Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης

Advancing Visual Word Disambiguation: A Hybrid Approach with Large Language Models, Transformers and Introduction to Novel Hybrid ArPa Model

Diploma Thesis

by

Aristi Papastavrou

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης

Αθήνα, Μάρτιος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
ΔΠΜΣ "Επιστήμη Δεδομένων και Μηχανική Μάθηση"
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Advancing Visual Word Disambiguation: A Hybrid Approach with Large Language Models, Transformers and Introduction to Novel Hybrid ArPa Model

Diploma Thesis

by

Aristi Papastavrou

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Μαρτίου, 2024.

.....	.....	.....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2024

.....

Αριστη Παπασταυρου
Πτυχιούχος Πληροφορικής και Τηλεπικοινωνιών Ε.Κ.Π.Α.

Περίληψη

Στην εποχή της ταχύτατης προόδου στην τεχνητή νοημοσύνη, η σύνθεση οπτικών και κειμενικών δεδομένων αποτελεί ένα συναρπαστικό μέτωπο για εξερεύνηση. "Προάγοντας την Οπτική Διαλεύκανση Λέξεων: Μια Υβριδική Προσέγγιση με Μεγάλα Μοντέλα Γλώσσας, Transformers και Εισαγωγή στο Νέο Υβριδικό Μοντέλο ArPa" εισχωρεί σε αυτή την ενδιαφέρουσα διασταύρωση, με στόχο να αποκρυπτογραφήσει τις πολυπλοκότητες της Οπτικής Διαλεύκανσης Νοήματος Λέξεων (V-WSD). Αυτή η διατριβή προτείνει μια υβριδική προσέγγιση που εκμεταλλεύεται τη δύναμη των μεγάλων μοντέλων γλώσσας και των transformers για να ενισχύσει την ερμηνευτικότητα και την ένταξη των πολυτροπικών πληροφοριών, καταλήγοντας στην εισαγωγή ενός καινοτόμου μοντέλου, του ArPa.

Στο επίκεντρο αυτής της εξερεύνησης είναι η προσπάθεια να αναδιαμορφωθούν οι διαδικασίες μέσω των οποίων οι μηχανές κατανοούν και πλαισιώνουν οπτικές και κειμενικές υποδείξεις ταυτόχρονα. Αξιολογώντας κορυφαία υπολογιστικά μοντέλα και εισάγοντας το μοντέλο ArPa - ένα υβριδικό πλαίσιο που συνδυάζει το αναλυτικό βάθος μεγάλων μοντέλων γλώσσας όπως το Bert με την αντιληπτική διορατικότητα οπτικών μετασχηματιστές (transformers) όπως το Swin Transformer, εμπλουτισμένο περαιτέρω με την ένταξη ενός Δικτύου Γράφων Νευρωνικών Δικτύων - αυτή η έρευνα αποσκοπεί στη θέσπιση νέων προτύπων στην πολυτροπική κατανόηση.

Η διατριβή αναλαμβάνει μια αποστολή εξερεύνησης και ανάλυσης, φιλοδοξώντας να φωτίσει τις πολυεπίπεδες προκλήσεις της Αποσαφήνισης Οπτικών Εννοιών. Μέσω μιας προσεκτικής εξέτασης τεχνικών προετοιμασίας δεδομένων, συμπεριλαμβανομένης της επέκτασης του κειμενικού πλαισίου και της προηγμένης επεξεργασίας εικόνων, καθώς και της εξερεύνησης ποικιλίας αρχιτεκτονικών μοντέλων, προσπαθούμε να βελτιστοποιήσουμε τη συνέργεια μεταξύ των κειμενικών και οπτικών δεδομένων, ενισχύοντας έτσι την απόδοση των μοντέλων σε ένα φάσμα εργασιών Αποσαφήνισης Οπτικών Εννοιών.

Αυτή η εργασία όχι μόνο συνεισφέρει νέες διορατικές προσεγγίσεις και μεθοδολογίες στον τομέα της τεχνητής νοημοσύνης αλλά και καλεί την επιστημονική κοινότητα και τους ενθουσιώδεις της τεχνολογίας να προσβλέπουν σε ένα μέλλον όπου η άρτια ένταξη γλώσσας και οράσεως μεταμορφώνει την αλληλεπίδρασή μας με την τεχνολογία. Προτείνοντας καινοτόμες προσεγγίσεις και αποκαλύπτοντας το μοντέλο ArPa, αυτή η διατριβή ανοίγει νέους δρόμους για έρευνα και εφαρμογή στην πολυτροπική μάθηση, υποσχόμενη να εμπλουτίσει το ψηφακό μας τοπίο με πιο έξυπνα και πιο διαισθητικά συστήματα τεχνητής νοημοσύνης.

Λέξεις-κλειδιά — Αποσαφήνιση Οπτικών Εννοιών, Πολυτροπική Μάθηση, Μεγάλα Μοντέλα Γλώσσας, Νευρωνικά Δίκτυα Γράφων, Μετασχηματιστές (Transformers), Μοντέλο ArPa.

