying time dependencies are inherent in the nature of the input data [105].

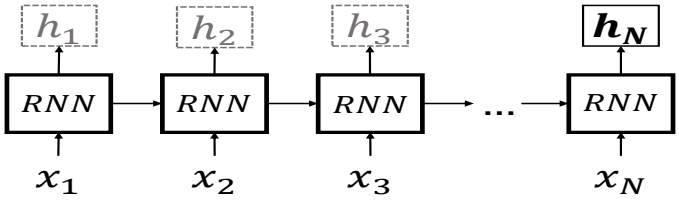


Figure 2.4: Regular unrolled RNN.

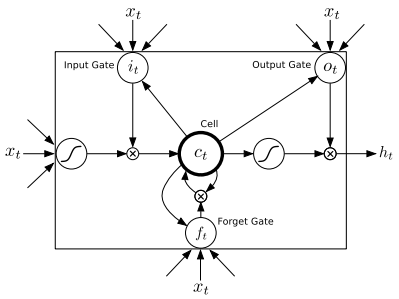


Figure 2.5: LSTM: Learn long term dependencies by asserting control over what goes in and out of memory cells[70].

As it can be observed from Figure 2.4, RNNs have nodes organized into successive "layers". Given the input $\boldsymbol{x}_{t\,t=0}^{N}$ where N is the length of the input sequence, processes every input vector at time step x_t from input sequence and outputs h_t (hidden state) and forwards both to the next step. Formally, at each time step t, the equations that describe the function of the RNN are:

$$h_t = q(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = r(W_{hy}h_t + b_y)$$

where y_t is the output vector at time step t, b_h is the bias for h, b_y is the bias for y and q, r are the activation functions for x and h respectively. Finally there are three parameter matrices who are notated as W_{xh} (input-to-hidden weights), W_{hh} (hidden-to-hidden), and W_{hy} (hidden-to-output).

2.4.2 Long Short Term Memory (LSTM) unit

Theoretically, RNNs are able to model arbitrarily long dependences between the input data. However, because of the nature the training algorithm for neural networks–backpropagation–the long-term dependencies of the input sequence yield the problem of vanishing or exploding gradients. Precisely, the recurrent topology of the network which implies the computation of the gradient and its flow over multiple timesteps gradients vanish or explode due to the finite-precision calculations when the error is tried to be propagated backwards. LSTMs [106] is one way to tackle this problem. The core functionality which controls the magnitude of gradients is its forget gate. In Figure 2.5 the block diagram of an LSTM cell is displayed.

The core components of the LSTM architrecture are the *forget,input,output* gates. Forget gate controls the informational flow from the networks memory. Intuitively, determines the portion of the information to be kept. Input gate controls the informational flow for the input vector x_t at the current timestep. The output gate determines the final activation of the cell h_t at the current timestep from its current precomputed state c_t. Formally, given a sequence $x_1, ..., x_n$ of vectors of an input sequence,

with inputs h_{t-1} and c_{t-1}, h_t and c_t for x_t are computed as follows:

$$
\begin{aligned}
i_t &= \tanh(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
j_t &= \sigma(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
c_t &= f_t \otimes c_{t-1} + i_t \otimes j_t \\
h_t &= \tanh(c_t) \otimes o_t
\end{aligned}
$$

Forget gate(f_t) This gate decides the proportion of the information which should be discarded by the use of the sigmoid function. In this way, the output will be a number between zero and one which the former corresponds to forget the input of the previous activation. In contrast, a value of 1 in this gate represents that the information of the previous activation h_t−1 alongside with the information from the current input vector x_t would be fully considered for the computation of the state of this LSTM.

Input gate (i_t) The input gate controls the information which will flow from the activation of the previous timestep h_{t-1} alongside with the information from the current input vector x_t when we are trying to update the current state weights.

Cell state (c_t) To calculate the updates on for the current state we are adding the information which has been controlled by the aforementioned gates (input and forget gates). First, the cell state is pointwise multiplied by the forget vector, an operation which decides the values that will be updated. Next, the output from the input gate is pointwise added.

Output gate (o_t) In order to compute the output at the current timestep t, we need to combine the precomputed state vector c_t after a nonlinear function $tanh(\cdot)$ is applied as well as information from the input vector and the activation from the previous timestep at the output gate multiplier.The output gate decides what the next hidden state should be. The new cell state and the new hidden is then carried over to the next time step.

2.4.3 Bidirectional LSTM

As mentioned above, RNNs capture information about the sequential data they have seen until time step t and encode it in their hidden state. However, it is also possible to acquire more information by reading a given sequence backwards, in order to make more accurate predictions. So, a bi-directional RNN operations are described next.

We encode the input sequence from the beginning to the end (forward RNN) and also in reverse (backward RNN). We then combine the hidden states of the two RNN layers in order to find the hidden state for each time step. Specifically, we separately compute the hidden state of the forward RNN $\overrightarrow{h_t}$ at time step t as well as the corresponding hidden state of the backward RNN $\overleftarrow{h_t}$ and concatenate them in order to compute the final hidden state at each timestep. To this end, the hidden state at time step t is simply the concatenation of the two vectors:

$$h_t = \overrightarrow{h_t} || \overleftarrow{h}_{T-t}$$

The same applies for all the $T + 1$ time steps of the input sequence.

2.4.4 Attention Mechanism

The basic idea behind attention is that not all vectors in a given sequence contribute to the same degree to the meaning that is expressed in the overall input. So, the model should not use all vectors equally to make a prediction, but focus on the parts of the input that contain the most relevant information

for a given task. To implement this approach, an attention mechanism [70, 107] can be used in order to find the relative importance of each input vector of a sequence. To amplify the contribution of the most informative vectors, we assign a weight a_i to the hidden step that corresponds to each vector h_i. We compute the fixed representation r of the whole input sequence, as the weighted sum of all hidden states.

$$e_i = tanh(W_h h_i + b_h), \quad e_i \in [-1, 1] \tag{2.25}$$

$$a_i = \frac{exp(e_i)}{\sum_{t=1}^{T} exp(e_t)}, \quad \sum_{i=1}^{T} a_i = 1 \tag{2.26}$$

$$r = \sum_{i=1}^{T} a_i h_i \tag{2.27}$$

where W_h and b_h are the attention layer's weights. A simple visualization of the mechanism is depicted in Figure 2.6.

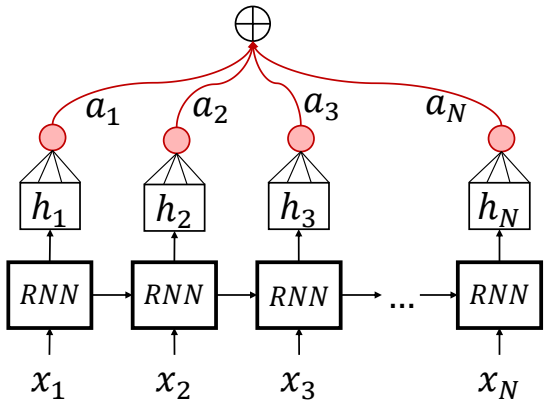


Figure 2.6: A high-level image of an RNN with attention.

2.5 Dimensionality Reduction Methods

Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables and can be categorized into feature selection and feature extraction. The first approaches try to find a subset of the original variables while the latter transforms the data in the high-dimensional space to a space of fewer dimensions. It can help by removing multicollinearity which improves the interpretation of the parameters of the machine learning model, data may become easier to visualize when reduced to very low dimensions such as 2D or 3D (especially in tasks we are interested in create representations of data) and also avoids the curse of dimensionality.

2.5.1 Principal Components Analysis

Principal component analysis can be used to analyze the structure of a data set or allow the representation of the data in a lower dimensional dataset (as well as many other applications).

Let $\{\vec{x}_i\}$ be a set of N column vectors of dimension D. Define the scatter matrix S_x of the data set as

$$S_x = \sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_x)(\vec{x}_i - \vec{\mu}_x)^T$$

where $\vec{\mu}_x$ is the mean of the dataset

$$\vec{\mu}_x = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i$$

The d largest principle components are the eigenvectors $\vec{w}_i$ corresponding to the d largest eigenvalues. d can be chosen arbitrarily with $d < D$. The eigenvectors of **S** can usually be found by using singular value decomposition. The dominant eigenvectors describe the main directions of variation of the data. The d eigenvectors can also be used to project the data into a d dimensional space. Define

$$\mathrm{W} = [\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_d]$$

The projection of vector $\vec{x}$ is $\vec{y} = \mathrm{W}^T\vec{x}$. The corresponding scatter matrix S_y of the vectors $\{\vec{y}_i\}$ is:

$$\mathrm{S}_y = \mathrm{W}^T\mathrm{S}_x\mathrm{W}$$

The matrix W maximizes the determinant of S_y for a given d. An intuitive illustation of how PCA works is given in Figure 2.7.

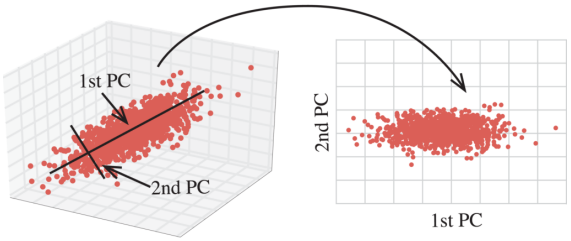


Figure 2.7: PCA applied in a synthetic dataset reducing its dimension from 3 to 2. The two primary components—dimensions with higher varianve— are clearly outlined.

2.5.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

Algorithm t-SNE is a non linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets [108]. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability. Let us for simplicity to take the example of mapping data point from a d-dimensional space to the $\mathbb{R}^2$.

A data point in the original d-dimensional space is defined as x_i. A mapped point is a point $y_i \in \mathbb{R}^2$. The positions of the mapped points is chosen to conserve the structure of the data. More specifically, if two data points are close together, we want the two corresponding mapped points to be close too. Hence, let $|x_i - x_j|$ be the Euclidean distance between two data points, and $|y_i - y_j|$ the distance between the mapped points. A conditional similarity between the two data points is firstly defined as:

$$p_{j|i} = \frac{e^{-|x_i - x_j|^2/2\sigma_i^2}}{\sum_{k\neq i} e^{-|x_i - x_k|/2\sigma_i^2}} \tag{2.28}$$

Equation 2.28 measures how close x_j and x_i, considering a Gaussian distribution around xi with a given variance σ_i^2. This variance is chosen such that points in dense areas are given a smaller variance than points in sparse areas as described in [108]. Next, the similarity is defined as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{2.29}$$

In the same manner, a similarity matrix for the mapped points is defined as:

$$q_{ij} = \frac{f(|x_i - x_j|)}{\sum_{k\neq i} f(|x_i - x_k|)} \text{with} (x) = \frac{1}{1+x^2} \tag{2.30}$$

Whereas the data similarity matrix (p_{ij}) is fixed, the mapped similarity matrix (q_{ij}) depends on the mapped points. We want is for these two matrices to be as close as possible. That corresponds to minimizing the Kullback-Leiber divergence between the two distributions p_{ij} and q_{ij}. This measures the distance between our two similarity matrices. To minimize this score, we perform a gradient descent.

Chapter 3

Natural Language Processing Background

Natural Language Processing (NLP) is a field of computer science, artificial intelligence (also called machine learning), and linguistics concerned with the interactions between computers and human (natural) languages. It is the process of a computer extracting meaningful information from natural language input and/or producing natural language output. It is analysis of human language based on semantics and various parsing techniques [109]. The goal of NLP is to identify the computational machinery needed for an agent to exhibit various forms of linguistic behavior. It also design, implement, and test systems that process natural languages for practical applications. NLP is a discipline between linguistics and computer science which is concerned with the computational aspects of the human language faculty. The main task of it is to construct programs in order to process words and texts in natural language.

3.1 Distributional Hypothesis

Distributional Semantics embraces a wide range of approaches based on the distributional hypothesis, in an attempt to capture meanings of linguistic entities (words, phrases) from their usage in language. This hypothesis is often described by the famous quote "You shall know a word by the company it keeps" [110], which presumes a correlation between distributional similarity and meaning similarity. The direct implication of this hypothesis is that two words that are considered to be semantically similar are expected to occur in similar *contexts*, and vice-versa. The conceptualization of this hypothesis, requires a definition of what constitutes a $context$ of a target word defined in a mathematical framework. In this work, we follow the commonly used definition of a context as the set of words existing within a window around each occurrence of the target word.

3.2 Language Representations

3.2.1 Count-based Models

In the simplest case of traditional DSMs, each dimension captures statistical information for context items observed to co-occur no further than a fixed distance c from the target's instance. This simple counting method results in a co-occurrence matrix, where the components of each vector can be interpreted as weights denoting the strength of the relationship between the target and the respective context word. It can be observed though, that raw co-occurrences are not a reliable source of information for revealing meaning correlation, as frequent yet uninformative context words tend to co-occur with most of the target words at a high rate.

In order to mitigate this phenomenon, non-linear operations can be applied on the co-occurrence matrix in an attempt to downplay the role of highly frequent words. Typically, the most widely used transformation is the Positive Pointwise Mutual Information (PPMI) defined by [111] as:

$$\text{PPMI}(word_i, word_j) = \max(0, \text{PMI}(word_i, word_j)) \tag{3.1}$$

$$\text{PMI}(word_i, word_j) = \log_2 \frac{P(word_i) \cap P(word_j)}{P(word_i)P(word_j)}. \tag{3.2}$$

In the above relation the numerator gives us information about how often the two words occur together, while the denominator tells us how often we would expect the two words to co-occur assuming they occurred independently, so their probabilities could just be multiplied (see example in Table 3.1).

	player	court	Athenian	cart	a
basketball	485	410	1	45	1053
democracy	1	2	350	10	375

	player	court	Athenian	cart	a
basketball	0.21	0	0	0	0.01
democracy	0	0	1.93	0	0

Table 3.1: Example of co-occurence matrix, extracted using raw counts (upper table), and after PPMI transformation (lower table)

Since count-based methods calculate the co-occurrence matrix for all words, they result in sparse high-dimensional representations —that is, most of the components of the vectors are zero— as a word is often semantically related to a small percentage of context instances. Commonly, dimensionality reduction is applied to the large matrix (in this case, the PPMI-weigthed co-occurrence matrix) in order to lessen the noise and reduce the sparsity of the vector space. The basic idea is to generate a lower-rank approximation of the original matrix, while in parallel retain the relations between the vectors. The resulted lower dimensional space is represented by the most important dimensions of the data set, along which most variation happens. The most popular method to generate matrix approximations of any given rank k is Singular Value Decomposition (SVD) [112], based on extracting the singular values of the initial matrix. An abstract scheme that summarizes the steps of creating a count-based DSM in depicted in 3.1.

Figure 3.1: Abstract representation of count-based Distributional Semantic Models procedure.

3.2.2 Word Embeddings

The key idea behind the unsupervised word vectors is that one would like the embedding vectors of "similar" words to have similar vectors. While word similarity is hard to define and is usually

very task-dependent, current approaches derive from the *distributional hypothesis* [19], stating that words are similar if they appear in similar contexts. Different methods all create supervised training instances in which the goal is to either predict the word from its context, or predict the context from the word. Perhaps the most important set of pretrained embedding vectors is *word2vec*. Word2vec is an approximation of language modeling, applied to a fixed word window.

Word2vec [25]
Word2vec is a shallow, two-layer neural network which is trained to reconstruct linguistic contexts of words. It takes as its input a large corpus of words and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. Word2Vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text.

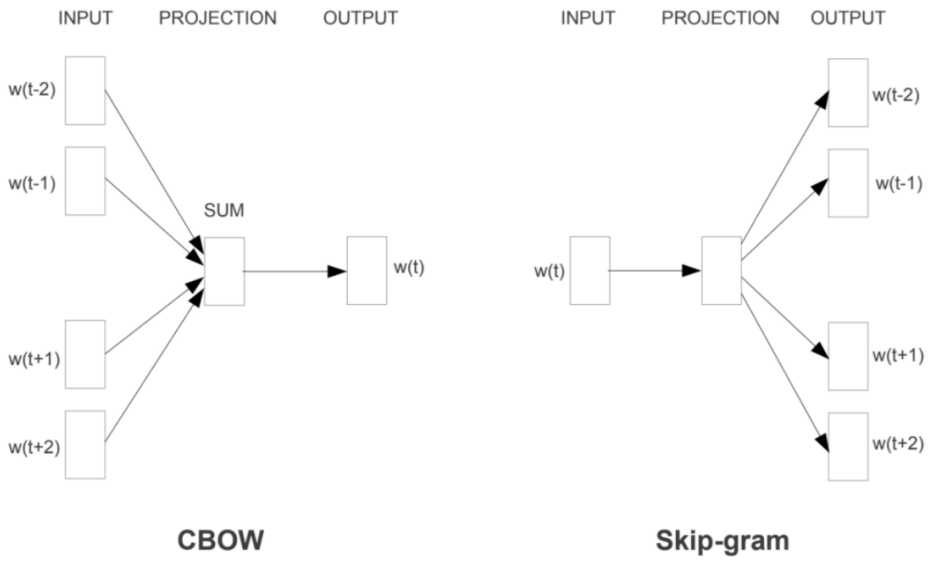


Figure 3.2: Word2vec training models. Taken from [25]

Given enough data, usage and contexts, word2vec can make highly accurate guesses about a word's meaning based on past appearances. Those guesses can be used to establish a word's association with other words (e.g. "king" is to "man" what "queen" is to "woman"), or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management.

Word2Vec has two forms, the *Continuous Bag-of-Words* (CBOW) model and the *Skip-Gram* model, as illustrated in Figure 3.2. When the feature vector assigned to a word cannot be used to accurately predict that word's context, the components of the vector are adjusted. Each word's context in the corpus is the *teacher* sending error signals back to adjust the feature vector. The vectors of words judged similar by their context are nudged closer together by adjusting the numbers in the vector.

- *Continuous Bag of Words (CBOW)*

  Supposing we want to predict word w_i, the input to the model could be $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$, the preceding and following words of the target word. The output of the neural network will be w_i. So, the CBOW model can be thought as learning word embeddings by training a model to predict a word given its context.

- *Skip-Gram*

  This model is the opposite of CBOW, as in this case the input of the model is w_i and the output would be $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$. The task, therefore, is to learn word embeddings by training a model to predict context given a word.

Even though the distributional hypothesis offers an appealing platform for deriving word similarities by representing words according to the contexts in which they occur, it has some inherent limitations, which should be taken into account when using the derived representations. The most important one, which has been largely examined in this thesis, is the lack of context.

Lack of context
The distributional approaches aggregate the contexts in which a term occurs in a corpus. The result is a context-free, or else *context-independent* word representation. An obvious problem that occurs is that polysemous words (words with obvious multiple senses) cannot be modeled properly. For example, a *bank* may refer to a financial institution or to the side of a river, a *star* may be an abstract shape, a celebrity or an astronomical entity, etc. By assigning the same vector to all the senses of a given word, language cannot be modeled in its complex form, as the meaning of numerous words evades.

Window of surrounding words
Another limitation comes from learning embeddings based only on a small window of surrounding words, sometimes words such as *good* and *bad* share almost the same embedding [113], which is problematic if used in tasks such as sentiment analysis [114]. At times these embeddings cluster semantically similar words which have opposing sentiment polarities. This leads the downstream model used for the sentiment analysis task to be unable to identify this contrasting polarities leading to poor performance.

3.3 Multiple-prototype representations

A large majority of current research on distributional semantics relies solely on models where each word is uniquely represented by one point in the semantic space. From a linguistic perspective, these models can not precisely capture the meaning of a polysemous word, resulting in a conflated representation of its diverse contexts. The problematic nature of single-prototype models could be better understood in the following two examples, which present two different contextual occurrences of the word *bank*.

- "...my friend and I are walking along the ***banks*** of the river when..."
- "...I need to go to the ***bank*** to withdraw money today in order to...."

Here, the inferred meanings of the word python are totally different in the two contexts referring in the edges of a river and financial instutition respectively. In pursuance of making these distinctions feasible in NLP, we need to account for polysemy in our models and turn the single-prototype representations to multiple-prototype representations. In the following sections we group the methods that assign multiple representations per word into two broad categories: *unsupervised* models induce multiple representations without leveraging external semantic lexical resources such as lexicons or large word databases, while *supervised* models rely on knowledge-based approaches.

3.3.1 Predict-Based Models

Fixed number of prototypes per word

[48] were the first to introduce multiple-prototype representations for lexical semantics. Motivated by the *distributional hypothesis*, they collected local contexts for each target word (as a vector formed by collecting frequency statistics in a fixed window around it) and applied clustering on them, with the

number of clusters being the single parameter of the model. The centroids of the created clusters were used in order to create a set of "sense-specific" vectors for each target word (Figure 3.3). Following the clustering approach, [49] proposed a recurrent neural network that incorporated both global and local context to learn multiple dense, low-dimensional embeddings. Again the number of possible senses corresponding to each word coincided with the fixed number of clusters.