Abstract

In the era of rapid advancements in artificial intelligence, the fusion of visual and textual data presents a compelling frontier for exploration. "Advancing Visual Word Disambiguation: A Hybrid Approach with Large Language Models, Transformers, and Introduction to Novel Hybrid ArPa Model" delves into this intriguing intersection, aiming to unravel the complexities of Visual Word Sense Disambiguation (V-WSD). This thesis proposes a hybrid approach that leverages the prowess of large language models and transformers to enhance the interpretability and integration of multimodal information, culminating in the introduction of a novel model, ArPa.

At the heart of this exploration is the quest to refine the processes through which machines understand and contextualize visual and textual cues in tandem. By evaluating state-of-the-art computational models and introducing the ArPa model—a hybrid framework that marries the analytical depth of large language models like Bert with the perceptual acuity of visual transformers such as Swin Transformer, enriched further by Graph Neural Network (GNN) integration—this research seeks to set new benchmarks in multimodal understanding.

The thesis embarks on a journey of experimentation and analysis, aiming to shed light on the multifaceted challenges of V-WSD. Through a meticulous examination of preprocessing techniques, including linguistic context expansion and advanced image processing, and exploring a variety of model architectures, we endeavor to optimize the synergy between textual and visual data, thereby enhancing model performance across a spectrum of V-WSD tasks.

This work not only contributes novel insights and methodologies to the domain of artificial intelligence but also beckons the scientific community and technology enthusiasts alike towards a future where the seamless integration of language and vision transforms our interaction with technology. By proposing innovative approaches and unveiling the ArPa model, this thesis opens new avenues for research and application in multimodal learning, promising to enrich our digital landscape with more intelligent, nuanced, and empathetic artificial intelligence systems.

Keywords — Visual Word Sense Disambiguation, Multimodal Learning, Large Language Models, Graph Neural Networks, Transformers, ArPa Model.

Ευχαριστίες

Αυτό το έργο δεν θα ήταν δυνατό χωρίς την υποστήριξη πολλών ανθρώπων. Ευχαριστώ πολύ τον επιβλέποντα μου, κ. Στάμου Γεώργιο, για την πολύτιμη καθοδήγηση του στην εκπόνηση αυτής της εργασίας. Ευχαριστώ επίσης την Μαρία Λυμπεραίου για την υποστήριξη της καθ' όλη τη διάρκεια εξερεύνησης των καινούριων αυτών αντικειμένων.

Πολύ σημαντική για εμένα ήταν ακόμα η συναισθηματική στήριξη των κοντινών μου προσώπων, των οποίων η ακλόνητη πίστη στις δυνατότητές μου έχει αποτελέσει σταθερή πηγή δύναμης και ενθάρρυνσης να κυνηγήσω τα όνειρα μου. Αφιερώνω λοιπόν την έρευνα μου, στην ομάδα μου ... ευχαριστώ που υπάρχετε.

Παπασταύρου Αρίστη, Φεβρουάριος 2024

Contents

List of Figures

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

1.1 Θεωρητικό υπόβαθρο

Αυτό το κεφάλαιο έχει σχεδιαστεί για να παρέχει στον αναγνώστη τις θεμελιώδεις γνώσεις που είναι απαραίτητες για να κατανοήσει το υπόλοιπο της ερευνητικής εργασίας. Αυτό το κεφάλαιο θα θέσει τις βάσεις παρουσιάζοντας τις αρχές της αποσαφήνισης οπτικών λέξεων, των γραφηματικών νευρωνικών δικτύων και των βασικών εννοιών της τεχνητής νοημοσύνης. Αυτή η ενότητα αποτελεί ουσιαστικά τον κορμό της διατριβής, συνδέοντας αυτές τις διάφορες έννοιες μαζί στο πλαίσιο της πρόβλεψης της αγοράς μετοχών. Ο τομέας της επεξεργασίας φυσικής γλώσσας (NLP) έχει γνωρίσει σημαντικές προόδους τα τελευταία χρόνια, οδηγούμενος από την ανάπτυξη περίπλοκων υπολογιστικών μοντέλων ικανών να κατανοούν και να παράγουν ανθρώπινη γλώσσα με πρωτοφανή ακρίβεια. Ανάμεσα σε αυτές τις εξελίξεις, η πρόκληση της αποσαφήνισης οπτικών λέξεων ξεχωρίζει ως ένα κρίσιμο σύνορο. Αυτό περιλαμβάνει τη διαδικασία της ακριβούς ταυτοποίησης της σημασίας των λέξεων που έχουν πολλαπλές ερμηνείες με βάση το οπτικό τους πλαίσιο.
Καθώς ο ψηφιακός κόσμος γίνεται όλο και πιο οπτικός, με μια έκρηξη εικόνων και βίντεο συνοδευόμενη από κειμενικά δεδομένα, η ικανότητα να αποσαφηνίζουμε αποτελεσματικά λέξεις μέσα σε αυτά τα οπτικά πλαίσια γίνεται απαραίτητη. Αυτή η διατριβή παρουσιάζει μια ολοκληρωμένη εξερεύνηση της αποσαφήνισης οπτικών λέξεων, προτείνοντας μια νέα υβριδική προσέγγιση που εκμεταλλεύεται τα πλεονεκτήματα των Μεγάλων Μοντέλων Γλώσσας (LLMs), των Transformers και παρουσιάζει το πρωτοποριακό Υβριδικό Μοντέλο ArPa.