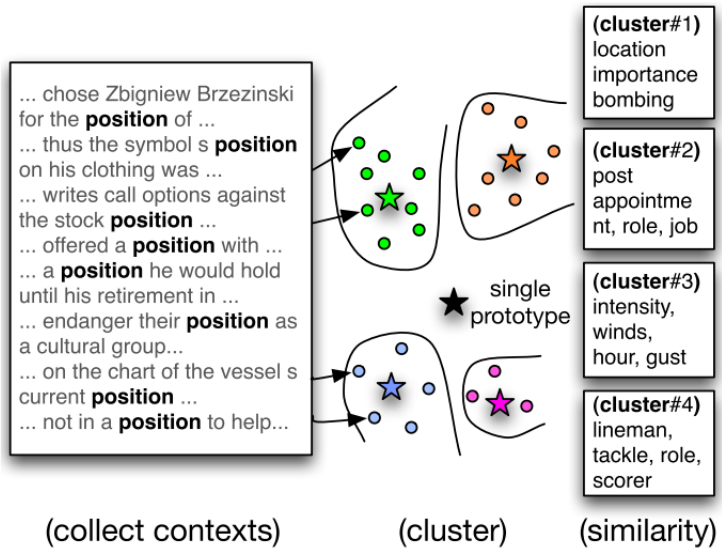


Figure 3.3: Overview of the multi-prototype approach using contextual clustering.

A probabilistic framework was later introduced by [52] who extended the Word2Vec model via representing the probability of a context word given the target word as a finite mixture of the prototypes of the target word. Using this framework, they designed an Expectation-Maximization algorithm to learn multiple embeddings, where the number of senses attributed to each word constituted a predetermined design decision.

Despite the fact that models with a fixed number of prototypes per word established the first attempts to provide vector representations that integrated the polysemic nature of words, more recent approaches provide more flexible solutions to the problem. Their flexibility is attributed to the fact that the real number of senses for words differs according to their polysemy degree (note that some words have only one sense, a.k.a. monosemous words) and changes through time as the evolution of language causes the creation of new senses (e.g., word *python* as a programming language).

Adaptive number of prototypes per word

More recent approaches mostly relied on neural network architectures that encode multi-sense information. [50] motivated by the clustering approaches of previous models, followed an online method of learning skip-gram sense embeddings during which they also estimated the number of clusters. Contrary to previous work, both context and word vectors were learned simultaneously, instead of learning context representations as part of a pre-processing step. Later, a dynamic Gaussian skip-gram mixture model was introduced by [115] enabling the detection of different number of senses for each word during training. In that work, each word was represented as a Gaussian mixture instead of a point vector in the embedding space, where each Gaussian component represented a sense of the word. In a more recent work, [66] made use of autoencoders to map each word to a context-specific representation, while [67, 116] implemented discrete sense selection through reinforcement learning.

All of the above methods utilized the context information of each word occurrence without taking into account the relative order of words in the context window. [117] suggested that this omission impairs the quality of multi-prototype representations derived by clustering-based methods, and noted that the order of context words matters to the meaning of the target word. To tackle this issue, they

used a neural network model, called CSV (Context-Specific Vector), which can generate both word and context representations. Their proposed neural network architecture contained a convolutional layer that was designed to produce context representations reflecting the order of their constituents. After the refined generated context representations were extracted, they were used to learn context-specific multi-prototype word embeddings.

Another definition of context was also described in [53], who treated context as a topic domain. Motivated by the observation that polysemous words usually change their meaning when they reside in different topic domains, they were the first to utilize topic modeling to learn multiple-prototype representations. Specifically, the Latent Dirichlet Allocation (LDA) algorithm was employed into the skip-gram model to get the distribution of a word over topics, which was further utilized to extract topic-word embeddings. In a more recent work, LDA was utilized to infer the weights of each topic. The weights were further used to define a mixture vector representation for each target word that predicted its corresponding context words [55]. Moreover, [118] exploited Wikipedia articles and assumed that words co-occurring in articles under the same subject share the same sense. The sense-aware prototypes were produced via clustering the Wikipedia pages based on the global and local contextual information of the target word.

A probabilistic approach was followed by [51], who proposed that a word should be associated with a new sense when there is evidence in its context suggesting that it sufficiently differs from its early senses. They also noted that such a theoretical scheme naturally points to Chinese Restaurant Process. According to this probabilistic framework, each word occurrence corresponds to a costumer while each table corresponds to a sense of a word. In these terms, a new word occurrence could either sit in an occupied table (assigned to an existing word sense), or choose an unoccupied table to sit (assigned to a new word sense).

[116] proposed a different theoretical framework to induce multiple sense-specific embeddings for each ambiguous word, using a recurrent neural network. Instead of using the contextual information of words as evidence of their possible meanings, they utilized bilingual resources motivated by the fact that a word with multiple senses could have a different translation in another language.

3.3.2 Knowledge-Based Models

The unsupervised methods reviewed so far attempt to conceptualize the polysemic nature of words via creating multiple-prototype representations from raw contextual information extracted from massive text corpora. More recent techniques that achieve state-of-the-art performance in contextual semantic similarity tasks, involve knowledge-based approaches. In general, these knowledge-based approaches utilize an incomplete knowledge base along with a large corpus of text and try to use the first as a prior knowledge to the problem. The most widely known sense inventory used as an auxiliary knowledge for multiple-prototype representations extraction is WordNet. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms called *synsets*, each expressing a distinct concept as described in [119].

[56] used the definitions provided for each word by WordNet, in order to assign vector representation to senses. Using these sense vectors as initial estimations along with single-prototype word vectors, they refined them through word sense disambiguation algorithms. Given the disambiguated words, they finally modified the skip-gram model in order to jointly train words and sense vectors. Later, [57] used BabelNet as their underlying sense inventory, which constitutes an enriched database of WordNet. By leveraging the knowledge of the inventory they automatically generated sense-annotated corpora, using a word sense disambiguation algorithm. Sense-agnostic representations were extracted via employing the skip-gram model over the annotated corpus.

Another knowledge based approach introduced by [88] thought of words as sums of their lexemes (units), and synsets as sums of their lexemes. The interpretation of this theoretical foundation naturally establishes algebraic operations between word vectors in a mathematical algorithm. More specifically, pre-trained word embeddings were extended to embeddings of lexemes and synsets, with the help of WordNet. Recently, [89] de-conflated pre-trained word representations based on the deep knowledge

derived from WordNet. After linking these pre-trained representations to WordNet, they extracted a list of semantically biased words towards the ambiguous word. Given the biased words and the target word's lemma representations, they extracted a sense-aware representation for the target word via searching for the vector with the minimum distance from it.

3.4 Latent Dirichlet Allocation

The Latent Dirichlet Algorithm (LDA), introduced by [58], is a generative probabilistic model of a corpus, that attempts to identify the hidden topics lying behind it. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. In linguistics, the word "topic" refers to an abstract scheme that gives us information about what is talked about in a set of words (i.e. sentence/document). From a mathematical view, in LDA, one could imagine that a "topic" in NLP applications is described as a set of words that frequently occur together. In statistical terms, it could be presented as a distribution over the vocabulary of a particular corpus. For interpretation reasons, if the distribution is extracted, we can obtain the set of most related words with respect to a specific topic via applying a threshold to its distribution (retain words with a probability above a determined threshold). Topic modeling is a classic problem in information retrieval. Related models and techniques are, among others, latent semantic indexing, independent component analysis, probabilistic latent semantic indexing, non-negative matrix factorization, and Gamma-Poisson distribution. The LDA model is highly modular and can therefore be easily extended. The main field of interest is modeling relations between topics.

3.4.1 Intuition

The basic idea of LDA is that documents (set of sentences) are represented as mixtures over topics, where each topic is characterized by a distribution over words. This assumption implies that a document could not exhibit only a single topic, which seems to be logical as documents are large entities of text. To gain insight into this assumption one can examine the distribution of possible topics, in a particular document, as presented in Figure 3.4.

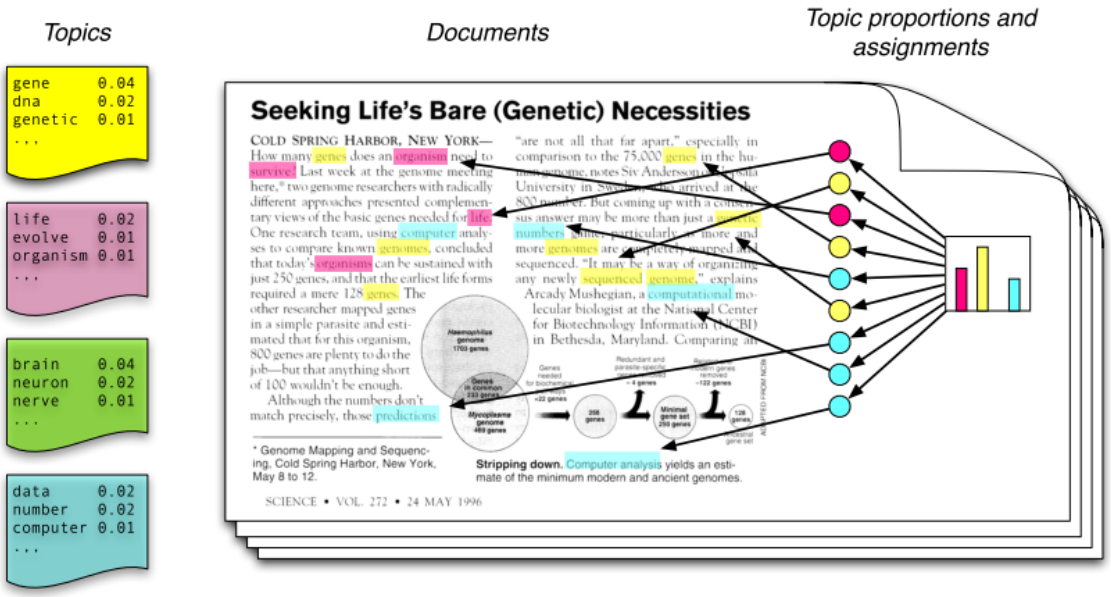


Figure 3.4: Intuition of LDA, an example presented in [71].

As explained in [71], the article is about using data analysis to determine the number of genes that an organism needs to survive. The article has been highlighted manually in order to create clusters of words that could be attributed to each of the topics residing in it: genetics, data analysis and evolutionary biology. The words about data analysis, such as "computational" and "prediction" have been highlighted in blue; words about evolutionary biology, such as "survive" and "organism", have been highlighted in pink; words about genetics, such as "sequenced" and "genes," have been highlighted in yellow. LDA tries to capture the above intuition, and automate the procedure of assigning topics to documents, and word distributions to topics. To do so, it does the following for each document:

1. Assume there are k topics across all of the documents

2. Distribute these k topics across document m (this distribution is known as α and can be symmetric or asymmetric, see for the example histogram on the right) by assigning each word a topic.

3. For each word in the document:

   (a) Assume its topic is wrong but every other word is assigned the correct topic.

   (b) Probabilistically assign word w a topic based on two things:

   - what topics are in the document
   - how many times this word has been assigned a particular topic across all of the documents

4. Repeat this process a number of times for each document.

3.4.2 Notation and Terminology

As we are going to put the above intuition into a mathematical framework, we should firstly introduce the basic notation and terminology needed to describe linguistic terms and concepts such as "words", "documents","corpora", "topic", as well as the document-topic and the topic-word distributions. Following [58] we define:

- A *word* is considered the basic unit of our data, defined to be an item from a vocabulary indexed by $\{1, ..., V\}$. Mathematically, it is represented as a vector that has a single component equal to one and all other components equal to zero. For example, the representation of the first word of the vocabulary corresponds to a V-dimensional vector $w_1 = [\ 1\ 0\ 0\ 0\ 0\ ...\ ]$.

- A *document* is a group of N words denoted by $\boldsymbol{d} = (w_1, w_2, ..., w_N)$, where w_n is the n-th word in the group.

- A *corpus* is a collection of M documents denoted by $\boldsymbol{C} = \{\boldsymbol{d_1}, \boldsymbol{d_2}, ..., \boldsymbol{d_M}\}$.

- A *topic* is a distribution over the vocabulary noted as β (β_k denotes the topic distribution of the k-th topic, where $k \in K$ and K corresponds to the total number of topics).

- The *document-topic* distribution for document $\boldsymbol{d}$ is defined as θ_d, while $\theta_{k,d}$ is the topic proportion of topic β_k in document $\boldsymbol{d}$.

- The *topic-word* distribution for document $\boldsymbol{d}$ is defined as z_d, while $z_{d,n}$ is the topic assignment for word w_n in document $\boldsymbol{d}$.

3.4.3 Algorithm

Generally, LDA could be described as a generative probabilistic model of a corpus, where the observed variables are documents and the latent variables are the topics residing in the corpus. As mentioned above, the basic idea of the algorithm is that each document could be assigned to a distribution over topics, where each topic is a distribution over words. In order to infer these distributions LDA assumes the following generative process for each document $\boldsymbol{d}$ in a corpus $\boldsymbol{C}$, whose graphical representation is given if Figure 3.5:

1. Choose $N \sim$ Poisson(ξ), where N corresponds to the number of words for $\boldsymbol{d}$.
2. Choose $\vartheta \sim$ Dirichlet(α)
3. For each of N words, w_n:
   (a) Choose a topic $z_n \sim$ Multinomial(ϑ)
   (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

The ultimate goal of the above process is to estimate the hidden distributions $\theta_{1:D}$, $z_{1:D}$, $\beta_{1:K}$, given the observed variables $w_{1:D}$. As a result, the key inferential problem that we need to solve in order to use LDA is that of computing the posterior distribution of the hidden variables given the corpus as analyzed in Equation 3.3, using the Bayes' Theorem.

$$p(\beta_{1:K}, \vartheta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{3.3}$$

The numerator of the above fraction can be computed as the joint distribution of all random variables. However, in order to compute the denominator of the fraction we have to marginalize over all possible topic structures defined by $\theta_{1:D}$, $z_{1:D}$ and $\beta_{1:K}$. When doing so a coupling between $\theta_{1:D}$ and $\beta_{1:K}$ arises making the separation of them in the computation of the log likelihood function impossible.

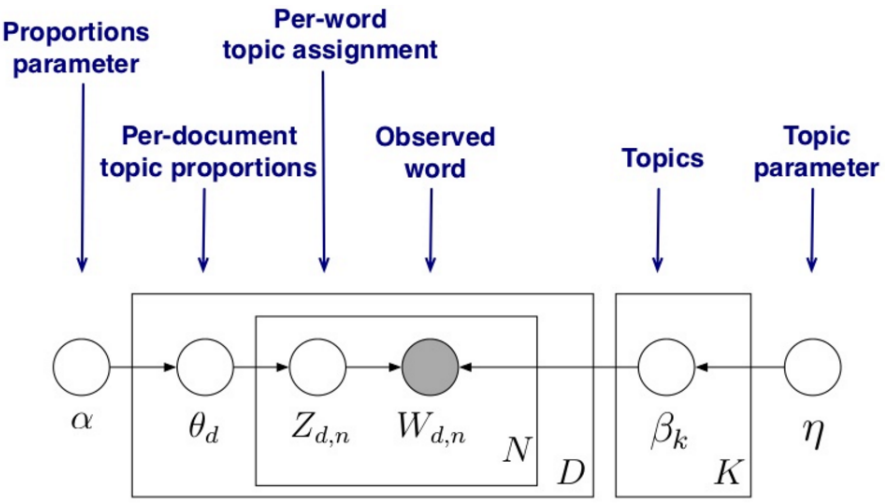


Figure 3.5: Graphical model representation of LDA as presented in [58].

So while exact inference is not tractable, various inference techniques have been proposed in order to approximate the above solution:

- **Variational Inference**. The idea proposed by [120] was to modify the original graphical model of Figure 3.4 by removing the edges and nodes which are responsible for creating the undesirable coupling mentioned above. As a result a simpler distribution is used in order to approximate the real.

- **Collapsed Gibbs Sampling**. The approximation introduced by [121] was that a high-dimensional distribution is simulated by sampling on lower-dimensional subsets of variables where each subset is conditioned on the value of all others. The sampling is done sequentially and proceeds until the sampled values approximate the target distribution.

- **Collapsed Variational Inference**. [122] made weaker factorization assumptions than those made by the Variational Inference algorithm in order to approximate the true posterior. Specifically, instead of assuming the parameters to be independent from latent variables they treat their dependence on the topic variables, in an exact fashion marginalizing out the θ and β variables.

- **Online Variational Inference**. Later, [123] noted that the Varational Inference algorithm requires a full pass through the entire corpus each iteration, making the whole procedure slow for large datasets. In this direction they proposed an online variational inference algorithm based on stochastic optimization with a natural gradient step. They also showed that the algorithm produces good parameter estimates on large datasets dramatically faster than batch algorithms.

3.5 Transformations in semantic spaces

As mentioned previously, neural network models —such as Word2Vec or FastText— have become very popular recently. The vector representations they produce, has been proved that significantly and continuously outperform the traditional count-based models [22]. Many scientists attributed this superiority to the natural edge of neural networks over methods that solely relied on word co-occurrence counts. One of the main characteristics of predictive Distributional Semantic Models is that they create semantic spaces which are not aligned to a fixed coordinate system, due to their non-deterministic nature. Basically, this means that if the algorithm runs under the same dataset twice, the resulted semantic spaces have drastically different global structures. For this reason, the problem of defining transformations between semantic spaces has attracted a lot of attention recently, as it enables the comparison of the distributed representations that belong to different datasets.

The most popular application of semantic spaces transformation is machine translation, where the ultimate goal is to automate the process of generating large dictionaries starting from few bilingual data. [72] were the first to introduce such mappings in order to predict translations between English and Spanish words. After learning word representations for the two languages using the Word2Vec model, they proposed a linear mapping between the two language-specific semantic spaces. After the alignment, the correct translation of a target word is expected to lie near the target word.

As they noted, their core motivation was that all common languages have similar geometric arrangements, as they share concepts that are concept in the real world. Later work on machine translation focused on the properties of the transformation matrices between languages [60], as well as on the properties of the embeddings being mapped to the shared space. Specifically, [124] showed that the neighborhoods of the mapped embeddings are highly polluted by *hubs*, which are defined as vectors that tend to be popular nearest neighbors of many items.

Another application of semantic spaces transformation was later studied by [125], who attempted to explore the semantic differences of words between the informal English of social media (Twitter corpus), and the formal English of well organized texts (Wikipedia corpus). Towards aligning the two semantic spaces, they assumed that a mapping existed between the most frequent words of the two corpora. After mapping the two languages to a common space, they employed a normalization of word distances based on term-frequency. Finally, they used these distances to find discriminative words —in terms of usage— between the two corpora.

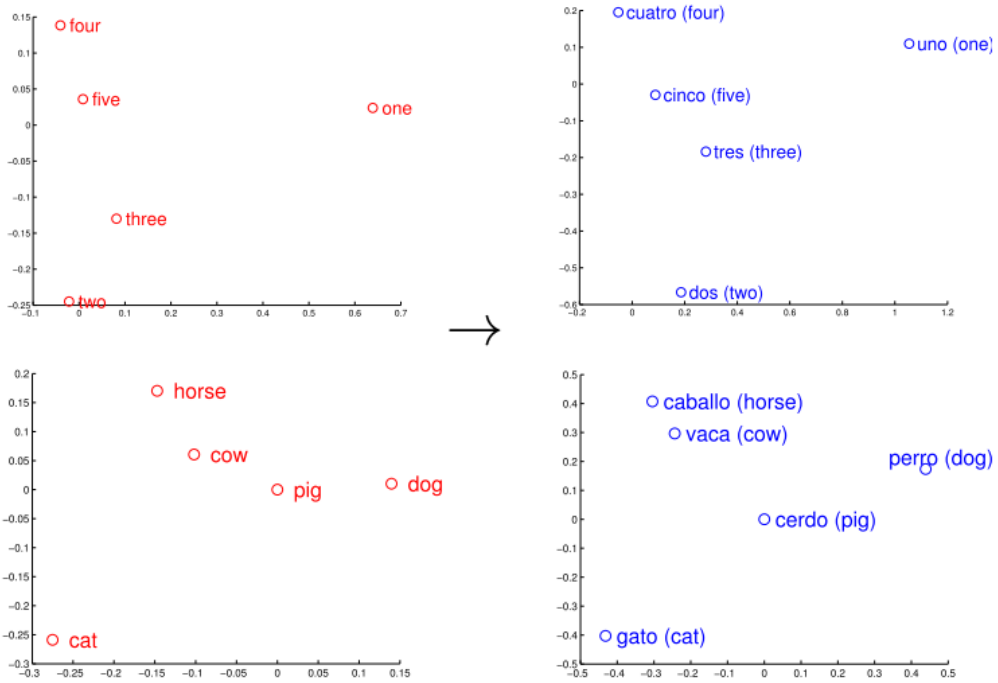


Figure 3.6: Projections of distributed word vector representations of numbers and animals in English (left) and Spanish (right) using PCA as presented in [72].

The semantic evolution of words' meaning can be captured in large-scale corpora that refer to different periods of time. [91] created diachronic embeddings, by constructing embeddings in each time-period and then learning consecutive linear rotational matrices that mapped the vector spaces of historic corpora that corresponded to different time intervals, to track the semantic drifts of words within-years. The relative high dimensionality of diachronical embeddings poses a challenge, as they are typically not embedded in 2 or 3 dimensions that can be easily interpreted by humans. For this reason, dimensionality reduction techniques usually take place in order to visualize the trajectory a word follows over time in a 2 dimensional space.

Figure 3.7 illustrates an example of words' trajectories that reveal semantic evolution of words through time. By comparing the relative position of the words with their "temporal" nearest neighbor we could track interesting semantic shifts in their meaning that could also reflect cultural evolution. For instance, the word gay shifted from meaning "cheerful" or "frolicsome" to referring to homosexuality. In the early 20th century broadcast referred to "casting out seeds"; with the rise of television and radio its semantics shifted to "transmitting signals". The word "awful" underwent a process of pejoration, as it shifted from meaning "full of awe" to meaning "terrible" as reported in [91].

Recently [59] applied semantic space transformations in an attempt to enrich the coverage of an existing vocabulary with rare or unseen words. The interesting property of their approach is that distributional information derived from text corpora, could be used in order to complete the missing parts of knowledge bases and vice-versa. To achieve it, they created a mapping between a distributional semantic space and a lexical ontology using *semantic bridges* of monosemous words.

Mapping Methods

We start by defining basic terminology in order to explain the most popular methods of alignment between semantic spaces that can be found in the literature. Let X and Y be the word embedding matrices of the source and target language, respectively. The i-th column of matrix X is the distributed vector representation $x_i \in \mathbb{R}^d$ of word i, while $y_i \in \mathbb{R}^d$ is its equivalent distributed vector represen-

Figure 3.7: Two-dimensional visualization of semantic change of three English words.

tation in the target language. We aim to find a transformation matrix $W \in \mathbb{R}^{d \times d}$ that maps the source language to the target language, such that WX is as close as possible to Y. As summarized in [126], this transformation matrix could be computed through linear, orthogonal of canonical methods.

- **Linear** methods in this area were introduced by [72] who used a linear mapping as the first attempt to align semantic spaces for machine translation. They used a least squares objective function that minimizes the sum of squared Euclidean distances between the translated pairs vectors of two languages, without imposing a restriction to the matrix. This problem (also known as Ordinary Least Square) has a closed-form solution as indicated in Equation 3.4.

$$W = \arg\min_{W} \|WX - Y\|_F = (X^t X)^{-1} X^t Y. \tag{3.4}$$

  Few years later, [124], incorporated an L2-regularization term to the objective function.

- **Orthogonal** methods were firstly proposed by [60] who noticed that both the source and the target vectors should remain normalized to unit length during the learning phase of the mapping algorithm. They also noted that normalization is a crucial characteristic that the aligned representations should hold, as it ensures that the dot product between two vectors falls back to their cosine similarity, the most widely used distance measurement between word embeddings. For this reason, they mapped the source space to the target via solving the constraint optimization problem of Equation 3.5.

$$W = \arg\min_{W} \|WX - Y\|_F, \;\; \text{subject to} \;\; WW^T = \mathbb{I}. \tag{3.5}$$

  From a mathematical perspective, the above problem is known as the orthogonal Procrustes problem and it has a closed form solution. The optimal W is recovered by UV^T, where U and V, are obtained through the Singular Value Decomposition (equal to $(U\Sigma V^T)$) of YX^T. For a more detailed review of the problem we refer the reader to [63].

- **Canonical** methods on the other side, compute two distinct linear mappings M_1 and M_2 first, where the objective is to maximize the correlation between the dimensions of the projected matrices M_1X ans M_2Y. After computing the two mappings, the transformation matrix W is recovered through a simple algebraic operation as noted in Equation 3.5. [127] were the first to use Canonical Correlation Analysis in order to map two semantic spaces, which was later proved to give similar results to the orthogonal mapping.

$$W = M_1^{-1} M_2, \;\; \text{where} \;\; M_1, M_2 = \underset{M_1, M_2}{\text{argmax}} \, \text{cov}(M_1 X, M_2 Y). \tag{3.6}$$

3.6 Natural Language & Cognition

The advances in artificial intelligence and the post-Google interests in information retrieval, in the recent decades, have made large-scale processing of human language data possible and produced impressive results in many language processing tasks. However, the wealth and the multilingualism of digital corpora have generated additional challenges for language processing and language technology. To overcome some of the challenges an adequate theory of this complex human language processing system is needed to integrate scientific knowledge from the fields of cognitive science and cognitive neuroscience, in particular. Over the last few years, emerging applications of NLP have taken a cognitive science perspective recognising that the modelling of the language processing is simply too complex to be addressed within a single discipline.

3.6.1 Brain Imaging Modalities

Before brain imaging technologies were developed, the study of language in the brain used reaction times and eye tracking, and a considerable amount of progress was made with these simple measurements. More sophisticated brain imaging technologies have become very popular in recent decades, and have allowed researchers to explore the brain's activity during a variety of tasks.The most common brain imaging technologies are Electroencephalography (EEG), Magnetoencephalography (MEG) and functional Magnetic Resonance Imaging (fMRI).

Each technique has its own unique advantages and disadvantages, and each measures brain activation in a different way. Electroencephalography (EEG) measures the voltage fluctuations along the scalp that occur when many neurons fire in a coordinated fashion. EEG has the benefit of being able to record changes in voltage by the millisecond, making it one of the best brain recording modalities in terms of time resolution (similar resolution to MEG). However, the largest drawback of EEG is poor spatial resolution, which is caused by interference from the skull and scalp. MEG measures the magnetic field caused by many neurons firing in synchrony. That is, MEG measures the currents caused by external neurons sending signals to groups of neurons that lie parallel to the skull. Like EEG, MEG has time resolution on the order of ms. The spatial resolution of MEG is better than EEG. The imaging technique with the greatest spatial resolution is functional Magnetic Resonance Imaging (fMRI), which can achieve resolution as fine as 1mm. An example fMRI image appears in Figure 3.8. fMRI measures changes in blood oxygenation in response to increased neuronal activity, called the blood-oxygen-level dependent (BOLD) response. Because fMRI depends on the transport of oxygen via blood to the brain, its time constant is governed by the rate at which blood can replenish oxygen in the brain.Though fMRI can acquire images at the rate of about 1 image per second, the BOLD response can take several seconds to reach its peak after a stimulus is shown. Thus, amongst the three modalities discussed here, fMRI has the worst time resolution and the best spatial resolution.

3.6.2 Semantics and Brain

Semantics in the brain has historically been studied not by comparing the magnitude of activity between conditions, but rather by the information encoded in the neural activity. One can measure the information encoded in neural activity by training machine learning algorithms to predict some feature of the input stimuli. Machine learning algorithms do not require large differences in magnitude between conditions, but rather leverage patterns in the recordings of neural activity, which may involve differences in signal in both the positive and negative direction indifferent areas of the brain at different times. We will discuss Machine learning to recover the neural information encoding in greater detail in Chapter 5.The study of semantics in the brain has often linked brain activation to linguistic measurements of semantics.

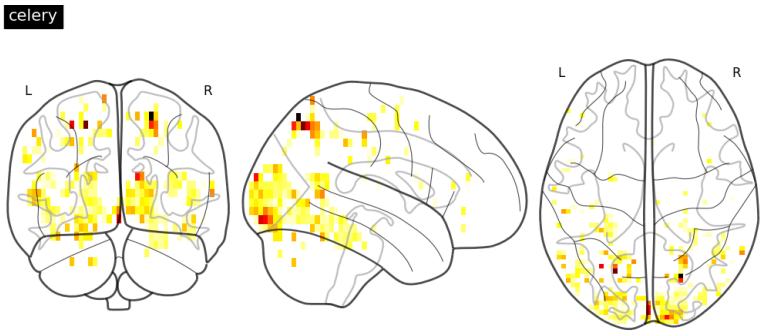


Figure 3.8: Neural activation image for the noun celery similar concrete nouns including the 500 most stable voxels (participant P1).An fMRI image is 3D. This figure shows just one horizontal slice in Montreal Neurological Institute (MNI) space of the three-dimensional image. The color of each voxel (pixel in brain space) represents the percent change over baseline of the BOLD response in that brain area.

[7] showed that the fMRI activity of people reading 60 common concrete nouns could be modeled as the linear combination of features derived from 11 verb co-occurrence with the target word. [128] extended this work to show that similar results could be obtained using feature norms (free-form naming of word characteristics), and [129] showed that the activity from noun reading could be tied to biologically relevant brain areas (e.g. manipulation-related words to motor cortex). [29] used Magnetoencephalography (MEG), the same 60 words of [7], and behavioral data collected via Mechanical Turk to explore the neural basis of semantic representation. [29] found that the semantic representation of a word unfolds overtime, and that different semantic elements appear at different times in different parts of the brain.In [30], they showed that a set of automatically derived corpus statistics (a VSM) could perform as well as the behavioral data from [29]. Another linguistic resource, WordNet, has also been used to study language in the brain. WordNet ([130]) is a lexical database where English words and relationships between words are recorded (e.g. cat "is a kind of" feline). Words may be associated with groups of synonymous words called "synsets". [10] annotated 2 hours of video with over1700 WordNet categories. These annotations were then used to map semantic categories onto the brain via linear regression. The study confirmed much that was already known about semantics in the brain (e.g. face stimuli give strong reactions in the fusiform face area - FFA) but also showed the extremely distributed nature of semantics in the brain. For example, videos containing people show activation in FFA, but also in posterior Superior Temporal Sulcus (pSTS - associated with gaze following), in the Extrastriate Body Area (EBA - activated by stimuli containing body parts) as well as widespread activation in frontal and temporal regions. A brain-browsing interface has been supplied by the authors (`http://gallantlab.org/brainviewer/huthetal2012/`) which can be used to explore WordNet in cortical space.

Recently, MEG has ben used to study the effect of context on brain activation while subjects read a chapter from a story. [131] used different linguistic techniques were used to represent semantics - Recurrent Neural Network Language Models (RNNLM) ([132]) and Neural Probabilistic Language Models (NPLM) ([133]). Each of these two models is a multilayer neural network which represents the history of words encountered. In the case of RNNLM, an unlimited lexical history is available, constrained only by the size of the hidden layer in the network, whereas a 3- or 5-word history is used to train a NPLM. Both models are trained to predict the next word, given the word's previous context. Then a model was trained to predict story-reading MEG activity from the hidden, output or embedding layers of the neural networks. [131] found that the hidden layer of a RNNLM performed best,followed by the hidden layer of a NPLM given 5 words of context. Context vectors were most useful for predicting brain activity 250ms after the onset of a word, perhaps reflecting the process of combining a new word with the current semantic state. Thus, [131] show that story context can

be used to differentiate brain states, and that some amount of the brain activation is correlated to the prediction of the next word in a story. However, when the semantics of the sentence changes due to different words or differing context, the semantic retrieval/memory and unification processes will also change, resulting in differential brain activity.

Chapter 4

Neural Activation Semantic Models

4.1 Computational Cognitive Semantic Models for Natural Language

In this section, our published work in the 27th International Conference on Computational Linguistics (COLING 2018) is presented [1].

4.2 Motivation

Mental process of encoding and decoding meaning of concepts is not fully understood. The process of mapping neural activations to word embeddings has been explored neuroscientists and computational linguists [134, 7, 30, 129]. However, the computational integration of brain information in word representations is little explored and could help us encode word semantics better. We propose an approach for calculating word similarity of concrete nouns from predicted neural activations. Its analysis show improvement of performance over conventional word embeddings for highly similar/dissimilar words. Moreover, we use predicted neural activations of concrete nouns along with conventional word embeddings and evaluate their performance in Taxonomy, Textual Entailment, Human Sense Classification tasks. Fusion of neural activations and conventional word embeddings can improve performance.

4.3 Related Work

A significant body of literature investigates neural activations by mapping word semantics to fMRI data. Most of them have in common a basic idea published in [7]. In this work, a model is introduced that maps low dimensional word coocurence vectors to neural activations. The approach is validated in a neural activation-based word classification task. This work shows that the mapping between lexical semantic spaces constructed via computational lexical semantic algorithms and 3D neural activations representations measured via fMRIs is possible.

A first variant of the aforementioned model was introduced in [8], where the use of WordNet features was investigated for constructing the lexical semantic space. Word classification results reported showed similar performance to [7], however, by fusion of the two lexical semantic models improved classification results where achieved.

A second approach, introduced in [135], extends the work in [7] by increasing the number of fMRI voxels used in the neural activation vectors and the number of featurs (dimension) of the lexical semantic model showing additional performance improvement.

Algorithms that count word co-ocurences and utilize hand-crafted features for constructing lexical semantic models can be found in [21, 38].Moreover, various lexical semantic models that predict a word based on its context have also elaborated [23, 25, 27]. Word prediction models tend to perform better in natural language processing tasks such as analogy, similarity, synonym detection, concept taxonomy [22] and sentiment analysis [113, 136]. However, their relationship with cognitive lexical representation is not yet well understood, at least to a degree that would allow us to improve current computation lexical semantic models. Along these lines, there have been two main lines of work. In [34], neural activations were integrated in the training procedure of lexical semantic models in

order to learn word embeddings that include latent neural information. Although a small number of words was used to bootstrap the neural activation representations, it have been shown that their model can predict unseen words and generalizes well across different topics. Ruan et al. [35], have shown that neural activations for different parts of the brain are correlated with word embeddings especially skip-grams. A semantic model was also proposed for training word embeddings as a first step towards including cognitive information in a word vector representation.

4.4 Decoding the meaning of nouns to predict human brain activity

The human ability of translating concepts into words and back depends on the ability of mind to decode and encode meaning. This mental process, which is not currently completely understood, has captivated the interest of both neuroscientists and computational linguists [134, 137, 138, 139, 129]. Specifically, when a person experiences a visual stimulus of a concept, reads, speaks or writes a word, particular neuronal regions in the brain are activated [31].

Various studies have been carried out to explore brain encoding and decoding mechanisms when a stimulus is present, as detailed next. For visual stimuli, studies have shown that is feasible to discriminate and reconstruct images using patterns of neural activity, mainly found in the visual cortex [12, 13, 14, 15, 16], the part of brain responsible for visual information processing. Other works have demonstrated the relationship between cognitive perception and speech [17, 18]. Regarding textual stimuli, researchers have shown distributed semantic maps of words are present in our brains [10, 29]. Lexical semantics are based on the assumption that similar words appear in similar contexts [19].

Based on that assumption, two different approaches for building semantic models have been proposed. The first approach is to encode word semantics, by applying dimensionality reduction of context-word occurrence matrix which was computed using large corpora [20, 21]. The second approach replaces thse "counting" by predictive models [22] based on neural networks [23, 24, 25, 26, 27]. Counting models calculate and weight context vectors, while predictive models learn word vectors by guessing the context in which these words tend to appear.

In pursuance of enriching such lexical semantic models with cognitive information, as well as discovering the cognitive representation of word semantics, a number of studies have attempted to examine the mapping between semantic representation of computational and cognitive models. In prior work, it has been shown that semantic of words are related to activation potentials in regions of the brain and that decoding between neural activations and semantic content [7, 28, 29, 30, 31] is possible. Furthermore, neural activations are shown to have predictive power with respect to semantics at the word [7, 8] and sentence [32, 33] level. Computational studies that aim to explore the influence of neural activations in word representations have shown that by incorporating neural activations when training lexical semantic models can improve their generalization ability despite the small amount of neural activation data used [34, 35].

These works show that a strong relationship exists between computational semantic models and neural representations. However, it remains to be seen how cognitive semantic representations, including localized neural activation patterns can help improve the performance of computational semantic models, especially for complicated classification and recognition tasks. Motivated by the aforementioned studies that show correlation between localized neural activations and word semantics, we propose a computational model for semantic similarity that utilizes predicted neural activations learned from a small set of concrete nouns.

The proposed model is applied to a variety of natural language processing tasks. The neural activation prediction model used here for lexical expansion is that proposed in [7]. In our list of experiments, we first compare the performance of the proposed neural activation model for a concrete noun semantic similarity task and show that for certain word pairs it outperforms the state-of-the-art. Then we evaluate the performance of neural activation vectors for a word classification, sensory modality (sense) classification and textual entailment task. The fusion of neural and traditional word embed-

ding vectors are shown to outperform the state-of-the-art. To our knowledge, this is the first time brain imaging data are successfully used for the aforementioned tasks.

4.5 Brain and Human Senses

4.5.1 Vision

Processing of visual information is quite complex compared to that of other special senses we possess. Vision processing in the brain is not particularly confined to a specific region, rather it follows a global circuitry (involving multiple regions of the brain). The visual cortex of the brain is a part of the cerebral cortex that processes visual information. It is located in the occipital lobe in the back of the head [45]. The ventral stream, sometimes called the "What Pathway", is associated with form recognition and object representation. It is also associated with storage of long-term memory. The dorsal stream, sometimes called the "Where Pathway" or "How Pathway", is associated with motion, representation of object locations, and control of the eyes and arms, especially when visual information is used to guide saccades or reaching.

4.5.2 Audition

The Temporal Lobe mainly revolves around hearing and selective listening. It receives sensory information such as sounds and speech from the ears. It is also key to being able to comprehend, or understand meaningful speech. In fact, we would not be able to understand someone talking to us, if it wasn't for the temporal lobe.[46] This lobe is special because it makes sense of the all the different sounds and pitches (different types of sound) being transmitted from the sensory receptors of the ears. The auditory cortex is the most highly organized processing unit of sound in the brain. This cortex area is the neural crux of hearing, and—in humans—language and music. The auditory cortex is divided into three separate parts: the primary, secondary, and tertiary auditory cortex. These structures are formed concentrically around one another, with the primary cortex in the middle and the tertiary cortex on the outside. The primary auditory cortex is tonotopically organized, which means that neighboring cells in the cortex respond to neighboring frequencies[140].

4.5.3 Touch

Cortical homunculus is a distorted representation of the human body, based on a neurological "map" of the areas and proportions of the brain dedicated to processing motor functions, or sensory functions, for different parts of the body. Touch is mediated by primary (SI) and secondary (SII) somatosensory cortex. Touch first arrives in cortex at the primary somatosensory cortex. This region is known for its homuncular organization, that is, the arrangement of neurons is determined by the arrangement of receptors on the skin[42].

4.5.4 Taste

The primary gustatory cortex is a brain structure responsible for the perception of taste. It consists of two substructures: the anterior insula on the insular lobe and the frontal operculum on the inferior frontal gyrus of the frontal lobe[43]. By using extracellular unit recording techniques,Kobayashi, Masayuki [44] have elucidated that neurons in the AI/FO respond to sweetness, saltiness, bitterness, and sourness, and they code the intensity of the taste stimulus.

4.5.5 Olfaction

Olfactory system is also located in gustatory cortex[47]. Once an odor molecule binds to a receptor, it initiates an electrical signal that travels from the sensory neurons to the olfactory bulb, a structure at the base of the forebrain that relays the signal to other brain areas for additional processing. One of these

areas is the piriform cortex, a collection of neurons located just behind the olfactory bulb that works to identify the smell. Smell information also goes to the thalamus, a structure that serves as a relay station for all of the sensory information coming into the brain. The thalamus transmits some of this smell information to the orbitofrontal cortex, where it can then be integrated with taste information. What we often attribute to the sense of taste is actually the result of this sensory integration.

4.6 Neural Activations Semantic Model

The neural activation prediction model used here is that proposed in Mitchell [7][1]. A high level illustration of the proposed model can be seen in 4.1. First activation potentials are measured from fMRI images. We consider voxels to be 3D pixels created by MRI scanning software depicting brain state. Every voxel v is associated with a $TN \times V'$ array, M, of neural activation values (blood flow), where V' is the number of voxels, T is the number of trials, N is the number of stimuli, in our case different nouns. The first step is to select the most stable (salient) voxels, V to include in the neural activation model. The stability score s_v for voxel v is computed as the average pairwise Pearson correlation ϱ for all the different row combinations of M, as follows:

$$s_v = \frac{2}{TN(TN-1)} \sum_{i=1}^{TN} \sum_{j=i+1}^{TN} \varrho(M_{i,:}, M_{j,:}), \quad \forall\, v = 1 \cdots V' \tag{4.1}$$

where $M_{i,:}$ is the ith row of matrix M. High stability scores, s_v, as described in Equation 4.1, indicate that corresponding voxels have consistent representations across different trials and nouns. Next, the neural activation predictive model proposed in [7] is defined.

For this purpose we identify a set of m seed words s_1, s_2, ..., s_m, and a function $f_i(w)$ that estimates the association between seed word s_i and word w. The core assumption of the model is that words that are closely associated have similar neural activation patterns, thus the mapping from the associative (semantic) space to the neural activation (voxel) space is estimated as follows:

$$y_v(w) = \sum_{i=1}^{m} c_{v,i}\ f_i(w), \quad \forall\, v = 1 \cdots V, \tag{4.2}$$

where $y_v(w)$ is the activation of voxel v for word w, $f_i(w)$ is a scalar value that reflects the association between the i^{th} seed word s_i and the word w, m is the number of seed words (semantic features), V is the total number of voxels and $c_{v,i}$ is a learned weight ranging between 0 and 1. A set of 25 verbs (seed words) was identified in [7] as semantic features s_i; seed words were manually selected according to psycholinguistic criteria. The similarity function $f_i(w)$ was set to the (normalized) co-occurrence frequency of the i^{th} seed word and w, estimated on a corpus. The weights $c_{v,i}$ were estimated using the fMRI data on words w using linear or ridge regression estimation. Once $c_{v,i}$ have been estimated, Equation 4.2 can be used to predict the neural activation of unseen words[2]. In this section we present baseline results on the neural activation prediction model of [7] and investigate the performance of the proposed neural activation semantic model of Equation 4.4 for a semantic similarity task on the MEN dataset. Our goal here is first to reproduce the results in [7] and then to investigate the properties of neural activation embeddings semantic models compared to traditional embedding models, e.g., word2vec in [25].