Η σημασία της αντιμετώπισης της αποσαφήνισης οπτικών λέξεων δεν μπορεί να υπερτονιστεί. Σε μια εποχή όπου το οπτικό περιεχόμενο κυριαρχεί στην ψηφιακή επικοινωνία, από τις πλατφόρμες κοινωνικών δικτύων έως τους εκπαιδευτικούς πόρους, η ενσωμάτωση κειμένου και εικονογραφήσεων έχει γίνει πανταχού παρούσα. Ωστόσο, η εγγενής ασάφεια στη γλώσσα αποτελεί σημαντική πρόκληση, καθώς πολλές λέξεις φέρουν πολλαπλές σημασίες ανάλογα με το πλαίσιο τους. Οι παραδοσιακές τεχνικές αποσαφήνισης με βάση το κείμενο έχουν κάνει σημαντική πρόοδο, ωστόσο συχνά αποτυγχάνουν όταν εφαρμόζονται σε περίπλοκα οπτικο-κειμενικά δεδομένα. Αυτός ο περιορισμός τονίζει την ανάγκη για πιο προηγμένα μοντέλα ικανά να ερμηνεύουν την λεπτή αλληλεπίδραση μεταξύ οπτικών στοιχείων και κειμενικών πληροφοριών.

Εισέρχοντας στον κόσμο των Μεγάλων Μοντέλων Γλώσσας και των Transformers, τεχνολογίες στην πρωτοπορία του NLP. Αυτά τα μοντέλα έχουν επαναστατήσει την προσέγγισή μας στην κατανόηση της γλώσσας, προσφέροντας βαθιές διεισδύσεις στις γλωσσικές δομές και το πλαίσιο. Τα LLMs, με τις τεράστιες βάσεις γνώσης και την περίπλοκη κατανόηση των νυάνσεων της γλώσσας, παρέχουν μια σταθερή βάση για τα καθήκοντα αποσαφήνισης. Παράλληλα, οι Transformers, γνωστοί για την ικανότητά τους να χειρίζονται δεδομένα ακολουθιών και να αναλαμβάνουν μακρινές εξαρτήσεις, έχουν γίνει αναντικατάστατοι στην επεξεργασία των περίπλοκων σχέσεων μεταξύ λέξεων και των οπτικών τους πλαισίων. Ωστόσο, παρά τα πλεονεκτήματά τους, αυτά τα μοντέλα ακόμη αντιμετωπίζουν προκλήσεις στην πλήρη κατανόηση των λεπτομερειών της αποσαφήνισης οπτικών λέξεων, απαιτώντας την ανάπτυξη πιο εξειδικευμένων προσεγγίσεων.

Αυτή η διατριβή παρουσιάζει το Υβριδικό Μοντέλο ArPa, μια πρωτοποριακή προσέγγιση που σχεδιάστηκε για να κλείσει το χάσμα στην αποσαφήνιση οπτικών λέξεων. Το μοντέλο συνδυάζει τις δυνατότητες κατανόησης πλαισίου των LLMs με τις δυνατότητες δομικής επεξεργασίας των Transformers και των γραφηματικών νευρωνικών δικτύων (GNNs). Στην καρδιά του, το Υβριδικό Μοντέλο ArPa ενσωματώνει αρχιτεκτονικές καινοτομίες και στρατηγικές επεξεργασίας που ενισχύουν την ερμηνεία των οπτικο-κειμενικών δεδομένων. Μέσω της ένταξης αυτών των στοιχείων, το μοντέλο επιτυγχάνει υψηλότερο επίπεδο ακρίβειας στην αποσαφήνιση λέξεων μέσα σε διάφορα οπτικά περιβάλλοντα.

Η ανάπτυξη του Υβριδικού Μοντέλου ArPa ενθαρρύνθηκε από την αναγνώριση ότι η πολυπλοκότητα της αποσαφήνισης οπτικών λέξεων απαιτεί μια πολυεπίπεδη προσέγγιση. Αυτό το μοντέλο διακρίνεται από την υβριδική του φύση, συνδυάζοντας στοιχεία από διαφορετικά τεχνολογικά παραδείγματα για να αντιμετωπίσει τις πολυάριθμες προκλήσεις που παρουσιάζονται από τα οπτικο-κειμενικά δεδομένα. Η ένταξη μιας ειδικά σχεδιασμένης μονάδας επεξεργασίας οπτικών στοιχείων εντός του μοντέλου, επιτρέπει μια πιο λεπτομερή ανάλυση εικόνων και βίντεο, διευκολύνοντας μια βαθύτερη κατανόηση του πλαισίου που περιβάλλει τις αμφίσημες λέξεις.