[1] The features in [7] are attractive because of their simplicity and low-dimensionality, and generalize well for lexical expansion to a large lexicon compared to other works described in Section 4.4 that potentially perform slightly better.

[2] Although Equation 4.4 can be used to perform lexical expansion an any token w for which $f_i(w)$ can be computed, the proposed framework (choice of $f()$ and associated fMRI data) is meant for concrete words and typically only neural activations for concrete words are reported in the literature.

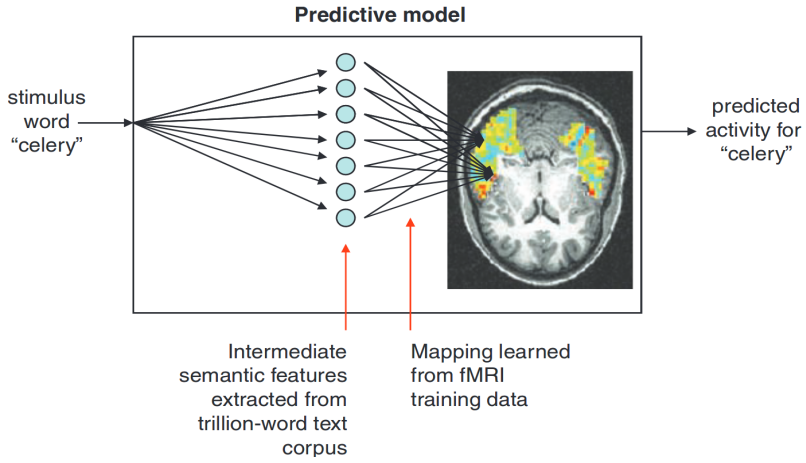


Figure 4.1: A high level overview of the neural predictor model [7].

4.6.1 Neural Activations Prediction Analysis

As proposed in [135], we choose the 500 most stable voxels from the fMRI images. Then, $c_{v,i}$ is estimated as in Equation 4.2 by applying linear regression per voxel across different words with regularization. To evaluate the neural activation prediction model we used cosine similarity in order to evaluate if our prediction for the possible pair of test words is correct or not. Correct prediction means that sum of the cosine similarities of the correct matched pairs is greater than the false matched pair as shown next:

$$cos(\vec{i_1}, \vec{p_1}) + cos(\vec{i_2}, \vec{p_2}) > cos(\vec{i_2}, \vec{p_1}) + cos(\vec{i_1}, \vec{p_2}), \tag{4.3}$$

where $\vec{i}$ is the actual image and $\vec{p}$ is the predicted image of 500 voxels. The dataset, which consists of 60 nouns, is split in train set (the rest 58 nouns) and test set (2 nouns) for all possible $\binom{60}{2} = 1770$ combinations using cross-validation. First, we examined the effects of two main parameters of the model. The number of stable voxels used for every participant and the effect of the regularization parameter applied in the linear model which maps the semantic to the neural space. In Figure 4.2, the variations of every participant's accuracy—calculated as in Equation 4.3— with respect to different values of the regularization parameter, $\hat{b}^{ridge}$, are depicted. We can clearly observe that smaller values of $\hat{b}^{ridge}$(in the interval$(0, 10]$) yield the best accuracy for most of the participants. However, Participants 3 & 9 appear to improve their performance until $\hat{b}^{ridge} \approx 400$, while Paricipants 2 & 6 demonstrate their best performance without regularization parameter. The variations in accuracy performance are at most $3 - 4\%$ except for Participant 2 which shows a higher sensitivity in the effect of regularization parameter. Overall, in our experiments we chose the regularization parameter which yielded the best performance for each participant. Next, the variations of every participant's accuracy(Equation 4.3) regarding to different numbers of stable voxels(V) is illustrated in Figure 4.3. Generally, we can see that the accuracy of different participants achieves its higher value for 200 to 500 stable voxels. Accuracy variations performance are at most 2% in that particular interval. As referred above, we selected the 500 more stable voxels in our setup to be consistent with [7], as their values after a particular threshold do not affect our final results and for participants which disagree with that threshold their respective best $\hat{b}^{ridge}$ mitigates accuracy loss.

Finally, in Figure 4.4 we show how the two parameters affect the average(across participants) accuracy. It is obvious, that small values of regularization parameter achieve the best performance as observed in most of the participants (Figure 4.2). Average accuracy demonstrates slight oscillations with regards to the number of stable voxels, appearing almost constant performance after 200 voxels.

After our experimentation with the two main parameters we chose to replicate the neural predictor model for $V = 500$ and $\hat{b}^{ridge}$ was selected separately for each participant according to his/her best accuracy. Then, the evaluation process(Equation 4.3) is followed for every one of 9 participants.

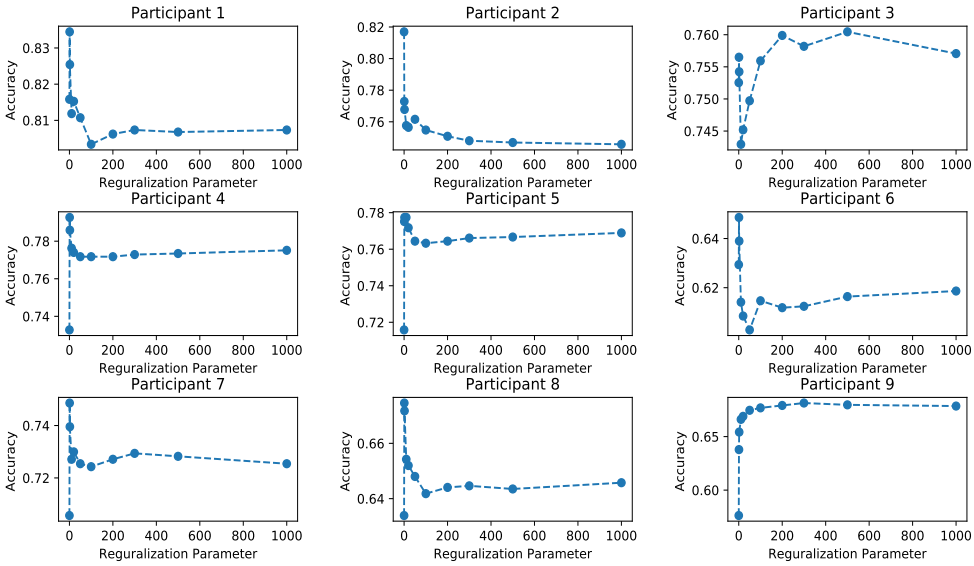


Figure 4.2: Accuracy of different participants for different values of regularization parameter. The number of stable voxels selected is 500.

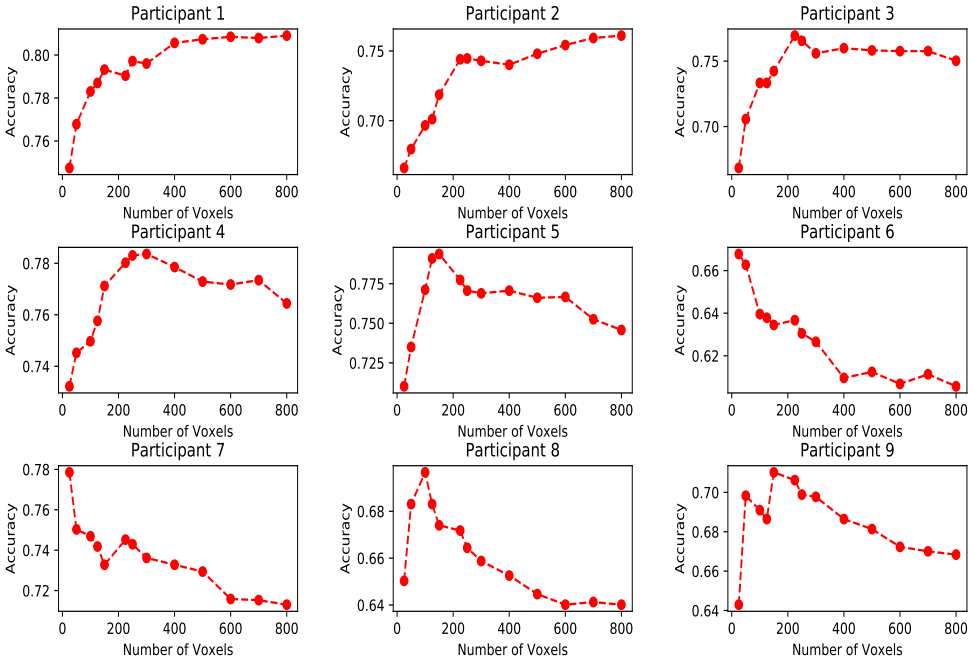


Figure 4.3: Accuracy of different participants for different values of stable voxels selected parameter. Regularization parameter is set to 1.

The results are shown reported in Table 4.1 when using linear and ridge regression to estimate $c_{v,i}$ in Equation 4.2. Results from [7] are also shown for comparison. Ridge regression performs very close to the results in [7], as expected. Results in Table 4.1 are on average consistent with the baseline results reported in [7]. We achieved higher performance for some participants and lower for others. This can be attributed to the different tools we used to implement the system (as we used scikit-learn [141]

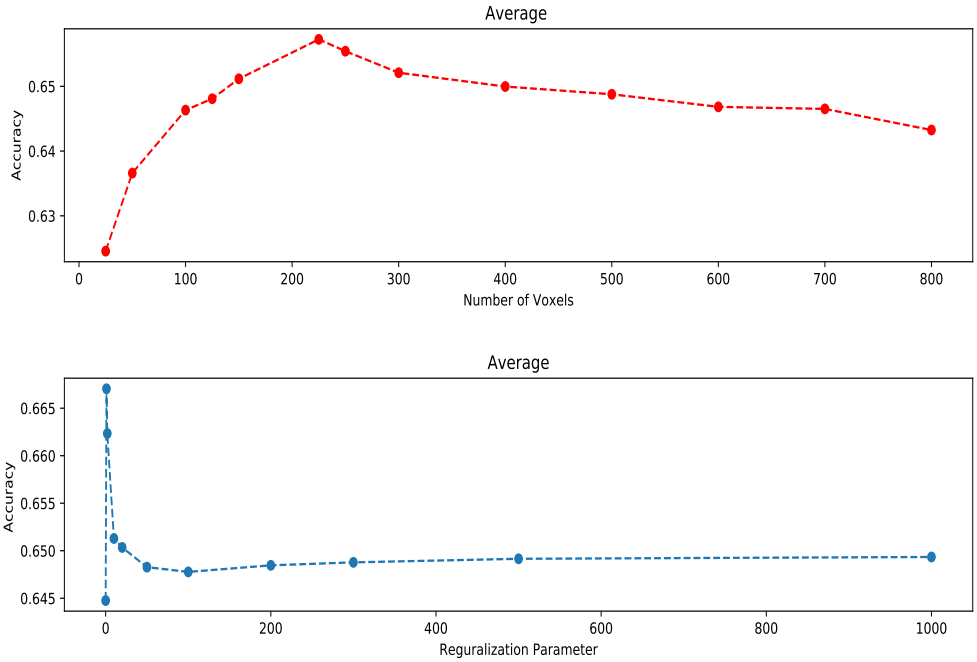


Figure 4.4: Average accuracy across participants for different values of regularization parameter (bottom) and regularization parameter(top). The number of stable voxels and regularization parameter were set as 500,1 respectively.

Participant ID	Linear Regression	Ridge Regression	Mitchell et. al
1	0.79	0.84	0.83
2	0.75	0.82	0.76
3	0.63	0.76	0.78
4	0.63	0.79	0.72
5	0.61	0.78	0.78
6	0.58	0.65	0.85
7	0.58	0.75	0.73
8	0.65	0.68	0.68
9	0.57	0.68	0.82
Mean	0.64	0.75	0.77

Table 4.1: Baseline Model Results

and [7] used Matlab). Moreover, we experimented with the number of voxels and our results agree with the findings reported in [135].

4.6.2 Semantic Similarity

Based on the hypothesis that similar words have similar neural activations, we propose a model to estimate word similarities based on neural activations predicted using Equation 4.2. We evaluated various metrics for computing semantic similarity from neural activations. We present only a top performing metric formulated as the weighted square distance, namely:

$$S(w_1, w_2) = \sum_{v=1}^{V} b_v \, (y_v(w_1) - y_v(w_2))^2, \tag{4.4}$$

where $S(w_1, w_2)$ is the semantic similarity between words w_1 and w_2, V represents the number of voxels used in the predicted neural image, $y_v(w)$ is the activation of a voxel for word w, and b_v is a learned weight of the contribution of a particular voxel to the similarity metric. In Figure 4.5 the

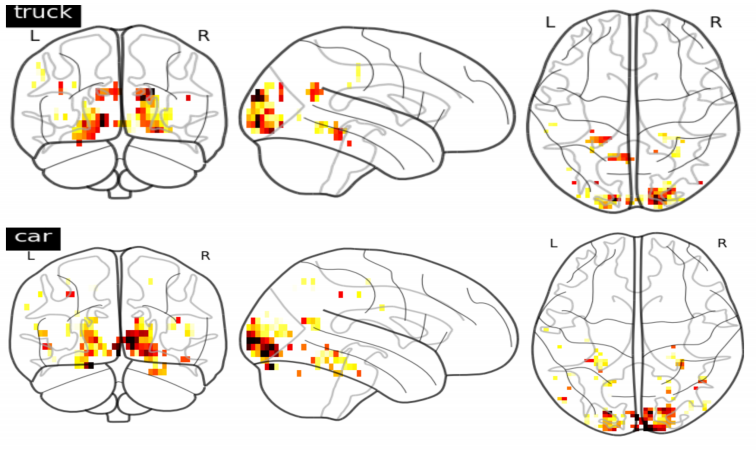


Figure 4.5: Neural activation images for two similar concrete nouns including the 500 most stable voxels (participant P1). This figure shows just one horizontal slice in Montreal Neurological Institute (MNI) space of the three-dimensional image.

predictions of neural activations for two highly similar nouns in fMRI dataset are presented. Visualizations were created from 500 voxels to gather insight for our computational model. Observe that both brain images have similar neural activations both in terms of which parts of the brain are activated and the activation values. Although, we don't utilize localization information, our weighting schema implicitly detects activation patterns by variations in b_i coefficients.

4.6.3 Taxonomy Creation

The performance of similarities computed by Equation 4.4 were also evaluated on a taxonomy creation task on the ESSLLI dataset [36]. Taxonomy creation is performed using the neural activation vectors $\vec{y}(w)$ estimated from Equation 4.2 and the coefficient vectors $\vec{b}$ defined in Equation 4.4 trained using linear regression on the whole of the MEN dataset. Then, the similarity matrix $S(w_i, w_j)$ is estimated for all pairs in the dataset using Equation 4.4 and then the spectral clustering algorithm proposed in [37] is applied to obtain the lexical classes. In this work, neural fusion refers to early fusion (vector concatenation) of word vectors and neural activation vectors. We used a purity-based metric for evaluating the quality of the automatically created clusters. The purity P of the taxonomy is defined as in [38]:

$$P = \frac{1}{d} \sum_{i=1}^{k} \max_{j}(e_{ij}), \tag{4.5}$$

where e_{ij} is the number of nouns assigned to the ith cluster that belong to the jth groundtruth class, k is the number of clusters, and d is the total number of concrete nouns included in the dataset. Purity expresses the fraction of nouns that belong to the true class, which is most represented in the cluster, taking values in the range $[0, 1]$.

4.6.4 Human Sense Classification

For the sensory modality (sense) classification task we use the Sensicon dataset to evaluate the performance of our model regarding sense discrimination. By definition all nouns in Sensicon are concrete nouns since they are associated with a real-world sensory stimulus. Sense classification is performed as described in the subsection 4.6.3, i.e., the similarity matrix in Equation 4.4 is calculated using the

weight vector $\vec{b}$ trained on the MEN dataset and then the spectral clustering [37] is applied for the five sense categories. The resulting clusters are used for sense classification either between two senses, one versus all or among all five senses.

4.6.5 Multisense prediction

A next step is to determine for a given word the degree it interacts with a particular human sense. Neural activations of each word are predicted based on the work proposed by Mitchell [7] described in 4.6. We want to characterize the sense content of words in a continuous valence range regarding their neural activations. We model the valence of each word as a linear combination of its semantic similarities 4.6.2 extracted from our neural our similarity model to a set of seed words and the valence ratings of these words:

$$\hat{s}^{sense}(w_j) = a_0 + \sum_{i=1}^{N} a_i \cdot s(w_i) \cdot d(w_i, w_j) \tag{4.6}$$

where w_j is the word we aim to characterize, $w_1, w_2, ..., w_N$ are the seed words, $u(w_i)$ is the valence rating for seed word,a_i is the weight corresponding to word (that is estimated as described next), d is a measure of semantic similarity between words. The similarity d is extracted using the same method as described in 4.6.3. Assuming we have a training corpus of words with known ratings (SenSicon) and a set of seed words (a subset of the lexicon) for which we need to estimate weights, we can use 4.6 to create a system of linear equations with unknown variables as the weights and the extra weight which is the shift (bias). The optimal weights are found by training our system via Ridge Regression. Once the weights of the seed words are estimated the valence of an unseen word can be computed using 4.6. Note that no additional training corpus or data are required here, the weights are estimated on the corresponding lexicon and are used to bootstrap the model.

4.6.6 Entailment

Next, we applied the neural activations to an entailment classification task. We used a Bi-LSTM model proposed in [142] featuring contextual attention (see Figure 4.6) as our baseline model. Word embeddings for concrete nouns were estimated using GloVe as detailed in [74] and used as input to the Bi-LSTM network. The neural activations vectors were then combined via early fusion (vector concatenation) with GloVe embeddings [74]. Evaluation results for GloVe vectors and neural fusion are shown in Table 4.6 in terms of prediction accuracy.

4.7 Experimental Setup

We built the neural activations prediction model as in [7] using fMRI data for 60 concrete nouns. We calculate the semantic features $f_i(w)$ for each concrete noun, w, by counting its co-occurences with 25 manually selected verbs in a large corpus created in [143] by aggregating results of web queries to Yahoo. The fMRI data used in our experiments were collected and processed by [7] and are publicly available. In this dataset, each one of the participants was presented 60 concrete nouns (for 6 times each) through a line drawing which was labeled with the corresponding noun. Each out of nine subjects, was asked to think about properties of the presented noun during scanning procedure. Finally, a vector representation of the whole cortex neural activation was extracted. Further details about the dataset can be found in [7] and its supplementary website[3]. Prior to training both training and test data are averaged across trials and the final neural activations of each noun are mean normalized. The following datasets have been used for semantic similarity, taxonomy creation, sense classification and entailment tasks. The code of the present work is publicly available[4].

[3] http://www.cs.cmu.edu/~tom/science2008/

[4] https://github.com/athn-nik/neural_asm

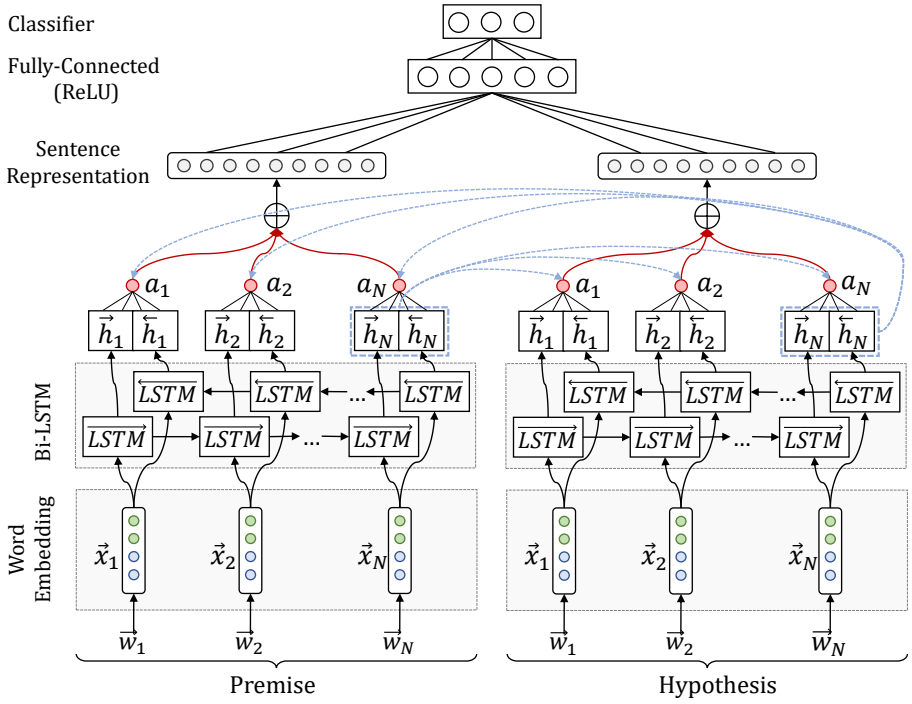


Figure 4.6: Bi-LSTM with context attention used in our experiments. Words' representations are either pretrained word embeddings or fusion of neural activations and word embeddings.

MEN: For the semantic similarity task we train and evaluate our model on the MEN dataset [39] which consists of 3000 word pairs (2000 for training set and 1000 pairs for test set). Each word pair is associated with a similarity score. This score was computed by averaging the similarities that provided by human annotators. We hand-labeled the dataset to keep only concrete nouns because the neural activation prediction model is trained only on concrete nouns. This resulted in 1114 pairs (562 unique words) in the training set and 524 pairs (438 unique words) in the test set. The similarity scores were normalized between zero and one. We also created 2 subsets of MEN of 39 highly similar and 79 highly dissimilar word pairs using a thresholding technique, where pairs with similarity score over 0.85 and under 0.1 belong in the first and second subset respectively.

ESSLLI: For the taxonomy creation task, we evaluate our model on the ESSLLI dataset [36]. It consists of a three-level hierarchy (2-3-6 classes). The lowest level of hierarchy contains 6 classes of concrete nouns (birds, land animals, fruit, greens, vehicles, tools), the middle 3 classes (vegetables, artifacts, animals) while the upper class is distinguished between living beings and objects.

Sensicon: For sense classification, we use the Sensicon [40] dataset. Sensicon is a dictionary which contains 22684 English words and associates each word with 5 numerical scores and a part of speech annotation. The scores correspond to the relevance of the word to each of the 5 senses, namely vision, hearing, taste, smell and touch. In order to use these scores for the sense classification task we selected nouns who have non-zero scores in only one sense results in 1011 words. For the multisense prediction task, SenSicon will be splitted into train,validation,test subsets and evaluation will be performed in test subset using 10-fold cross validation. Because of the size of Sensicon dataset we intuitively selected to include in our evaluation setup words having a consistently large score (at least 0.1). The prementioned setup eventually selected 1180 words.

SNLI dataset: For the entailment task, we used the Stanford Natural Language Inference (SNLI) dataset [41] which contains around 570K sentence pairs with three labels: entailment, contradiction and neutral. We preprocessed the initial dataset to keep only training and testing examples that have at least two or three concrete nouns that are also in the MEN dataset for both premise and hypothesis.

This resulted in 30,498 training and 592 test samples for the case of at least three common words and 171,528 training and 3201 test samples for the case of at least two common words with MEN.

4.8 Experimental Results

4.8.1 Semantic Similarity

For the semantic similarity task, we applied Equation 4.4 for the word pairs of the MEN dataset. The $y_v(.)$s of Equation 4.4 were computed using Equation 4.2 utilizing up to 250 voxels. We exploited the training subset of MEN for learning the b weights of Equation 4.4 using linear regression. Those weights were used for computing the similarities for the test subset of MEN. The Spearman correlation coefficient between the human similarity scores (ground truth) and the similarity scores computed by Equation 4.4 was used as evaluation metric. The results are presented in Table 4.2, where we compare the performance of the proposed neural model (averaged across participants) against the performance yielded by the w2vec word embeddings [25] trained on the GoogleNews corpus.

Subset	Number of voxels	Neural model	w2vec
All Concrete nouns	50	0.43	0.76
	100	0.47	0.76
	150	0.48	0.76
	200	0.48	**0.76**
Most & Least similar	50	0.58	0.73
	100	0.82	0.73
	150	0.82	0.73
	200	**0.88**	0.73
Least similar	50	0.43	0.43
	100	0.44	0.43
	150	0.47	0.43
	200	**0.63**	0.43
Most similar	50	0.28	0.14
	100	0.63	0.14
	150	0.68	0.14
	200	**0.83**	0.14

Table 4.2: Evaluation results on the concrete nouns subset of the MEN test set, and on most and least similar concrete word subsets.

Overall, the w2vec model outperforms the neural model achieving 0.76 correlation on all concrete nouns. For the neural model, performance increases as more voxels are exploited reaching 0.48 correlation is obtained for at least 150 voxels. In Table 4.2, the performance is also shown for three subsets of the MEN test set, namely, "Most & Least similar", "Least similar" and "Most similar" concrete nouns. For all three subsets, the performance achieved by the neural model exceeds[5] the performance of w2vec when at least 100 voxels are used. The performance improvement becomes more pronounced as the number of voxels increases. The best correlation score achieved is 0.88 for the case of the "Most & Least similar" subset for 200 voxels, outperforming the w2vec model (0.73 correlation). Especially for the case of the "Most similar" evaluation subset, we observe a remarkable difference between the two models, i.e., 0.83 vs. 0.14 for the neural and the w2vec model, respectively.

[5] The differences between the similarity scores estimated by our model and the baseline (i.e., w2vec) were found to be statistically significant at 99% level according to paired-sample t-test.

4.8.2 NLP Tasks

Next we present the performance of the neural activation semantic model for a taxonomy creation (semantic class classification), sensory modality (sense) classification, and lexical entailment task. Neural vectors are averaged across participants and evaluated both standalone and in combination (early or late fusion) with traditional word embedding models. In this work, neural fusion refers to early fusion (vector concatenation) of word vectors and neural activation vectors. We used a purity-based metric for evaluating the quality of the automatically created clusters. Purity expresses the fraction of nouns that belong to the true class, which is most represented in the cluster, taking values in the range [0,1].

Dataset	Neural Model	w2vec	Neural Fusion
ESSLI (6 classes)	0.61	0.70	**0.71**
ESSLI (3 classes)	0.77	0.95	**0.95**
ESSLI (2 classes)	0.66	**0.77**	0.72

Table 4.3: Evaluation results for taxonomy creation.

The evaluation results are presented in Table 4.3 for the neural activation model, the w2vec word embeddings [25] trained on the GoogleNews corpus and the late fusion of the two models (denoted as neural fusion) with equal weighting of their similarity matrices S. All results shown are computed on $V = 250$ voxels. The neural model performs worse that the w2vec model in all three subtasks (6, 3 or 2 classes), however, the proposed neural fusion achieves the best purity scores for 6 and 3 classes, at 0.71 and 0.95 purity, respectively. For the case of 2 classes, the best performance is yielded by the w2vec model (0.77). However, the purity of the clusters yielded by the neural model for 2 and 6 classes is comparable with the w2vec's performance. For the sensory modality (sense) classification task we use the Sensicon dataset to evaluate the performance of our model regarding sense discrimination. By definition all nouns in Sensicon are concrete nouns since they are associated with a real-world sensory stimulus.

Sense	Number of voxels	Number of seeds	Neural Model	w2vec
Vision	225	300	0.28	0.27
Audition	100	900	0.57	0.58
Touch	50	500	0.42	0.43
Taste	100	500	0.72	0.71
Smell	225	800	0.48	0.48

Table 4.4: Pairwise evaluation results for sense taxonomy task .

The evaluation results[6] are presented in Table 4.5 for the neural model, the w2vec model (same as the one used in the previous section) and the late fusion of the two (neural fusion) using equal weighting on the similarity matrices S. The evaluation metric used is the purity of clusters defined in Equation 4.5. All results shown are computed on $V = 250$ voxels.

The neural and w2vec models achieve very similar results for two-way classification tasks, with the neural model performing better 0.37 versus 0.33 for five-way sense classification. The neural fusion model outperforms both neural and w2vec models for the majority of the two-way classification tasks and also achieves top performance for the five-way classification task reaching 0.39 purity score. Overall, the neural and neural fusion models show strong performance for this task, which is

[6] Note that the sense smell is not always present in Table 4.5 because smell has only four nouns compared in to other senses that contain more than 100 nouns each.

Classes	Neural Model	w2vec	Neural Fusion
Vision, Audition	0.55	0.55	**0.57**
Vision, Touch	0.68	0.66	**0.69**
Vision, Taste	0.60	0.60	**0.61**
Audition, Touch	0.59	0.58	**0.59**
Audition, Taste	0.57	0.55	**0.57**
Taste, Touch	0.54	0.54	0.54
Vision, Other	0.68	0.68	0.68
Audition, Other	0.74	0.74	0.74
Touch, Other	0.81	0.81	0.81
Taste, Other	0.78	0.78	**0.79**
Audition, Vision, Smell, Touch, Taste	0.37	0.33	**0.39**

Table 4.5: Evaluation results for two-way, one-vs-all, and five-way sense classification.

reasonable given the localization of sensory representations in the human cortex. The results for the human multisense regression task are presented in Table 4.4. We observe that especially for Taste and Audition senses our simple approach achieves results comparable to lexical embeddings. In the cases of Touch, Vision and Smell the two models perform similarly. However, it can be seen that Taste is clearly discriminated by the two models while other senses that may be contained in more abstract words such as Vision and Touch are more difficult to be estimated. Note that, our results are indicative and between the different configurations (seeds, voxels) do not differ more than $1-2\%$. These results are also consistent with neuroscientific research [42, 43, 44, 45, 46, 47]. The results are reported for

Dataset(SNLI)	Dimensions (GloVe, neural)	GloVe	Neural Fusion
3-common	(300,250)	68.2	**68.7**
2-common	(300,250)	76.6	**77.7**

Table 4.6: Entailment task accuracy for GloVe and neural fusion vector input to the Bi-LSTM.

two subsets of the SNLI dataset, namely, 3-common and 2-common (see section 4.7). We observe that the top accuracy is achieved by the fusion scheme for both the 3-common and 2-common subsets ($68.7 \pm 0.9\%$ and $77.7 \pm 0.9\%$). Note that, here we chose to compare neural activations with different state-of-the-art embeddings to extend our evaluation in lexical embeddings with different flavor.

4.9 Further Experimentation

In this section, we will demonstrate some preliminary experiments which outline our ideas for the future directions of our work in Chapter 4. As a first step we wanted to expand the neural predictor beyond concrete nouns and test the performance of our similarity model. Secondly, we tried to investigate the compositional ability of neural activations at a sentential level. A high level illustration of our experimentation can be found in Figure 4.7. Note that the weighted additive schema in compositionality investigation is left as future research. Rank classification evaluation is a method used in [73] which classifies an answer as correct depending on the highest correlation to the actual answer among other alternatives. It is also left as future work.

4.9.1 Data Description

We used a recently outsourced dataset [73] which contains fMRI images of a wide variety of words, sentences and concepts. Specifically, the imaging dataset consists of three different experiments which

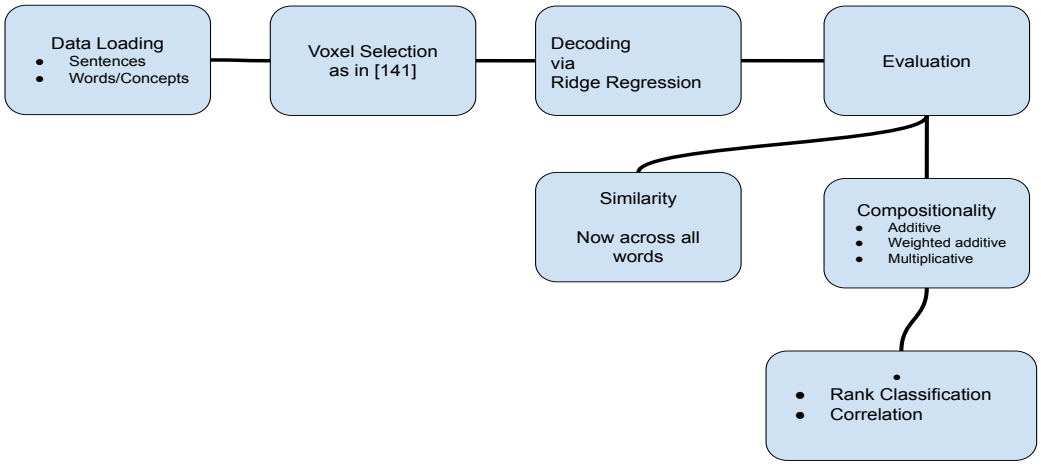


Figure 4.7: Our experimentation procedure on the dataset released in [73].

are described below and are publicly available on the paper website (https://osf.io/crwz7).

Experiment 1

In Experiment 1, words are presented in participants namely 128 nouns, 22 verbs, 29 adjectives, 1 function word. They are presented in three ways:

- In a "cloud" along with other words, while the target word/concept is outlined.
- In a sentence where its meaning is clearly discriminated.
- Via representative images of its meaning and uses.

In our experimentation, we manipulated those three ways separately and on average.

Experiment 2

The second experiment, includes 24 different topics. Every topic is associated with four passages four sentences length each. The passages usually refer to words or concepts that semantically fall into the particular topic. We have one fMRI image per sentence. An example of the data demonstrated to the participants is presented below:

Topic: Fruit	
Passage 1	**Passage 2**
An apple is a fruit that can be green, red or yellow. Apples have thin skin, a crisp, sweet pulp and seeds inside. Some very tart apples are used to make cider. Apples can be eaten raw, roasted or baked in pies.	Banana is a fruit that grows in bunchs, with a soft edible inside. Banana when ripe can be yellow or purple and have small brown spots. The greatest producers are tropical countries, such as India. Unripe bananas and plantains are staple foods and often cooked like potatoes.
Passage 3	**Passage 4**
Peach is an orange-yellow fruit with a characteristic smell. Skin of a peach is thin and covered in small, fine hairs. Peaches have a large, red-brown stone inside which contains the seed. Peaches are sweet and delicate, and must be harvested after ripening.	Raspberry is a fruit that grows in forest clearings or fields. A single raspberry consists of many small fruits joined together. Raspberries are eaten by themselves or cooked with sugar into jam. Leaves of the raspberry are used fresh or dried in herbal teas.

Experiment 3

The third experiment, includes 24 different topics as in Experiment 2. Every topic is associated with

three passages three or four sentences length each. The difference from the previous experiment is that the topic are mainly abstract concepts and all the passages refer to the topic in an explicit way. We also have one fMRI image per sentence. An example of the data shown to every participants is presented below:

Topic: Beekeeping
Passage 1
Beekeeping encourages the conservation of local habitats. It is in every beekeeper's interest to conserve local plants that produce pollen. As a passive form of agriculture, it does not require that native vegetation be cleared to make way for crops. Beekeepers also discourage the use of pesticides on crops, because they could kill the honeybees.
Passage 2
Artisanal beekeepers go to extremes for their craft, but their product is worth the effort. Artisanal honey-making emphasizes quality and character over quantity and consistency. To produce the finest honey, beekeepers become micromanagers of their honeybees. They scout the fields, know when nectar flows, and select the best ways to extract honey.
Passage 3
As the beekeeper opens the hive, the deep hum of 40,000 bees fills the air. The beekeeper checks honey stores, pollen supplies, and the bee nursery. Bees crawl across his bare arms and hands, but they don't sting, because they're gentle.

Their experiment in [73] focus on how different topics and words differ in terms of neural activations, examining all three different experiments. They first used a neural predictor similar to [7] to map the lexical space to the neural space. Then, they tested the learned neural representations behavior in different topic and words discriminative tasks.

4.9.2 Abstract Concepts Decoding

In order to determine if the neural predictor can help our similarity model to generalize in words other than concrete nouns we used the fMRI images of Experiment 1 and trained our neural predictor. Then, we tested our similarity model on the whole MEN dataset.

Subject	Wordcloud	Pictures	Sentences	Average	Glove
M01	0.37	0.35	0.37	0.35	0.74
M02	0.41	0.4	0.46	0.35	0.74
M03	0.37	0.39	0.37	0.39	0.74
M04	0.36	0.34	0.36	0.39	0.74
M05	0.33	0.38	0.33	0.4	0.74
M06	0.37	0.37	0.37	0.33	0.74
M07	0.25	0.38	0.43	0.3	0.74
M08	0.39	0.36	0.39	0.45	0.74
M09	0.42	0.39	0.38	0.35	0.74
M10	0.32	0.34	0.33	0.4	0.74
M13	0.36	0.315	0.14	0.39	0.74
M14	0.37	0.41	0.38	0.33	0.74
M15	0.34	0.42	0.39	0.34	0.74
M16	0.35	0.36	0.39	0.39	0.74
M17	0.44	0.3	0.37	0.38	0.74
P01	0.36	0.39	0.45	0.37	0.74
AVG	0.36	0.37	0.37	0.37	0.74

Table 4.7: Evaluation results across all words for MEN similarity dataset.

In Table 4.7, spearman correlation results for the whole MEN dataset are presented. We observe that lexical word embeddings perform better in all cases.

Subject	Wordcloud		Pictures		Sentences		Average		Glove	
	Low	High	Low	High	Low	High	Low	High	Low	High
M01	-0.64	-0.05	-0.07	0.19	0.55	0.23	0.23	0.12	0.43	0.35
M02	0.15	-0.66	0.85	-0.43	-0.43	-0.04	0.14	0.52	0.43	0.35
M03	0.76	0.21	0.84	0.44	0.15	0.46	0.28	-0.72	0.43	0.35
M04	0.76	-0.14	-0.18	0.16	0.15	-0.16	0.64	-0.56	0.43	0.35
M05	0.99	0.66	-0.06	0.71	0.99	0.65	-0.21	0.77	0.43	0.35
M06	-0.46	-0.05	0.48	-0.59	-0.46	-0.05	-0.9	-0.14	0.43	0.35
M07	0.81	-0.07	-0.99	0.4	0.78	-0.35	0.55	-0.13	0.43	0.35
M08	-0.83	-0.21	0.49	0.22	0.43	0.35	0.16	0.22	0.43	0.35
M09	0.16	0.33	-0.44	-0.67	-0.78	-0.18	0.15	-0.36	0.43	0.35
M10	-0.59	-0.21	0.16	0.64	0.22	0.69	0.15	0.26	0.43	0.35
M13	-0.83	0.12	-0.18	-0.94	-0.12	0.05	0.16	-0.24	0.43	0.35
M14	0.78	0.18	0.28	-0.06	0.78	0.18	-0.9	0.68	0.43	0.35
M15	-0.95	0.27	-0.43	0.28	-0.74	0.22	-0.22	-0.12	0.43	0.35
M16	0.90	-0.25	0.55	0.29	0.49	0.56	0.43	0.12	0.43	0.35
M17	-0.27	-0.23	-0.95	0.1	0.21	0.34	-0.1	-0.6	0.43	0.35
P01	0.21	-0.25	0.07	-0.3	-0.74	0.46	0.98	-0.06	0.43	0.35

Table 4.8: Evaluation results on the low and high similarity subsets of the MEN dataset.

Next, we experimented with the highly similar and dissimar subsets of MEN dataset. In Table 4.8, spearman correlation results for the the high and low similar subsets of MEN dataset are shown. We can see that for some participants and specific data subsets neural activations outperform lexical embeddings. However, they demonstrate extreme variations making it infeasible to reach a general conclusion.

4.9.3 Compositionality of words in Brain

Semantic composition is one part of a larger cognitive process termed semantic unification. Semantic unification includes not only composing the meaning of words in phrases, but also phrases in sentences, and sentences in larger thematic structures. However, most of the studies were performed in EEG data examining regional characteristics of semantics understanding and how particular EEG components are related to syntactic or semantic characteristics of language [144, 145, 29, 146, 147]. As a preliminary experimentation step in examination of sentence compositionality we calculated Spearman correlation of a neural representation of sentence extracted by its words —by using the trained model on the 180 words— compared with its actual neural representation. We chose to combine word neural activation either additively or multiplicatively.

Subject	Multiplicative	Additive
M02	0.03	-0.04
M03	0.11	0.24
M04	0.01	0.04
M07	0.02	0.04
M15	0.003	0.01
P01	0.44	0.61

Table 4.10: Evaluation results on sentence compositionality for Experiment 3.

Subject	Multiplicative	Additive
M02	0.03	0.04
M04	0.02	0.01
M07	0.05	0.07
M08	0.01	0.02
M09	-0.01	-0.03
M14	-0.03	-0.04
P01	0.04	0.09

Table 4.9: Evaluation results on sentence compositionality for Experiment 2.

In Tables 4.9, 4.10 the Spearman correlation of performance of our method is presented. Although, P01 subject appears to have a higher correlation demonstrating a compositional tendency according to our simple method, the other subjects do not validate the same hypothesis. Their correlation values indicate almost no correlation between the word combined and the actual sentence's neural activations representation.

4.10 Experimental Summary & Discussion

The analysis of the neural model word performance showed that the proposed model can differentiate between very similar and very dissimilar concrete nouns better than state-of-the-art word embeddings semantic models, while it performs worse overall for the word semantic similarity task. This is a strong indication that the semantic discriminability of neural activation vectors is of a different flavor than that of traditional word embedding vectors, and thus neural vectors can be used to augment state-of-the-art semantic representations.

Results on the taxonomy classification, sense classification and entailment task indeed verify the different flavor of neural embeddings. For certain tasks, e.g., sense classification, neural models provide state-of-the-art performance. For other tasks, the fusion of neural and word2vec embeddings provides significant improvement.

Overall, (predicted) localized neural activation vectors can also be used in conjunction with other semantic representations and deep architectures to improve the results in challenging tasks, like concept entailment.

Chapter 5

Cross-topic Distributional Representations

In this chapter we present our work in 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics which was made in cooperation with a former group member Eleftheria Briakou [2].

5.1 Multiple Embedding Models via cross unsupervised mappings

In traditional Distributional Semantic Models (DSMs) the multiple senses of a polysemous word are conflated into a single vector space representation. In this work, we propose a DSM that learns multiple distributional representations of a word based on different topics. First, a separate DSM is trained for each topic and then each of the topic-based DSMs is aligned to a common vector space. Our unsupervised mapping approach is motivated by the hypothesis that words preserving their relative distances in different topic semantic sub-spaces constitute robust *semantic anchors* that define the mappings between them. Aligned cross-topic representations achieve state-of-the-art results for the task of contextual word similarity. Furthermore, evaluation on NLP downstream tasks shows that multiple topic-based embeddings outperform single-prototype models.