Αυτή η διατριβή δομείται έτσι ώστε να παρέχει μια ολοκληρωμένη κατανόηση της αποσαφήνισης οπτικών λέξεων, ξεκινώντας με μια εκτενή ανασκόπηση της τρέχουσας κατάστασης της έρευνας στον τομέα. Στη συνέχεια, εμβαθύνει στις θεωρητικές βάσεις των Μεγάλων Μοντέλων Γλώσσας και των Transformers, διευκρινίζοντας τους ρόλους τους στην προώθηση του NLP. Τα κεντρικά κεφάλαια είναι αφιερωμένα στην λεπτομερή εξιστόρηση του Υβριδικού Μοντέλου ArPa, περιλαμβάνοντας την έννοια βάση του, τον αρχιτεκτονικό σχεδιασμό και την υλοποίησή του. Μέσω εμπειρικών μελετών και συγκριτικής ανάλυσης, η διατριβή αποδεικνύει την αποτελεσ-

ματικότητα του μοντέλου στην αποσαφήνιση λέξεων σε μια ποικιλία οπτικών πλαισίων, επισημαίνοντας την πιθανότητα του να μεταμορφώσει το τοπίο της επεξεργασίας οπτικής γλώσσας. Η δομή αυτής της διατριβής είναι η εξής:

- Αρχικά, θα παρέχουμε όλο το απαραίτητο υπόβαθρο στους βασικούς αλγόριθμους και τις έννοιες Μηχανικής Μάθησης της αποσαφήνισης οπτικών λέξεων, καθώς και των transformers και των Μεγάλων Μοντέλων Γλώσσας, ώστε να μπορέσουμε να εξηγήσουμε και να δικαιολογήσουμε την ιδέα της χρήσης των Γραφηματικών Νευρωνικών Δικτύων σε μια αρχιτεκτονική διαδρομή. Αφού το πράξουμε αυτό, θα παρέχουμε μια λεπτομερή περιγραφή των παραλλαγών GNN που σχετίζονται με αυτή την εργασία.
- Θα εξερευνήσουμε διαφορετικά μοντέλα και στρατηγικές που χρησιμοποιούνται για την αντιμετώπιση εργασιών V-wsd
- Τελικά, θα προτείνουμε την αρχιτεκτονική μας ArPa για την αποσαφήνιση οπτικών λέξεων με βάση μια διαδρομή από LLMs, transformers και GNNs. Παράλληλα, θα τονίσουμε την απόδοση του μοντέλου μας χρησιμοποιώντας διάφορες υπερπαραμέτρους και θα συγκρίνουμε αυτά τα αποτελέσματα με πιο συμβατικές μεθόδους που περιγράφονται στις προηγούμενες ενότητες και θα καταλήξουμε σε συμπεράσματα βασισμένα στην εκφραστικότητα και την αξιολόγηση των ποσοτικών και ποιοτικών αποτελεσμάτων.

1.1.1 Νευρωνικά Δίκτυα Γράφων

Δεδομένου ότι η δομή του γράφου αναδύεται φυσικά παντού γύρω μας, εφευρέθηκαν νευρωνικά δίκτυα που λειτουργούν απευθείας σε δεδομένα αυτού του τύπου. Τα γραφήματα είναι μη ευκλείδεια δεδομένα και επομένως τα GNN μπορούν να ομαδοποιηθούν στην ευρύτερη κατηγορία της Γεωμετρικής Μάθησης [3]. Τα Νευρωνικά Δίκτυα Γράφων (GNN) είναι γνωστά για την εκφραστική τους ισχύ και πρόσφατα κερδίζουν δημοτικότητα λόγω των αυξανόμενων δυνατοτήτων τους σε διάφορες εφαρμογές όπως τα συστήματα συστάσεων και το μοριακό δακτυλικό αποτύπωμα [26].