5.2 Motivation

Current word representation learning models encode the semantic and syntactic information of words adopting the distributional hypothesis [19]. Word-level representation algorithms encode contextual information of words into dense feature vectors (embeddings). However, such models(w2vec, Glove, fasttext) learn single point representations, which cannot capture the distinct meanings of polysemous words (e.g., *bank* or *book*). This leads to conflated word representations of diverse contextual semantics. Thus, the creation of multi-sense embeddings, which encode different word meanings in the semantic space can help us to improve natural language understanding.

5.3 Related Work

Methods that assign multiple distributed representations per word can be grouped into two broad categories.[1] Unsupervised methods induce multiple word representations without leveraging semantic lexical resources. [48] were the first to create a multi-prototype DSM with a fixed number of vectors assigned to each word. In their model, the centroids of context-dependent clusters were used to create a set of "sense-specific" vectors for each target word. Based on similar clustering approaches, follow-up works introduced neural network architectures that incorporated both local and global context in a joint training objective [49], as well as methods that jointly performed word sense clustering and embedding learning as in [50, 51]. A probabilistic framework was introduced by [52], where the Skip-Gram model of Word2Vec was modified to learn multiple embedding vectors. Furthermore, latent topics were integrated into the Skip-Gram model, resulting in topical word embeddings which modeled the semantics of a word under different contexts [53, 54, 55]. Another topic-related embedding creation

[1] We limit our discussion to related works that use monolingual DSMs and corpora.

approach was proposed in [90] where a mixture of topic-based semantic models was extracted by topical adaptation of in-domain corpora. Other approaches used autoencoders [66], convolutional neural networks designed to produce context representations that reflected the order of words in a context [117] and reinforcement learning [67, 68].

Supervised approaches, based on prior knowledge acquired by sense inventories (e.g., WordNet) along with word sense disambiguation algorithms, were also introduced for sense-specific representations extraction [56, 57]. In other works, pre-trained word embeddings have been extended to embeddings of lexemes and synsets [88] or were de-conflated into their constituent sense representations [89] by exploiting semantic lexical resources.

5.4 Unified Multi-Topic Model

Our system follows a four-step approach which can be visualized in Figure 5.1:

1. **Global Distributional Semantic Model.** Given a large collection of text data we train a DSM that encodes the contextual semantics of each word into a single representation, also referred to as Global-DSM.

2. **Topic-based Distributional Semantic Models.** Next, a topic model is trained using the same corpus. The topic model splits the corpus into K (possibly overlapping) sub-corpora. A DSM is then trained from each sub-corpus resulting in K topic-based DSMs (TDSMs). The topical adaptation of the semantic space takes into account the contextual variations a word exhibits under different thematic domains and therefore leads to the creation of "topic-specific" vectors (topic embeddings).

3. **Mappings of topic embeddings.** Next, we map the vector space of each topic-based DSM to the shared space of the Global-DSM, using a list of anchor words selected through an unsupervised self-learning scheme. In the unified semantic space each word is represented by a set of topic embeddings that were previously isolated in distinct vector spaces, thus creating a Unified multi-Topic DSM (UTDSM).

4. **Smoothing of topic embeddings.** As an optional step, we employ a smoothing approach in order to cluster a word's topic embeddings into N Gaussian distributions via a Gaussian Mixture Model (GMM). This step lessens the noise introduced to our system through the semantic mappings and sparse training data.

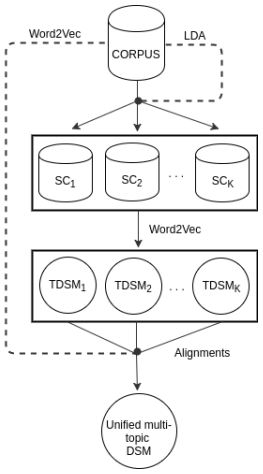


Figure 5.1: Starting from an initial corpus, K topic subcorpora (SC_i) are created and subsequently K topic embedding spaces are created ($TDSM_i$) which are then projected in a the global space.

5.4.1 Creation of Topic Spaces

The first step towards the thematic adaptation of the semantic space is the induction of in-domain corpora, using the Latent Dirichlet Algorithm (LDA) [58]. LDA is a generative probabilistic model of a corpus. Its core idea is that documents are represented as random mixtures over topics; where each topic is defined as a probability distribution over a collection of words. Given as input a corpus of documents, LDA trains a topic model and creates a distribution of words for each topic in the corpus. Using the trained LDA model, we infer a topic distribution for each sentence in the corpus. Afterward, following a soft clustering scheme each sentence is included in a topic-specific corpus when the posterior probability for the corresponding topic exceeds a predefined threshold. The resulting topic sub-corpora are then used to train topic-based DSMs. Any of the DSM training algorithms proposed in the literature can be used for this purpose; in this paper, we opt for the Word2Vec model [25].

5.4.2 Mapping across different Topic Spaces

The intrinsic non-determinism of the Word2Vec algorithm leads to the creation of continuous vector spaces that are not naturally aligned to a unified semantic reference space, precluding the comparison between words of different thematic domains. To circumvent this limitation, we need to map the word representations of TDSMs to a shared vector space. In particular, we hypothesize that TDSMs capture meaningful variations in usage of polysemous words, while the relative semantic distance between monosemous words is preserved. This hypothesis motivated us to think of monosemous words as *anchors* between semantic spaces, as illustrated in Figure 5.2. One way to retrieve the list of anchors is to extract monosemous words from lexical resources such as WordNet [59]. However, this method is restricted to languages where such lexical resources exist and depends on the lexical coverage and quality of such resources.

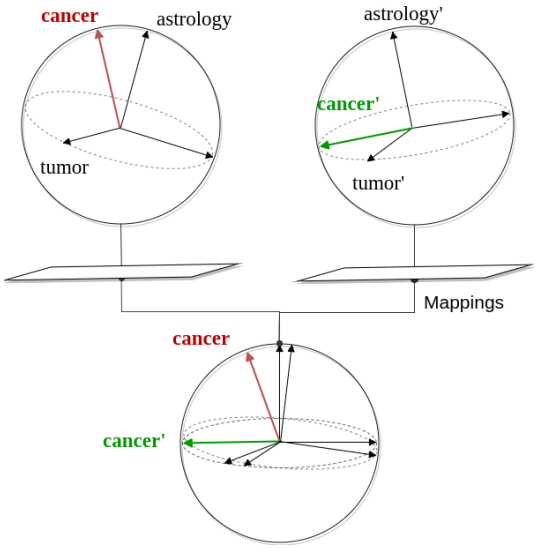


Figure 5.2: Simplified depiction summarizing the intuition behind the alignment process of topic embeddings. In the unified vector space, the polysemous word *cancer* is represented by two topic vectors that capture different semantic properties of the word under a zodiacal and a medical topic. Words *astrology* and *tumor* are examples of *semantic anchors* that define the mappings.

To overcome the above limitations, we propose a fully unsupervised method for semantic anchor induction. Although the embeddings of the topic and global semantic vector spaces are not aligned, their corresponding similarity matrices (once normalized) are. Based on this observation, we compute the similarity between a given word and every other word in the vocabulary (similarity distribution) for the different topic and global spaces. Then, we assume that good semantic anchors should have

similar similarity distributions across the topic-specific and the global space, as illustrated in Figure 5.3. [148] was based on a similar observation to align vector semantic spaces in bilingual machine translation context.

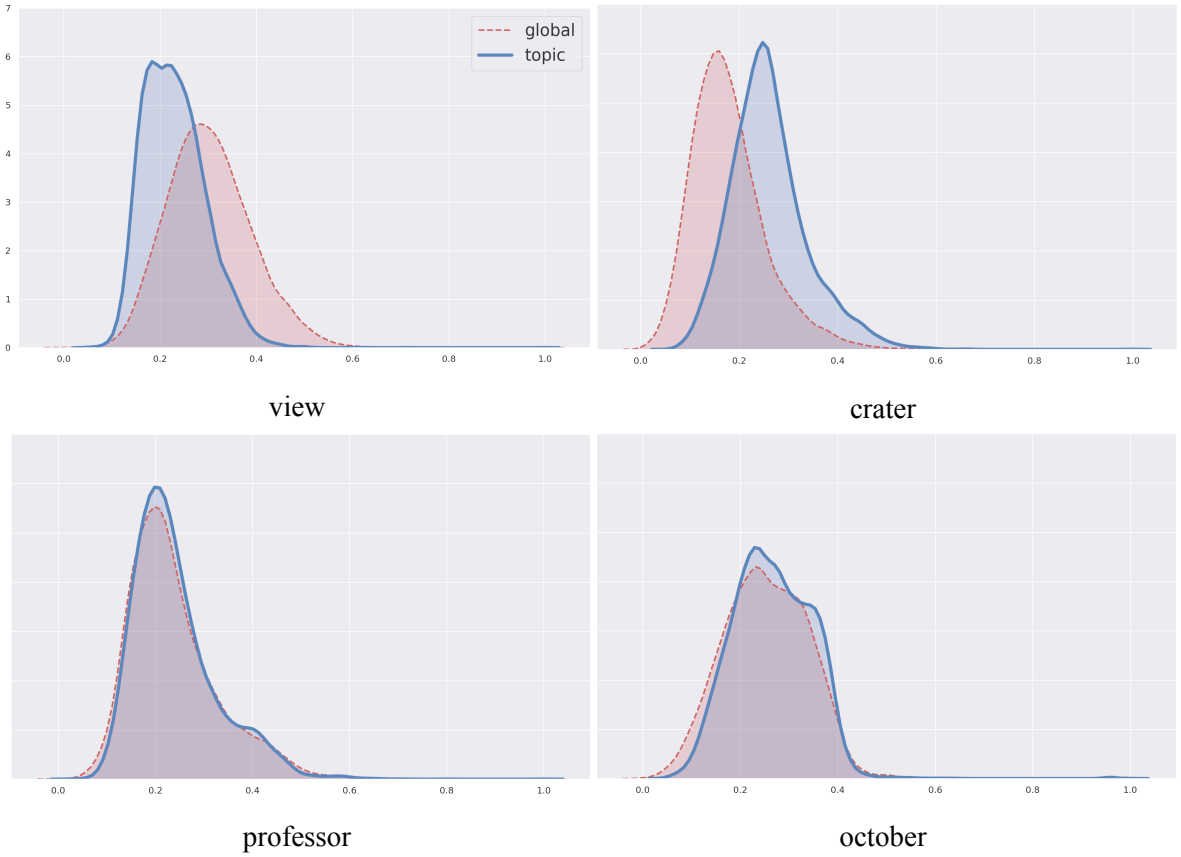


Figure 5.3: Similarity distributions of four different words (corresponding to the smoothed density estimates of the similarity matrices) in topic domain space as defined in Equation 5.1 and global space s_g^i. Selected anchors ("professor" and "october") have more similar distributions in the global and topic spaces, when compared to unselected ones ("view" and "crater"). We observe that the selected anchors are less ambiguous, while the not selected ones are expected to have diverse contextual semantics.

Let V be the intersection of the Global-DSM and the K TDSMs vocabularies and d the embedding dimension. We then define $X_k \in \mathbb{R}^{|V| \times d}$ as the embedding matrix of the k-th TDSM, and $Y \in \mathbb{R}^{|V| \times d}$ as the embedding matrix of the global DSM, where the i-th row of each matrix corresponds to the unit normalized representation of a word in V. Then, we define $S_k = X_k X_k^T$, $S_g = YY^T \in \mathbb{R}^{|V| \times |V|}$ to be the similarity distribution matrices for the k-th TDSM and the global-DSM, respectively. Our objective is to extract a list of semantic anchors A that minimizes the Euclidean distance between the two different similarity distributions. Specifically, for every word i we calculate the average semantic distribution across all topics:

$$< s_k^i >_k = \frac{1}{K} \sum_{k=1}^{K} s_k^i \tag{5.1}$$

$$\| < s_k^i >_k \; - s_g^i \|_2 \, , \quad \forall \, i = 1, \ldots, |V| \tag{5.2}$$

where s_g^i, s_k^i is the i-th row of the S_g and S_k similarity matrix, respectively, representing the similarity distribution between word i and every other word in the vocabulary V. We then choose $|A|$ anchors as the words with the smallest values according to criterion 5.2. Furthermore, we assume that there exists an orthogonal transformation matrix between the topic embeddings of the extracted

semantic anchors of each TDSM (source space) and the corresponding representations of the global-DSM (target space). The orthogonality constraint on the transformation matrix is widely adopted by the literature for various semantic space alignment tasks [60, 61, 62]. Assume $\alpha_k^j \in \mathbb{R}^d$ is the vector representation of the j-th *anchor* word in the source space and $\alpha_g^j \in \mathbb{R}^d$ is its corresponding vector representation in the target space. The transformation matrix $M_k \in \mathbb{R}^{d \times d}$ that projects the first space to the latter is learned via solving the following constraint optimization problem:[2]

$$\min_{M_k} \sum_{j=1}^{|A|} \|M_k \alpha_k^j - \alpha_g^j\|_2^2, \quad \text{s.t.} \quad M_k M_k^T = \mathbb{I} \tag{5.3}$$

The induction of multiple topic embeddings in the unified vector space is achieved via applying Equation 5.3 to each TDSM. Specifically, given a word and its k-th topic distributed representation $x_k \in \mathbb{R}^d$, we compute its projected representation $x'_k \in \mathbb{R}^d$ as follows:

$$x'_k = M_k x_k \tag{5.4}$$

5.4.3 Clustering of Topic Embeddings

Starting from the set of aligned topic embeddings $\{x'_k\}_{k=1}^K$ for each word, we learn a Gaussian Mixture Model with N components, where closely positioned topic embeddings are assigned to the same component. This step operates as an implicit way of segmenting the space of topic embeddings for each word in order to capture more useful hyper-topics—union of topics—which better represent their different meanings. We suggest that each Gaussian distribution forms a semantically coherent unit that corresponds to closely related semantics of the target word. Subsequently, the mean vector of each Gaussian distribution is used as a representative vector of each component, leading to a new set of *smoothed* topic embeddings $\{x_n^*\}_{n=1}^N$ for each word, where $x_n^* \in \mathbb{R}^d$.

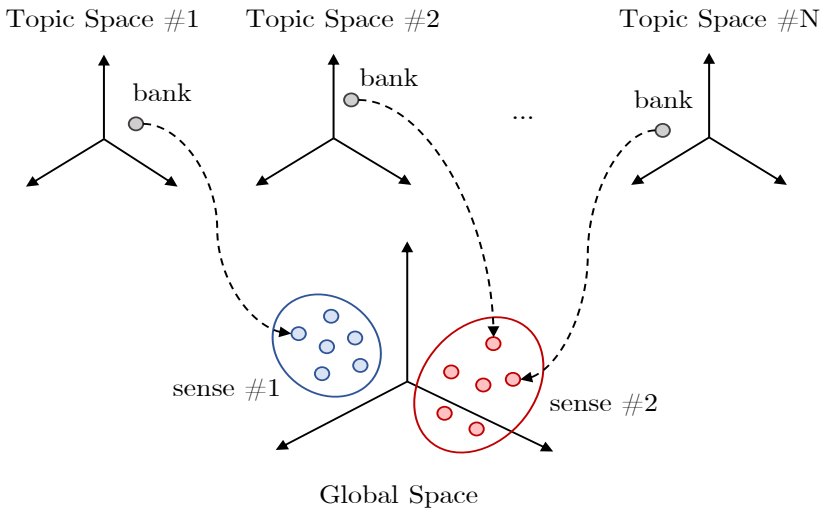


Figure 5.4: Mapping of topic embeddings. The embeddings from each topic space are projected to the global space. The embeddings of each word are then clustered together, forming the corresponding sense clusters.

[2] This problem is known as the orthogonal Procrustes problem and it has a closed form solution as proposed in [63].

5.5 Experimental Setup

5.5.1 DSM Settings

As our initial corpus we used the English Wikipedia, containing 8.5 million articles [149]. During the training of the topic model, we used the articles found in the Wikipedia corpus and employed the Gensim implementation of LDA [150] setting the number of topics K to 50. Using a threshold of 0.1, we followed a soft-clustering approach, to bootstrap the creation of topic sub-corpora, using our trained topic model. Finally, we used Gensim's implementation of Word2Vec and Continuous Bag-of-Words method to train both the global-DSM and the TDSMs. The context window parameter of Word2Vec is set to 5, while the dimensionality d of all the constructed DSMs is equal to 300 or 500.[3]

5.5.2 Semantic Anchors

The number of *semantic anchors* that determine the mappings between our source and target spaces is set to $|A| = 5\,000$ [4] according to our unsupervised approach (criterion 5.2). Those are selected from the common set of words that are represented in all semantic spaces with $|V| \sim 12\,000$.

As a second experiment, we randomly sample $|A|$ words from the vocabulary of each TDSM to define its transformation matrix. We repeat this experiment 10 times, every time sampling a different list from the corresponding vocabulary and report average performance results.

5.5.3 Gaussian Mixture Model

To apply the smoothing technique on the set of a word's topic embeddings we use the Scikit-learn implementation of Gaussian Mixture Model clustering algorithm [65]. We initialize the mean vector of each component using k-means algorithm and the parameters of the model are estimated using Expectation-Maximization (EM) algorithm.

5.5.4 Semantic Contextual Word Similarity Dataset

To estimate the semantic similarity between a pair of words provided in sentential context, we use the standard evaluation Stanford Contextual Word Similarity (SCWS) [49] dataset which consists of $2\,003$ word-pairs with assigned semantic similarity scores computed as the average estimations of several human annotators. Following the evaluation guidelines proposed in literature, we employ the AvgSimC and MaxSimC contextual metrics, firstly discussed in [48]. In particular, given the word-pair (w, w'), and their provided contexts (c, c') we define:

$$\text{AvgSimC}(w, w') = \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} \text{p}(j|w, c)\text{p}(k|w', c')\text{d}(x'_j(w), x'_k(w')), \tag{5.5}$$

$$\text{MaxSimC}(w, w') = \text{d}(\hat{x'}(w), \hat{x'}(w')), \tag{5.6}$$

$$\hat{x'}(w) = u_{\text{argmax}_{1 \leq j \leq K}\, \text{p}(j|w,c)}(w) \tag{5.7}$$

An example following the notation used in 5.4.2, K is the number of topics returned by the trained LDA model, x'_j is the word embedding trained on the sub-corpus corresponding to the j-th topic

[3] Any parameter not mentioned is set to default values of the corresponding implementations (e.g., Word2Vec, Gensim LDA).

[4] We have experimented with different values of anchors from $\{1\,000, 2\,000, 3\,000, 4\,000, 5\,000\}$ and report results for the best setup.

after being projected to the unified vector space, $\mathrm{p}(j|w,c)$ denotes the posterior probability of topic j returned by LDA given as input the context c of word w, d denotes the cosine similarity between the two input representations and finally $\hat{x'}(w)$ is the vector representation of word w that corresponds to the topic with the maximum posterior for c. Intuitively, a higher score in MaxSimC indicates the existence of more robust multi-topic word representations. On the other hand, AvgSimC provides a topic-based smoothed result across different embeddings. A simplified example taken from the dataset can be found below:

war ***battle***
...modern French Army retains two Dragoons regiments from the 32 it possessed at the beginning of World ***War*** *I : the 2nd, which is a nuclear, bacteriologic and chemical protection regiment, and the 13th, which is a special-ops parachute...*

Battle *of Namakura, it would be one of Britain's numerous embarrassing colonial defeats of the war. The largest sieges of the war, however, took place in Europe. The initial German advance into Belgium produced four major sieges, the* ***Battle*** *of Liege...* $\rightarrow$9.08

war ***hostility***
...was striving to drag all of the Arab countries into a ***war****. After the Samu raid, these apprehensions became the deciding factor in Jordan's decision to participate in the* ***war****. King Hussein was convinced Israel would try to occupy the West Bank whether Jordan went to* ***war****, or not. Israel and Syria. In addition to...*

... history of the Talmud reflects in part the history of Judaism persisting in a world of ***hostility*** *and persecution. Almost at the very time that the Babylonian...* $\rightarrow$ 5.6

5.5.5 Downstream NLP Tasks Datasets

Text classification. We used the 20NewsGroup[5] dataset, which consists of about 20 000 documents. Our goal is to classify each document into one of the 20 different newsgroups based on its content.
Paraphrase Identification. For this task we aimed at identifying whether two given sentences can be considered paraphrases or not, using the Microsoft Paraphrase dataset [64].

5.5.6 NLP Tasks

Besides the standard evaluation benchmark of contextual word similarity, we also investigate the effectiveness of our mapped cross-topic embeddings on document and sentence level downstream NLP tasks: text classification and paraphrase identification. We report weighted-averaging precision, recall, F1-measure and accuracy performance metrics.
Document and Sentence level representations.
Given a document or a sentence D, where w_d corresponds to the d-th word in D, we extract its feature representation using three different ways:

$$\mathrm{AvgC_D} = \frac{1}{|D|}\sum_{d=1}^{|D|}\sum_{k=1}^{K}\mathrm{p}(k|D)x'_k(w_d), \tag{5.8}$$

$$\mathrm{Avg_D} = \frac{1}{|D|}\sum_{d=1}^{|D|}\sum_{k=1}^{K}\frac{1}{K}x'_k(w_d), \tag{5.9}$$

[5] http://qwone.com/ jason/20Newsgroups/

$$\text{MaxC}_\text{D} = \frac{1}{|D|}\sum_{w=1}^{|D|} x'_m(w_d)$$
$$\text{s.t.} \quad m = \underset{k=1,..,K}{\text{argmax}}\{\text{p}(k|D)\}, \tag{5.10}$$

where $\text{p}(k|D)$ denotes the posterior probability of topic k returned by LDA given as input the sentence/document D and $x'_k(w_d)$ is the mapped representation of word w_d for topic k. For the case of paraphrase identification, we extract a single feature vector for each sentence-pair via concatenating the features of the individual sentences.

After feature extraction, we train a linear Support Vector Classifier (SVM) [65] using the proposed train/test sets for both tasks. We report the best results for each experimental configuration after tuning the SVM's penalty parameter of the error term using 500-dimensional word embeddings.

5.6 Experimental Results & Discussion

5.6.1 Contextual Similarity

In Table 5.1 we compare our model (UTDSM) with our baseline (Global-DSM) and other state-of-the-art multi-prototype approaches for the contextual semantic similarity task. It is clear that all different setups of UTDSM perform better than the baseline for both contextual semantic similarity metrics. Using a single Gaussian distribution (UTDSM + GMM (1)) at the smoothing step of our method produces similar results to the baseline model. This is anticipated as both methods provide a centroid representation of a word's diverse semantics. In terms of MaxSimC the model consistently yields higher performance when the list of semantic anchors is induced via our unsupervised method instead of using randomly selected anchor words (UTDSM Random). We also observe that random anchoring performs slightly worse than UTDSM with respect to AvgSimC. This result validates our hypothesis that the representations of words, which share consistent similarity distributions across different topic domains, constitute informative *semantic anchors* that determine the mappings between semantic vector spaces.

Furthermore, we observe that GMM smoothing has a different effect on the MaxSimC and AvgSimC metrics. Specifically, for AvgSimC we consistently report lower results when GMM smoothing is applied for different number of components. We attribute this behavior to a possible loss of model capacity—decrease in the number of topic embeddings—that is capable of capturing additional topic information. At the same time, our smoothing technique highly improves the performance of MaxSimC for all possible configurations. Given that this metric is more sensitive to noisy word representations, this result indicates that our technique lessens the noise introduced to our system and captures finer-grained topic senses of words.

Overall, the performance of our model is highly competitive to the state-of-the-art models in terms of AvgSimC, for 500-dimensional topic embeddings. We also achieve state-of-the-art performance for the MaxSimC metric, using smoothed topic embeddings of 300 or 500 dimensions with 2 or 3 Gaussian components.

5.6.2 NLP Tasks

Besides the standard evaluation benchmark of contextual word similarity, we also investigate the effectiveness of our mapped cross-topic embeddings on document and sentence level downstream NLP tasks: text classification and paraphrase identification. We report weighted-averaging precision, recall, F1-measure and accuracy performance metrics.

Evaluation results on text classification are presented in Table 5.2. We observe that our model performs better than the baseline across all metrics for both averaging approaches (AvgC_D, Avg_D),

Method	AvgSimC	MaxSimC
Liu et. al(2015)[54]	67.3	68.1
Liu et. al(2015b)[53]	69.5	67.9
Amiri et. al(2016)[66]	70.9	-
Lee et. al(2017)[67]	68.7	67.9
Guo et. al(2018)[68]	69.3	68.2
300-dimensions		
Global-DSM	67.1	67.1
UTDSM Random	69.1 ± 0.1	66.4 ± 0.2
UTDSM	**69.6**	67.1
UTDSM + GMM (1)	67.4	67.4
UTDSM + GMM (2)	68.4	**68.3**
UTDSM + GMM (3)	68.9	**68.3**
UTDSM + GMM (8)	69.1	68.0
UTDSM + GMM (10)	69.0	67.8
500-dimensions		
Global-DSM	67.6	67.6
UTDSM Random	69.4 ± 0.1	66.5 ± 0.3
UTDSM	**70.2**	68.0
UTDSM + GMM (1)	67.6	67.6
UTDSM + GMM (2)	68.8	**68.6**
UTDSM + GMM (3)	69.0	68.5
UTDSM + GMM (8)	69.5	68.5
UTDSM + GMM (10)	69.2	68.0

Table 5.1: Performance comparison between different state-of-the-art approaches on SCWS, in terms of Spearman's correlation. UTDSM refers to the projected cross-topic representation, UTDSM Random refers to the case when random words served as anchors and GMM (c) corresponds to GMM smoothing with c components.

Method	Precision	Recall	F1-score	Accuracy
LDA	39.7	41.8	38.8	41.8
Global-DSM	62.9	63.3	62.9	63.3
MaxC_D	61.9	63.0	62.0	63.0
Avg_D	63.5	64.6	63.3	64.3
AvgC_D	**64.6**	**65.5**	**64.5**	**65.5**

Table 5.2: Evaluation results of multi-class text classification.

while the usage of dominant topics appears to have lower performance (MaxC_D). Specifically, we get an improvement of $2 - 2.5\%$ on topic-based average and $0.5 - 1\%$ on simple average combination compared to using Global-DSM. The performance difference between the dominant and the average topic arises because the topics discovered by LDA algorithm is possibly different than the different topics of the specific dataset. Hence, a smoothed mixture of vectors achieves better results than choosing vectors based on the highest topic probability, thus, ignoring topics which may have similar probabilities and have a major contribution to the result.

Method	Precision	Recall	F1-score	Accuracy
Global-DSM	68.6	69.2	62.0	69.2
MaxC_D	**69.0**	69.3	62.1	69.3
Avg_D	67.7	**69.4**	**64.0**	**69.4**
AvgC_D	68.8	69.4	62.6	69.4

Table 5.3: Evaluation results on paraphrase detection task.

Results for the paraphrase identification task are presented in Table 5.3. Avg_D yields the best results especially in F1 metric showing that cross-topic representations are semantically richer than single embeddings baseline (Global-DSM). Although we apply the topic distributions $p(k|D)$ extracted from LDA (document-level model) to a sentence-level task, improvements over the baseline are also shown in the AvgC_D and MaxC_D cases. "Hard" vector choice performs better than other methods in precision metric which can be attributed to the fact that paraphrase identification examples are significantly shorter and condensed in meaning than those of text classification.

Overall, the proposed UTDSM model outperforms the baseline Global-DSM model on contextual semantic similarity and downstream tasks. Specifically, our smoothing approach improves our results in MaxSimC which is more noise-dependent, while it slightly affects AvgSimC. In downstream task evaluation, we observe that in the case of text classification average combination methods achieve better results which means that multiple topic vectors contribute to overall topic selection. In paraphrase identification, selection of the vector with the highest probability improves precision results, while still averaging methods perform better in other metrics. In almost all cases, except for the MaxC_D in text classification task, our multiple embeddings perform better than single-representation model.[6]

5.7 Cross Domain Analysis

In this section, we are going to perform a qualitative analysis of our results, visualize our model results and examine the impact of our alignment for specific words and semantic neighborhoods.

5.7.1 Semantic Neighborhoods

Finally, we carry out a cross-domain semantic analysis to detect the variations of a word's meaning in different topic domains. To that end, we use a list of known polysemous words and measure the semantic similarity between different topic representations of the same ambiguous word. The ultimate goal of this analysis is to validate that our model captures known thematic variations in semantics of polysemous words.

Word	Topic Words	Nearest Neighbors	Similarity
drug	health, medical, cancer, treatment, disease	insulin, therapy, heparin, chemotherapy, vaccines	0.61
	drug, health, marijuana, alcohol, effects	meth, cocaine, methamphetamine, mdma, heroin	
act	law, court, legal, tax, state	bylaw, legislature, complying, entities, entitlement	0.39
	music, guitar, piano, dance, theatre	touring, pantomime, weekend, shakespeare, musical	
python	garden, plant, fish, bird, animal	macaw, crocodile, hamster, albino, rattlesnake	0.27
	software, forum, download, windows, web	algorithm, parser, notepad, gui, tutorial	
rock	mountain, river, park, road, trail	geology, slab, limestone, waterfalls, canyon	0.43
	music, guitar, piano, dance, theatre	touring, acoustic, americana, songwriter, combo	
nursery	garden, plant, tree, flower, gardening	camellias, succulents, greenhouse, ornamental, grower	0.46
	university, school, college, education, program	prep, montessori, grammar, preschool, infant	

Table 5.4: Examples of polysemous words and the change of meaning between different topic domains. First column lists the example target words. Second column includes the most probable words of the topic domains —a distribution over words— these words are assigned to. Each row corresponds to a different topic domain. Third column shows the nearest monosemous neighbors of the target word in the corresponding topic domain. The last column corresponds to the cosine similarity between the two topic representations of the target word.

Table 5.4 includes examples of our analysis. The most probable words of the topics (second

[6] Similar results were obtained for each metric using smoothed word embeddings. Also, there are no standard evaluation approaches for comparison of previous works on downstream tasks.

column) give an intuitive sense of their major contexts, while their nearest neighbors (third column) infer the sense of the target word in the corresponding topic domain. Specifically, we observe that the word *python* shifts from meaning "snake" in a topic about animals and nature, to referring to a "programming language" under a topic about computers. Word *drug* is mostly related to "medication" in a broad medical domain; it experiences though a slight shift from this meaning when it resides in a topic about "illegal substances". The highly polysemous word *act* shifts from meaning "statute" to meaning "performance" under the corresponding law and art topics. In a thematic domain about music the word *rock* refers to a "music style" while in a more broad context about nature it refers to "stone". Finally, the word *nursery* corresponds to a "childcare facility" in a topic about education, whereas its meaning changes to "seedbed" in a topic about plants. Moreover, in Figure 5.5 we visualize the latent semantic space of the neighborhoods for the two discriminative senses of word *python*, using principal component analysis. By examining the local neighborhoods of the words subjected to analysis, we show that our model produces meaningful results that reflect the expected topic semantics of words.

5.7.2 Visualizing Semantic Variance

Finally, we carry out a cross-domain semantic analysis to detect the variations of a word's meaning in different topic domains. To that end, we use a list of known polysemous words and measure the semantic similarity between different topic representations of the same ambiguous word. The ultimate goal of this analysis is to validate that our model captures known thematic variations in semantics of polysemous words.

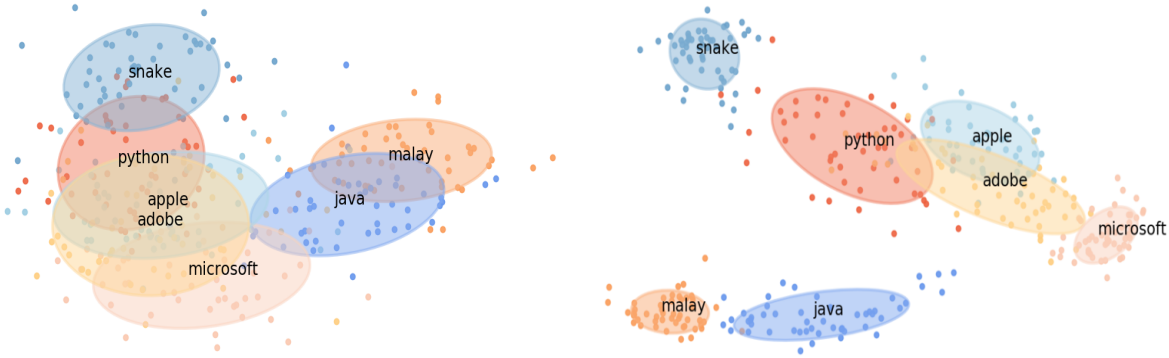


Figure 5.5: A 2-dimensional projection—using PCA—of the latent semantic space encoded in our unified vector space model, depicting the topic word representations of 7 words before (left) and after (right) mapping the TDSMs to the global semantic space.

Table 5.4 includes examples of our analysis. The most probable words of the topics (second column) give an intuitive sense of their major contexts, while their nearest neighbors (third column) infer the sense of the target word in the corresponding topic domain. For example, the word *drug* is mostly related to "medication" in a broad medical domain; it experiences though a slight shift from this meaning when it resides in a topic about "illegal substances". Furthermore, the highly polysemous word *act* shifts from meaning "statute" to meaning "performance" under the corresponding law and art topics. Similar semantic variations are observed for words *python*, *rock* and *nursery*.

Moreover, in Figure 5.5 we visualize the topic embeddings of seven words before and after projecting the topic-based DSMs to the unified space, using principal component analysis. We additionally depict the Gaussian distribution learned from the topic representations of each word reflecting the uncertainty of their meanings. The center of each distribution is specified by the mean vector and contour surface by the covariance matrix. On the left, we depict the position of words prior to applying the unsupervised mapping approach where the topic sub-spaces are unaligned. In the unaligned space, words demonstrate similar area coverage regardless of their polysemy. After the mappings, we see on the right that the area under a word's distribution is indicative of its degree of polysemy. Specifically,

we observe that the variance of the learned representations becomes larger for the cases of polysemous words such as “python”, “java”, “adobe” in order to assign some probability to their diverse meanings. Monosemous words such as “snake”, “microsoft” and “malay” have smaller variances. Comparing the two different illustrations, we can see that their semantic range changes according to their polysemic nature. In specific, we observe similar variances in their unaligned space gaussians versus variances that agree with a word’s degree of polysemy in the aligned space. Furthermore, we observe that the semantic relationships between words are much better captured by their corresponding positions in the aligned space. In the same way, in Figure 5.6 we visualize the topic embeddings

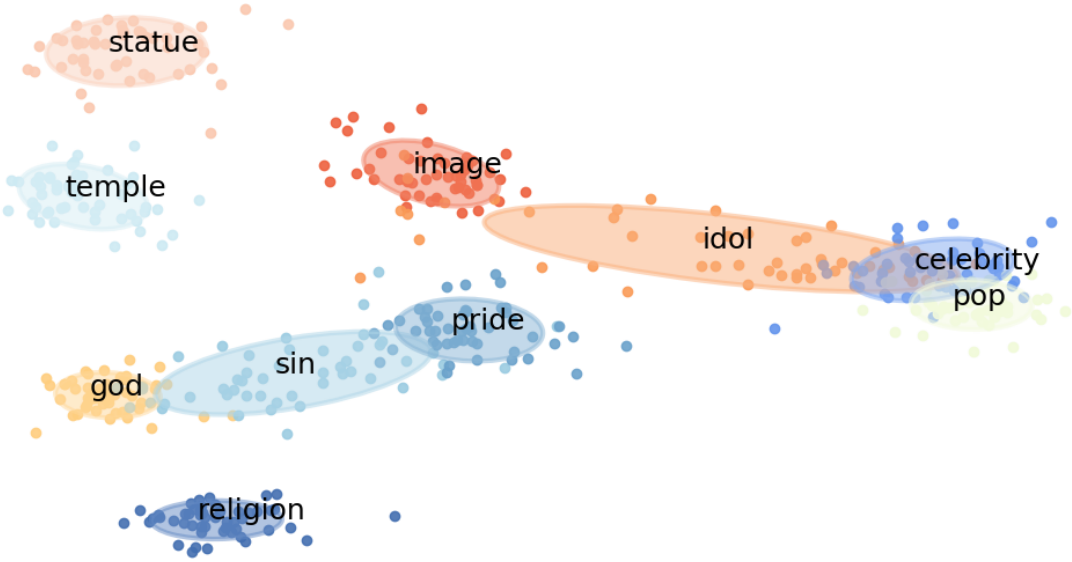


Figure 5.6: A 2-dimensional projection—using tSNE algorithm—of the latent semantic space encoded in our unified vector space model, depicting the topic word representations of 10 words after mapping the TDSMs to the global semantic space.

of a subset which includes ten words after projecting the topic-based DSMs to the unified space, using tSNE algorithm. In the mapped space, we observe again that the area under a word’s distribution is indicative of its degree of polysemy. Moreover, it is clear that the usage of more than one gaussian components in our smoothing approach can improve the meanings capture as for example, “image” and “pride” have some points that tend interfere with other words’ points correctly and cannot be included into a single gaussian component. In Figure 5.7 we can see the visualization of 7 words using

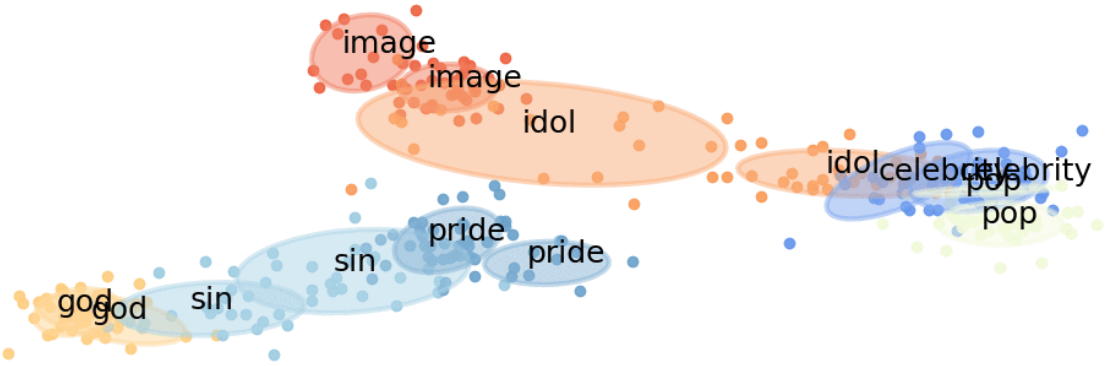


Figure 5.7: A 2-dimensional projection of the latent semantic space encoded in our unified vector space model, depicting the topic word representations of 7 words after mapping the TDSMs to the global semantic space using 2 gaussian components for each word.

2 gaussian components. It is clear, that for polysemous words such as “idol” and “pride” the use of multiple gaussians better discriminates their different meanings. Similarly, monosemous words such as “pop”, “celebrity” and “god” have almost identical gaussians which agrees with their monosemous nature.

5.8 Experimental Summary & Discussion

Overall, our proposed method for unsupervised multiple embeddings creation achieved state-of-the-art results for the major benchmark datasets. Although topics and different word senses are not perfectly aligned, our intuitive smoothing approach improved our results. An adaptive gaussian mixture schema determines the number of gaussian components per word could be more effective as the number of senses differ for different words.

Experiments on downstream NLP tasks showed that a simple exploitation of our topical embeddings improves performance over single-representation models. Additionally, our in depth qualitative analysis demonstrates our model interpretability and validates the differences between aligned and unaligned spaces. Finally, it clearly outlines future directions such as the use of adaptive number of gaussians.

Chapter 6

Conclusions

We draw some conclusions from this work which can be divided into two main categories corresponding to Chapters 4 and 5 respectively. Hence, we will list the most profound conclusions we drew from the work presented in this thesis in respect to the Chapters referred above.

6.1 Cognition & Natural Language Representations

6.1.1 Final Remarks

We proposed a simple neural activation semantic model extending the work of [7]. The performance of the neural model was investigated for the tasks of word semantic similarity, taxonomy creation, sensory modality classification and concept entailment.

The results of our model revealed the different flavor of neural activations compared to conventional embeddings. First, we observed that neural activations alone perform better than stat-of-the-art embeddings in similarity estimation concerning highly similar and dissimilar words. Next, we tested our similarity approach in taxonomy creation and sensory modality classification, achieving performance improvements or similar performance with word embeddings. Finally, their usage as feature representations along with word embeddings in entailment task validated that they provide additive information in the semantic space.

Although the collection of neuroimaging data has many limitations such as variation across participants, high signal-to-noise ratio and the need of expensive equipment for data capture, it provides an alternative view of how lexical and sensory information in localized in the human brain. Despite the very small dataset used in our experiments, results are encouraging about the value of neural activations patterns for computational tasks.

6.1.2 Future Work

As a next step, we will investigate learning how abstract concepts can be depicted in brain and if a neural predictor can be built for abstract concepts. Then, the performance of neural activations could be examined for a whole dictionary in a wider variety of tasks.

Another interesting direction is to determine the differences between textual representations and neural activations in a more specific way and evaluate the semantic or syntactic value of brain representations. The way that word embeddings and neural activations interact in different tasks and respective performance differences may indicate the laters' impact. Furthermore, we would like to explore alternative methods for combining neural activations and word embeddings such as employing more efficient and sophisticasated fusion schemas.

Another parameter that could be investigated is neural activations' localization properties. Utilizing their spatial patterns explicitly might further improve their efficiency and lead us to discover properties of different brain regions with regards to word semantics.

6.2 Cross Topic Natural Language Representations

6.2.1 Final Remarks

We present an unsupervised approach of mapping multiple topic-based DSMs to a unified vector space in order to capture different contextual semantics of words. We assume that words having consistent similarity distributions regardless of the domain they exist in could be considered informative semantic anchors that determine the mappings between semantic spaces. The projected word embeddings yield state-of-the-art results on contextual similarity compared to previously proposed unsupervised approaches for multiple word embeddings creation, while they also outperform single vector representations in downstream NLP tasks. In addition, we provide insightful visualizations and examples that demonstrate the capability of our model to capture variations in topic semantics of words.