Τα GNN δημιουργήθηκαν γιατι οι περισσότεροι συμβατικοί αλγόριθμοι Machine ή Deep Learning είναι ειδικά κατασκευασμένοι για να καλύπτουν συγκεκριμένο τύπο δεδομένων, όπως εικόνες ή κείμενο, όχι όμως γράφους. Οι περισσότερες αναπαραστάσεις δεδομένων μπορούν να γενικευθούν σε γράφους, αλλά το αντίθετο δεν ισχύει. Στη γενική περίπτωση, τα γραφήματα είναι πιο πολύπλοκα, έχοντας έναν μη σταθερό αριθμό μη ταξινομημένων κόμβων μέσα σε γειτονιές μεταβλητού μεγέθους, και επομένως τα υπάρχοντα μοντέλα δεν μπορούν να τα χειριστούν. Επιπλέον, οι περισσότεροι κοινοί αλγόριθμοι υποθέτουν την ανεξαρτησία στιγμιοτύπων. Αυτό δεν ισχύει όταν εκτελούνται εργασίες σε επίπεδο κόμβου όπου ένα γράφημα είναι η είσοδος του νευρωνικού δικτύου και τα στιγμιότυπα είναι οι κόμβοι του. Τέλος, τα κλασικά Συνελικτικα Νευρωνικά Δίκτυα λειτουργούν σε εικόνες ή γενικότερα κανονικά πλέγματα. Η έλλειψη εντοπιότητας με την παραδοσιακή έννοια στα δεδομένα γράφων, το αυθαίρετο μέγεθος και η αμετοβλητότητα τους σε μεταθέσεις καθιστούν δύσκολη την εκτέλεση της κανονικής συνέλιξης.

Τα Νευρωνικά Δίκτυα Γράφων μπορούν να ταξινομηθούν με διάφορους τρόπους: α) ανάλογα με το επίπεδο του γράφου στο οποίο λειτουργούν σε επιπέδου κόμβου, ακμής ή γραφου, β) ανάλογα με την αρχιτεκτονική που ακολουθούν σε συνελικτικά, επαναλαμβανόμεσα, αυτοκωδικοποιητές και χωροχρονικά και γ) ανάλογα με τον τρόπο εκπαίδευσης σε επιβλεπόμενα, μη επιβλεπόμενα και μερικώς επιβλεπόμενα. Παρακάτω θα αναλύσουμε τις τρεις εκδοχές συνελικτικών δικτύων που θα χρησιμοποιηθούν στο πειραματικό μέρος.

Το **Graph Convolutional Network (GCN)** παρουσιάζει την ιδέα της χρήσης μιας προσέγγισης πρώτης τάξης του ChebNet προκειμένου να μετριαστεί η υπερπροσαρμογή. Στην πραγματικότητα, υποθέτει $K = 1$ και $\lambda_{max} = 2$. Στην ίδια κατεύθυνση το μοντέλο επιβάλλει τον περιορισμό $\theta = \theta_0 = -\theta_1$. Μετά την επιβολή αυτών των περιορισμών, η λειτουργία συνέλιξης είναι:

$$x *_G g_\theta = \theta(I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x \tag{1.1.1}$$

Αφού διαπιστώθηκε εμπειρικά ότι ο όρος $I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ προκαλεί αριθμητική αστάθεια, χρησιμοποιήθηκε ένα τέχνασμα *επανακανονικοποίησης*. Ο όρος $D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = \bar{A}$ αντικαταστάθηκε από $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} = \hat{A}$ όπου $\tilde{A} = I_n + \bar{A}$ και $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. Όλα τα παραπάνω μπορούν να περιγραφούν με αυτή τη συμπαγή εξίσωση:

$$H = X *_G g_\theta = f(\hat{A} X \Theta) \tag{1.1.2}$$

όπου το f είναι μια συνάρτηση ενεργοποίησης και επιτρέπονται πολλαπλές είσοδοι και έξοδοι λόγω της χρήσης πινάκων.

Το GCN είναι μια ειδική περίπτωση φασματικής προσέγγισης αφού μπορεί να εκληφθεί και ως χωρική. Στην παρακάτω εξίσωση, μπορούμε να δούμε πώς θα γίνει η συγκέντρωση πληροφοριών εντός της γειτονιάς. Σε αυτήν την περίπτωση ο ίδιος ο κόμβος θεωρείται επίσης ως γείτονάς του εαυτού του, ενός βήματος.

$$h_v = f(\Theta^T(\sum_{u \in N(u) \cup v} \hat{A}_{v,u} x_u)) \quad \forall u \in V \tag{1.1.3}$$

Αυτό το μοντέλο χρησιμοποιείται πολύ συχνά ως μέρος πιο σύνθετων αρχιτεκτονικών στη λογοτεχνία λόγω της απλότητας και της καλής πειραματικής του απόδοσης.

Το **Graph Attention Network (GAT)** [18] υιοθετεί την ιδέα της προσοχής που προτείνεται από το [17] προκειμένου να αποφασίσει ποια μέλη της γειτονιάς ενός κόμβου έχουν πιο σημαντικές πληροφορίες. Στόχος του είναι να μάθει τα σχετικά βάρη μεταξύ γειτονικών κόμβων και επομένως διαφέρει από προηγούμενες προσεγγίσεις όπως το GCN και το GraphSAGE επειδή η έννοια της γειτονιάς δεν είναι προκαθορισμένη ή πανομοιότυπη.