6.2.2 Future Work

As future work, one can hypothesize that the area a word covers in the mapped space reveals its semantic range. In this direction, a refinement of the semantic anchor selection approach could be explored in an iterative way assuming that the variance of a word's Gaussian distribution denotes its degree of polysemy [151].

Then, the anchor points can be refined by estimating their polysemy in the global space. Semantic range (degree of polysemy) estimation after their projection in the global space can be determined by calculating the determinant of the covariance matrix after fitting a Gaussian distribution on each words' topic vectors [152]. A new subset of anchor words can then be chosen constituted by the words with the highest variance. Then we can use the new subset of anchor words to learn the projection of the topic vectors to the global space. The aforementioned procedure can be performed iteratively and halt when only a small subset—defined by a threshold parameter—of the anchor words changes.

Moreover, we would like to explore a more sophisticated smoothing technique where the number of Gaussian components is adapted for each word. Given that Gaussian mixture embeddings could capture the uncertainty of a word's representation in the semantic space one could also investigate different metrics for measuring the semantic relationship between word pairs that go beyond their point-wise comparison.

Specifically, clustering operates as an implicit way of segmenting the space of topic embeddings for each word, in order to capture more useful hyper-topics (i.e. union of topics), which better represent their different meanings. A Gaussian Mixture Model is used to iteratively cluster via Expectation-Maximazation, topic embeddings into N Gaussian distributions and reduce the number of embeddings to N by representing each Gaussian component with its corresponding mean vector. Alternatively, one can use k-means algorithm and use the respective centroid of each cluster. However, such approaches require a predetermined number of clusters which is not a natural constraint for different word meanings. In this direction, one can employ hierarchical clustering [153] or density-based estimation [154]. A more intuitive method would be either to select the number of Gaussian components using BIC criterion or start by using 2 components and iteratively split the component with the highest variance until a specific threshold.

Finally, it may be helpful to investigate non-linear mappings between semantic spaces using deep neural network architectures.

Bibliography

[1] 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*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2867–2878. [Online]. Available: https://www.aclweb.org/anthology/C18-1243

[2] E. Briakou, N. Athanasiou, and A. Potamianos, "Cross-topic distributional semantic representations via unsupervised mappings," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 1052–1061. [Online]. Available: https://www.aclweb.org/anthology/N19-1110

[3] A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukacs, M. Ganea, P. Young *et al.*, "Smart reply: Automated response suggestion for email," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 955–964.

[4] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 1995, pp. 181–184.

[5] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[6] S. Marmaridou, *Pragmatic Meaning and Cognition*. John Benjamins Publishing, 2000.

[7] 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, no. 5880, pp. 1191–1195, 2008. [Online]. Available: http://science.sciencemag.org/content/320/5880/1191

[8] A. B. Jelodar, M. Alizadeh, and S. Khadivi, "Wordnet based features for predicting brain activity associated with meanings of nouns," in *Proceedings of the NAACL HLT 2010 First Workshop on Computational Neurolinguistics*, ser. CN '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 18–26. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866686.1866689

[9] A. Huth, W. A. de Heer, T. L. Griffiths, F. Theunissen, and J. Gallant, "Natural speech reveals the semantic maps that tile human cerebral cortex," vol. 532, pp. 453–458, 04 2016.

[10] A. Huth, S. Nishimoto, A. Vu, and J. Gallant, "A continuous semantic space describes the representation of thousands of object and action categories across the human brain," *Neuron*, vol. 76, no. 6, pp. 1210 – 1224, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0896627312009348

[11] U. Noppeney, W. D. Penny, C. J. Price, G. Flandin, and K. J. Friston, "Identification of degenerate neuronal systems based on intersubject variability," *Neuroimage*, vol. 30, no. 3, pp. 885–890, 2006.

[12] K. Kay, T. Naselaris, R. J Prenger, and J. Gallant, "Identifying natural images from human brain activity," vol. 452, pp. 352–5, 04 2008.

[13] B. Thirion, E. Duchesnay, E. Hubbard, J. Dubois, J.-B. Poline, D. Lebihan, and S. Dehaene, "Inverse retinotopy: Inferring the visual content of images from brain activation patterns," *NeuroImage*, vol. 33, no. 4, pp. 1104 – 1116, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811906007373

[14] S. Nishimoto, A. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. Gallant, "Reconstructing visual experiences from brain activity evoked by natural movies," *Current Biology*, vol. 21, no. 19, pp. 1641 – 1646, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960982211009377

[15] T. Naselaris, R. J Prenger, K. Kay, M. Oliver, and J. Gallant, "Bayesian reconstruction of natural images from human brain activity," vol. 63, pp. 902–15, 09 2009.

[16] Y. Miyawaki, H. Uchida, O. Yamashita, M. aki Sato, Y. Morito, H. C. Tanabe, N. Sadato, and Y. Kamitani, "Visual image reconstruction from human brain activity using a combination of multiscale local image decoders," *Neuron*, vol. 60, no. 5, pp. 915 – 929, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0896627308009586

[17] R. R. Benson, D. Whalen, M. Richardson, B. Swainson, V. P. Clark, S. Lai, and A. M. Liberman, "Parametrically dissociating speech and nonspeech perception in the brain using fmri," *Brain and Language*, vol. 78, no. 3, pp. 364 – 396, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0093934X01924848

[18] E. Formisano, F. De Martino, M. Bonte, and R. Goebel, ""who" is saying "what"? brain-based decoding of human voice and speech," *Science*, vol. 322, no. 5903, pp. 970–973, 2008. [Online]. Available: http://science.sciencemag.org/content/322/5903/970

[19] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[20] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[21] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd," *Behavior Research Methods*, vol. 44, no. 3, pp. 890–907, Sep 2012. [Online]. Available: https://doi.org/10.3758/s13428-011-0183-8

[22] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 238–247. [Online]. Available: http://www.aclweb.org/anthology/P14-1023

[23] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 932–938, 01 2000.

[24] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 641–648.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[26] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 384–394. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858721

[27] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 160–167. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390177

[28] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1410–1418. [Online]. Available: http://papers.nips.cc/paper/3650-zero-shot-learning-with-semantic-output-codes.pdf

[29] G. Sudre, D. Pomerleau, M. Palatucci, L. Wehbe, A. Fyshe, R. Salmelin, and T. Mitchell, "Tracking neural coding of perceptual and semantic features of concrete nouns," *NeuroImage*, vol. 62, no. 1, pp. 451 – 463, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811912004442

[30] B. Murphy, P. Talukdar, and T. Mitchell, "Selecting corpus-semantic models for neurolinguistic decoding," in *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, ser. SemEval '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 114–123. [Online]. Available: http://dl.acm.org/citation.cfm?id=2387636.2387658

[31] F. Pulvermüller, "Brain reflections of words and their meaning," *Trends in Cognitive Sciences*, vol. 5, no. 12, pp. 517 – 524, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661300018039

[32] W. Jing, C. V. L., and J. M. Adam, "Predicting the brain activation pattern associated with the propositional content of a sentence: Modeling neural representations of events and states," *Human Brain Mapping*, vol. 38, no. 10, pp. 4865–4881, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23692

[33] 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, no. 9, pp. 4379–4395, 2017. [Online]. Available: http://dx.doi.org/10.1093/cercor/bhw240

[34] A. Fyshe, P. P. Talukdar, B. Murphy, and T. M. Mitchell, "Interpretable semantic vectors from a joint model of brain- and text- based meaning," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 489–499. [Online]. Available: http://www.aclweb.org/anthology/P14-1046

[35] Y.-P. Ruan, Z.-H. Ling, and Y. Hu, "Exploring semantic representation in brain activity using word embeddings," in *Proceedings of the 2016 Conference on Empirical Methods in Natural*

Language Processing. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 669–679. [Online]. Available: https://aclweb.org/anthology/D16-1064

[36] M. Baroni, S. Evert, and A. Lenci, "Bridging the gap between semantic theory and computational simulations," in *Proc. of ESSLLI Distributional Semantic Workshop*, 2008.

[37] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 849–856.

[38] M. Baroni and A. Lenci, "Distributional memory: A general framework for corpus-based semantics," *Computational Linguistics*, vol. 36, no. 4, pp. 673–721, 2010.

[39] E. Bruni, N. Tram, M. Baroni *et al.*, "Multimodal distributional semantics," *The Journal of Artificial Intelligence Research*, vol. 49, pp. 1–47, 2014.

[40] S. S. Tekiroglu, G. Özbal, and C. Strapparava, "Sensicon: An automatically constructed sensorial lexicon," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, October 2014, pp. 1511–1521. [Online]. Available: http://www.aclweb.org/anthology/D14-1160

[41] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 632–642. [Online]. Available: http://aclweb.org/anthology/D15-1075

[42] T. Heed, "Touch perception: How we know where we are touched," *Current Biology*, vol. 20, no. 14, pp. R604–R606, 2010.

[43] E. N. Marieb and K. Hoehn, *Human anatomy & physiology*. Pearson Education, 2007.

[44] M. Kobayashi, "Functional organization of the human gustatory cortex," *Journal of Oral Biosciences*, vol. 48, no. 4, pp. 244–260, 2006.

[45] L. G. Ungerleider, "Two cortical visual systems," *Analysis of visual behavior*, pp. 549–586, 1982.

[46] J. O. Pickles, *An introduction to the physiology of hearing*. Academic press London, 1988, vol. 2.

[47] L. Buck and R. Axel, "A novel multigene family may encode odorant receptors: a molecular basis for odor recognition," *Cell*, vol. 65, no. 1, pp. 175–187, 1991.

[48] J. Reisinger and R. Mooney, "Mixture Model with Sharing for Lexical Semantics," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010, pp. 1173–1182.

[49] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012, pp. 873–882.

[50] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1059–1069.

[51] J. Li and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?" in *Proceedigns of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1722–1732.

[52] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T.-Y. Liu, "A probabilistic model for learning multi-prototype word embeddings," in *Proceedings International Conference on Computational Linguistics (COLING)*, 2014, pp. 151–160.

[53] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," in *Proc. AAAI Conference on Artificial Intelligence*, 2015, pp. 2418–2424.

[54] P. Liu, X. Qiu, and X. Huang, "Learning context-sensitive word embeddings with neural tensor skip-gram model," in *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 1284–1290.

[55] D. Q. Nguyen, D. Q. Nguyen, A. Modi, S. Thater, and M. Pinkal, "A mixture model for learning multi-sense word embeddings," in *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics*, 2017, pp. 121–127.

[56] X. Chen, Z. Liu, and M. Sun, "A unified model for word sense representation and disambiguation," in *Proceedings Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1025–1035.

[57] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Sensembed: Learning sense embeddings for word and relational similarity," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015, pp. 95–105.

[58] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[59] V. Prokhorov, M. T. Pilehvar, D. Kartsaklis, P. Liò, and N. Collier, "Learning rare word representations using semantic bridging," *CoRR*, vol. abs/1707.07554, 2017.

[60] C. Xing, D. Wang, C. Liu, and Y. Lin, "Normalized word embedding and orthogonal transform for bilingual word translation," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, 2015, pp. 1006–1011.

[61] M. Artetxe, G. Labaka, and E. Agirre, "Learning principled bilingual mappings of word embeddings while preserving monolingual invariance," in *Proc. Conference on Empirical Methods in Natural Language Processing*, vol. 1, 2016, pp. 2289–2294.

[62] W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change," in *Proc. 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2016, pp. 1489–1501.

[63] P. H. Schönemann, *A generalized solution of the orthogonal procrustes problem*, 1966.

[64] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[66] H. Amiri, P. Resnik, J. Boyd-Graber, and H. D. III, "Learning text pair similarity with context-sensitive autoencoders," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 1, 2016, pp. 1882–1892.

[67] G.-H. Lee and Y.-N. Chen, "Muse: Modularizing unsupervised sense embeddings," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMLP)*, 2017, pp. 327–337.

[68] F. Guo, M. Iyyer, and J. Boyd-Graber, "Inducing and embedding senses with scaled gumbel softmax," *arXiv preprint arXiv:1804.08077*, 2018.

[69] R. Quiza and J. Davim, *Computational Methods and Optimization*, 01 2011, pp. 177–208.

[70] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6645–6649.

[71] D. M. Blei, "Introduction to Probabilistic Topic Modeling," *Communications of the ACM*, pp. 77–84, 2012.

[72] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting Similarities among Languages for Machine Translation," *arXiv:1309.4168*, 2013.

[73] 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," *Nature communications*, vol. 9, no. 1, p. 963, 2018.

[74] 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: Association for Computational Linguistics, October 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[75] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[76] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[77] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 19–27.

[78] P. Pinel, F. Fauchereau, A. Moreno, A. Barbot, M. Lathrop, D. Zelenika, D. Le Bihan, J.-B. Poline, T. Bourgeron, and S. Dehaene, "Genetic variants of foxp2 and kiaa0319/ttrap/them2 locus are associated with altered brain activation in distinct language-related regions," *Journal of Neuroscience*, vol. 32, no. 3, pp. 817–825, 2012.

[79] A. S. Desroches, N. E. Cone, D. J. Bolger, T. Bitan, D. D. Burman, and J. R. Booth, "Children with reading difficulties show differences in brain regions associated with orthographic processing during spoken language processing," *Brain research*, vol. 1356, pp. 73–84, 2010.

[80] R. C. Berwick, A. D. Friederici, N. Chomsky, and J. J. Bolhuis, "Evolution, brain, and the nature of language," *Trends in cognitive sciences*, vol. 17, no. 2, pp. 89–98, 2013.

[81] R. Jackendoff and R. S. Jackendoff, *Foundations of language: Brain, meaning, grammar, evolution*. Oxford University Press, USA, 2002.

[82] M. Stamenov and V. Gallese, *Mirror neurons and the evolution of brain and language*. John Benjamins Publishing, 2002, vol. 42.

[83] T. W. Deacon, *The symbolic species: The co-evolution of language and the brain*. WW Norton & Company, 1998.

[84] M. Visser, E. Jefferies, and M. Lambon Ralph, "Semantic processing in the anterior temporal lobes: a meta-analysis of the functional neuroimaging literature," *Journal of cognitive neuroscience*, vol. 22, no. 6, pp. 1083–1094, 2010.

[85] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[86] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *Proc. 53rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1556–1566, 2015.

[87] A. Sharaf, S. Feng, K. Nguyen, K. Brantley, and H. Daumé III, "The umd neural machine translation systems at wmt17 bandit learning task," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 667–673.

[88] S. Rothe and H. Schütze, "Autoextend: Extending word embeddings to embeddings for synsets and lexemes," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015, pp. 1793–1803.

[89] M. T. Pilehvar and N. Collier, "De-conflated semantic representations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 1680–1690.

[90] F. Christopoulou, E. Briakou, E. Iosif, and A. Potamianos, "Mixture of topic-based distributional semantic and affective models," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, 2018, pp. 203–210.

[91] W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2016.

[92] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[93] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2011.

[94] G. J. McLachlan and K. E. Basford, "Mixture models," 1988.

[95] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B*, pp. 1–38, 1977.

[96] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[97] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[98] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[99] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[100] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[101] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[102] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[103] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[104] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[105] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[106] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[107] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[108] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[109] S. Dhuria, "Natural language processing: An approach to parsing and semantic analysis," *International Journal of New Innovations in Engineering and Technology*, 2015.

[110] J. R. Firth, "A synopsis of linguistic theory," *Studies in Linguistic Analysis*, pp. 1–32, 1957.

[111] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," in *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*, ser. ACL '89. Stroudsburg, PA, USA: Association for Computational Linguistics, 1989, pp. 76–83. [Online]. Available: https://doi.org/10.3115/981623.981633

[112] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological Review*, pp. 211–240, 1997.

[113] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 151–161. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145450

[114] X. Wang, Y. Liu, S. Chengjie, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1343–1353.

[115] X. Chen, X. Qiu, J. Jiang, and X. Huang, "Gaussian mixture embeddings for multiple word prototypes," *arXiv preprint arXiv:1511.06246, 2015*, 2015.

[116] J. Guo, W. Che, H. Wang, and T. Liu, "Learning sense-specific word embeddings by exploiting bilingual resources," in *Proc. International Conference on Computational Linguistics (COLING)*, 2014, pp. 497–507.

[117] X. Zheng, J. Feng, Y. Chen, H. Peng, and W. Zhang, "Learning context-specific word/character embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 3393–3399.

[118] Z. Wu and C. L. Giles, "Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015, pp. 2188–2194.

[119] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International Journal of Lexicography*, pp. 235–244, 1990.

[120] C. Reed, "Latent dirichlet allocation: Towards a deeper understanding," 2012.

[121] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5228–5235, 2004.

[122] Y. W. Teh, D. Newman, and M. Welling, "A collapsed variational bayesian inference algorithm for latent dirichlet allocation," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, 2006, pp. 1353–1360.

[123] M. D. Hoffman, D. M. Blei, and F. Bach, "Online learning for latent dirichlet allocation," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, vol. 1, 2010, pp. 856–864.

[124] G. Dinu and M. Baroni, "Improving zero-shot learning by mitigating the hubness problem," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2014.

[125] L. Tan, H. Zhang, C. L. A. Clarke, and M. D. Smucker, "Lexical comparison between wikipedia and twitter corpora by using word embeddings," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015, pp. 657–661.

[126] M. Artetxe, G. Labaka, and E. Agirre, "Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[127] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 426–471.

[128] K.-m. K. Chang, T. Mitchell, and M. A. Just, "Quantitative modeling of the neural representation of objects: how semantic feature norms can account for fmri activation," *NeuroImage*, vol. 56, no. 2, pp. 716–727, 2011.

[129] M. A. Just, V. L. Cherkassky, S. Aryal, and T. M. Mitchell, "A neurosemantic theory of concrete noun representation based on the underlying brain codes," *PLOS ONE*, vol. 5, no. 1, pp. 1–15, 01 2010. [Online]. Available: https://doi.org/10.1371/journal.pone.0008622

[130] G. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[131] L. Wehbe, A. Vaswani, K. Knight, and T. Mitchell, "Aligning context-based statistical models of language with brain activity during reading," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 233–243.

[132] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, vol. 80, 2012.

[133] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1387–1392.

[134] J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini, "Distributed and overlapping representations of faces and objects in ventral temporal cortex," *Science*, vol. 293, no. 5539, pp. 2425–2430, 2001. [Online]. Available: http://science.sciencemag.org/content/293/5539/2425

[135] J. P. Levy and J. A. Bullinaria, "Using enriched semantic representations in predictions of human brain activity," *Connectionist Models of Neurocognition and Emergent Behavior: From Theory to Applications. Singapore: World Scientific*, pp. 292–308, 2012.

[136] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, October 2013, pp. 1631–1642. [Online]. Available: http://www.aclweb.org/anthology/D13-1170

[137] A. Ishai, L. G. Ungerleider, A. Martin, J. L. Schouten, and J. V. Haxby, "Distributed representation of objects in the human ventral visual pathway," *Proceedings of the National Academy of Sciences*, vol. 96, no. 16, pp. 9379–9384, 1999.

[138] G. S. Cree and K. McRae, "Analyzing the factors underlying the structure and computation of the meaning of chipmunk, cherry, chisel, cheese, and cello (and many other such concrete nouns)." *Journal of Experimental Psychology: General*, vol. 132, no. 2, p. 163, 2003.

[139] N. G. Kanwisher, J. Mcdermott, and M. M. Chun, "The fusiform face area: A module in human extrastriate cortex specialized for face perception," vol. 17, pp. 4302–11, 07 1997.

[140] J. L. Lauter, P. Herscovitch, C. Formby, and M. E. Raichle, "Tonotopic organization in human auditory cortex revealed by positron emission tomography," *Hearing research*, vol. 20, no. 3, pp. 199–205, 1985.

[141] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[142] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom, "Reasoning about entailment with neural attention," *arXiv preprint arXiv:1509.06664*, 2015.

[143] E. Iosif, S. Georgiladakis, and A. Potamianos, "Cognitively motivated distributional representations of meaning," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Paris, France: European Language Resources Association (ELRA), may 2016.

[144] P. Hagoort, "Nodes and networks in the neural architecture for language: Broca's region and beyond," *Current opinion in Neurobiology*, vol. 28, pp. 136–141, 2014.

[145] ——, "On broca, brain, and binding: a new framework," *Trends in cognitive sciences*, vol. 9, no. 9, pp. 416–423, 2005.

[146] J. C. Hoeks, L. A. Stowe, and G. Doedens, "Seeing words in context: the interaction of lexical and sentence level information during reading," *Cognitive brain research*, vol. 19, no. 1, pp. 59–73, 2004.

[147] G. R. Kuperberg, "Neural mechanisms of language comprehension: Challenges to syntax," *Brain research*, vol. 1146, pp. 23–49, 2007.

[148] M. Artetxe, G. Labaka, and E. Agirre, "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 789–798.

[149] P. D. Turney, "Domain and function: A dual-space model of semantic relations and compositions," *Journal of Artificial Intelligence Research*, vol. 44, pp. 533–585, 2012.

[150] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, pp. 627–633, 1965.

[151] L. Vilnis and A. McCallum, "Word representations via gaussian embedding." in *International Conference on Learning Representations (ICLR)*, 2015.

[152] C. Sun, H. Yan, X. Qiu, X. Huang, and Z. Chen, "Gaussian word embedding with a wasserstein distance loss," *arXiv preprint arXiv:1808.07016*, 2018.

[153] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 407–416.

[154] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.