Η συνεκτική λειτουργία ορίζεται ως:

$$h_v^{(k)} = \sigma(\sum_{u \in N(u) \cup v} \alpha_{vu}^{(k)} W^{(k)} h_u^{(k-1)}) \tag{1.1.4}$$

όπου τα βάρη προσοχής για κάθε κόμβο v μπορούν να οριστούν ως:

$$\alpha_{vu}^{(k)} = softmax(LeakyReLU(a^T[W^{(k)} h_v^(k-1) || W^{(k)} h_u^{(k-1)}])) \tag{1.1.5}$$

Η μεταβλητή a αντιπροσωπεύει ένα σύνολο παραμέτρων με δυνατότητα εκμάθησης. Η αναπαράσταση των κρυφών επιπέδων αρχικοποιείται με τα χαρακτηριστικά κάθε κόμβου και η συνάρτηση softmax διασφαλίζει ότι τα βάρη της προσοχής αθροίζονται σε ένα.

Ο παραπάνω μηχανισμός ονομάζεται *self-attention*, αλλά το GAT χρησιμοποιεί επιπλέον *multi-head attention* για να σταθεροποιήσει τη μάθηση και να κάνει το μοντέλο πιο εκφραστικό. Οι ακριβείς εξισώσεις βρίσκονται στο [18].

Το GAT είναι αποτελεσματικό αφού από τα ζεύγη κόμβου-γείτονα μπορούν να υπολογιστούν ταυτόχρονα. Επιπλέον, τα μεγέθη της γειτονιάς του είναι αδιάφορα και μπορεί να εφαρμοστεί εύκολα σε επαγωγικά μαθησιακά προβλήματα.

1.1.2 Transformers - Μετασχηματιστές

Οι μετασχηματιστές (Transformers) αποτελούν μια επαναστατική αρχιτεκτονική στον τομέα της τεχνητής νοημοσύνης και ιδιαίτερα στην επεξεργασία φυσικής γλώσσας (NLP), προσφέροντας τη δυνατότητα για πολύ αποδοτική και βαθιά κατανόηση γλωσσικών δεδομένων. Η καινοτομία τους βασίζεται στην ικανότητά τους να χειρίζονται ταυτόχρονα όλες τις λέξεις (ή τα δεδομένα) σε μια πρόταση ή έγγραφο, επιτρέποντας την παραλληλική επεξεργασία και την αποτελεσματική κατανόηση των σχέσεων και των συνδέσεων μεταξύ των λέξεων, ανεξαρτήτως της απόστασης μεταξύ τους στο κείμενο.

Όσον αφορά τους οπτικούς μετασχηματιστές (Visual Transformers), αυτοί εφαρμόζουν την ίδια βασική αρχή στην ανάλυση και κατανόηση οπτικών δεδομένων, όπως εικόνες. Με την αντιμετώπιση της εικόνας ως μια σειρά από «λέξεις» ή περιοχές, οι οπτικοί μετασχηματιστές είναι σε θέση να εξάγουν σημασιολογική πληροφορία και να κατανοήσουν τις σχέσεις μεταξύ διαφορετικών μερών της εικόνας, επιτρέποντας προηγμένη ανάλυση και ερμηνεία.

Ειδικότερα, ο Swin Transformer αποτελεί μια πρόσφατη εξέλιξη στην κατηγορία των οπτικών μετασχηματιστών, η οποία υιοθετεί μια ιεραρχική δομή για να διαχειρίζεται αποδοτικά τις εικόνες σε διαφορετικά επίπεδα ανάλυσης. Αυτό επιτρέπει στον Swin Transformer να εστιάζει σε λεπτομερείς περιοχές της εικόνας όταν είναι απαραίτητο, ενώ παράλληλα διατηρεί την ικανότητα να αντιλαμβάνεται την ευρύτερη σύνθεση και τη σχετικότητα

των στοιχείων σε αυτή. Αυτή η προσέγγιση καθιστά τον Swin Transformer ιδιαίτερα αποτελεσματικό σε εφαρμογές όπως η αναγνώριση αντικειμένων, η σημασιολογική τμηματοποίηση και η ανάλυση εικόνων, προσφέροντας μια προηγμένη ικανότητα κατανόησης του οπτικού περιεχομένου.

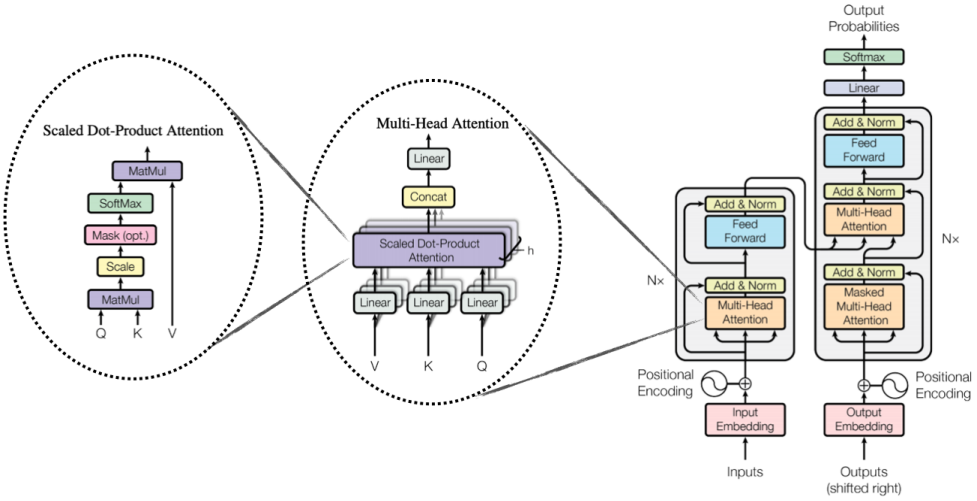

Figure 1.1.1: Αρχιτεκτονική Μετασχηματιστών (Transformers) [17]

1.1.3 Πολυτροπικά Μοντέλα - MultiModal Models

Τα πολυτροπικά μοντέλα αποτελούν μια σημαντική κατηγορία στον τομέα της τεχνητής νοημοσύνης, που ενσωματώνουν και επεξεργάζονται πληροφορία από πολλαπλές πηγές ή τύπους δεδομένων, όπως κείμενο, εικόνα, ήχος και βίντεο. Η πολυτροπικότητα επιτρέπει στα μοντέλα να κατανοούν πιο πλούσια και σύνθετα περιβάλλοντα, προσφέροντας μια πιο ολοκληρωμένη και βαθιά ερμηνεία του κόσμου γύρω μας. Στην ουσία, η πολυτροπικότητα μιμείται τον τρόπο με τον οποίο οι άνθρωποι αντλούν νόημα από τον κόσμο, συνδυάζοντας διαφορετικές αισθήσεις και τύπους πληροφορίας για να καταλήξουν σε πιο πλήρεις κατανοήσεις και αποφάσεις. Η ενσωμάτωση και ανάλυση διαφορετικών μορφών δεδομένων απαιτεί προηγμένες τεχνικές και αλγορίθμους, όπως η σύνθεση νευρωνικών δικτύων που ειδικεύονται σε κάθε τύπο μέσου και η χρήση τεχνικών βαθιάς μάθησης για την εξαγωγή και τη συνένωση χαρακτηριστικών από κάθε τρόπο.

Ένα από τα πιο σημαντικά πολυτροπικά μοντέλα που έχει παρουσιαστεί πρόσφατα είναι το CLIP (Contrastive Language–Image Pre-training) από την OpenAI. Το CLIP συνδυάζει κείμενο και εικόνες σε ένα ενιαίο μοντέλο, εκπαιδεύοντας σε μια τεράστια ποικιλία εικόνων και των περιγραφών τους από το διαδίκτυο. Αυτή η προσέγγιση επιτρέπει στο μοντέλο να κατανοήσει και να ερμηνεύσει τη σχέση μεταξύ κειμένου και εικόνας σε ένα εξαιρετικά ευρύ φάσμα περιστάσεων, παρέχοντας μια πολύ ευέλικτη και γενικευμένη ικανότητα κατανόησης.

Το CLIP επιτυγχάνει αυτό χρησιμοποιώντας τεχνικές contrastive learning, οι οποίες εκπαιδεύουν το μοντέλο να συσχετίζει σωστά τις εικόνες με τις αντίστοιχες περιγραφές τους, ενισχύοντας την ικανότητά του να κατανοεί το περιεχόμενο και τη σημασία που περικλείεται τόσο στο κείμενο όσο και στην εικόνα. Αυτή η ικανότητα καθιστά το CLIP ιδιαίτερα χρήσιμο για μια σειρά από εφαρμογές, όπως η αναζήτηση εικόνων με βάση κειμενικές περιγραφές, η αυτόματη κατηγοριοποίηση εικόνων, και η δημιουργία περιεχομένου. Η ικανότητα του CLIP να καταλαβαίνει τη σχέση μεταξύ διαφορετικών τρόπων πληροφορίας ανοίγει νέους δρόμους για την ανάπτυξη πιο εξελιγμένων και αποδοτικών πολυτροπικών μοντέλων στο μέλλον.

Τα πολυτροπικά μοντέλα βρίσκουν εφαρμογές σε μια πληθώρα τομέων, όπως η αυτόματη ερμηνεία και γεννήτρια παραγωγή περιεχομένου, η αναγνώριση αντικειμένων και σκηνών σε εικόνες, η ανάλυση συναισθημάτων από προφορικό λόγο και κείμενο, καθώς και η διεπαφή μεταξύ ανθρώπου και υπολογιστή. Η ανάπτυξη και η βελτίωση των πολυτροπικών μοντέλων ανοίγει νέους δρόμους για την εξέλιξη της τεχνητής νοημοσύνης, καθώς αποκαλύπτει νέες δυνατότητες για την κατανόηση και την αλληλεπίδραση με τον πολύπλοκο κόσμο μας.

1.1.4 Οπτική Αποσαφήνιση Εννοιών

Η οπτική αποσαφήνιση εννοιών (Visual Word Sense Disambiguation - WSD) αποτελεί έναν προκλητικό τομέα στην επεξεργασία φυσικής γλώσσας και την υπολογιστική όραση, που αφορά την ακριβή ερμηνεία της σημασίας λέξεων με βάση το οπτικό τους πλαίσιο. Η διαδικασία αυτή είναι κρίσιμη σε εφαρμογές όπου οι λέξεις έχουν πολλαπλές σημασίες, και η σημασία τους πρέπει να καθοριστεί από τις συνοδευτικές οπτικές πληροφορίες. Για παράδειγμα, η λέξη «ποντίκι» μπορεί να αναφέρεται είτε στο τρωκτικό είτε στο hardware που χρησιμοποιούμε για να ελέγχουμε την οθόνη του υπολογιστή μας, με τη συγκεκριμένη σημασία να προκύπτει μόνο μέσα από την εξέταση της οπτικής σκηνής όπου χρησιμοποιείται η λέξη.

Η αντιμετώπιση του V-WSD απαιτεί την ανάπτυξη προηγμένων αλγορίθμων που είναι ικανοί να αναλύουν ταυτόχρονα κείμενο και εικόνα, προσδιορίζοντας τις λεπτές σχέσεις μεταξύ τους. Αυτό συμπεριλαμβάνει την κατανόηση του πλαισίου, την αναγνώριση αντικειμένων και την ερμηνεία της σκηνής για να καθοριστεί η σωστή σημασία της κάθε λέξης. Η επιτυχία στην οπτική WSD ενισχύει τις δυνατότητες των συστημάτων σε περιβάλλοντα όπου το κείμενο και η εικόνα συνυπάρχουν, βελτιώνοντας την αυτοματοποιημένη ανάλυση και την παραγωγή περιεχομένου.

Semeval 2023 Task 1

Ο διαγωνισμός SemEval (Semantic Evaluation) παρέχει μια πλατφόρμα για την αξιολόγηση και σύγκριση συστημάτων σημασιολογικής ανάλυσης, περιλαμβάνοντας ειδικά tasks που στοχεύουν στην οπτική αποσαφήνιση σημασιών λέξεων (VWSD). Αυτές οι προκλήσεις αναδεικνύουν τη σημασία της ενσωμάτωσης της οπτικής πληροφορίας στην επεξεργασία φυσικής γλώσσας και προσφέρουν ευκαιρίες για την εξέλιξη των τεχνολογικών προσεγγίσεων σε αυτό το πεδίο. Μερικές από τις πιο δημοφιλείς μεθόδους για την αντιμετώπιση της VWSD περιλαμβάνουν τη χρήση μεγάλων μοντέλων γλώσσας που έχουν εκπαιδευτεί σε πλούσια σετ δεδομένων τα οποία περιλαμβάνουν κείμενο και σχετικές εικόνες, επιτρέποντας την αποτελεσματική εξαγωγή σημασιών από σύνθετες οπτικο-κειμενικές πληροφορίες. Επιπλέον, η εφαρμογή τεχνικών βαθιάς μάθησης και νευρωνικών δικτύων που εστιάζουν στην οπτική ανάλυση και στην κατανόηση της γλώσσας είναι άλλος ένας δημοφιλής τρόπος για την επίλυση του προβλήματος, καθώς επιτρέπουν τη λεπτομερή ερμηνεία των σχέσεων μεταξύ κειμένου και εικόνας.

1.2 Κοινώς Χρησιμοποιούμενες Τεχνικές

Στην αντιμετώπιση της οπτικής αποσαφήνισης σημασιών λέξεων (V-WSD), έχουν αναπτυχθεί διάφορες μεθοδολογίες με σκοπό τη βελτίωση της απόδοσης των πολυτροπικών μοντέλων στην κατανόηση της σχέσης μεταξύ κειμένου και εικόνας. Παρακάτω περιγράφονται μερικές από αυτές τις μεθοδολογίες:

- Αρχιτεκτονικές

- **Vanilla CLIP**: Το αρχικό μοντέλο CLIP από την OpenAI αποτελεί μια βασική μεθοδολογία, η οποία χρησιμοποιεί τεχνικές contrastive learning για να συσχετίσει εικόνες με τις αντίστοιχες περιγραφές τους στο κείμενο, δίνοντας έμφαση στην ολοκληρωμένη κατανόηση των δύο τρόπων πληροφορίας.

- **Wiki-Enhanced CLIP**: Αυτή η προσέγγιση επεκτείνει τον Vanilla CLIP με την ενσωμάτωση γνώσης από τη Wikipedia, βελτιώνοντας την ικανότητα του μοντέλου να κατανοεί και να αποσαφηνίζει τη σημασία λέξεων με βάση πιο πλούσιες και πολύπλοκες κειμενικές περιγραφές, καθώς και την αναφορικότητα με την οπτική πληροφορία.

- **LiT (Language-Image Transformer)**: Το LiT (Locked-image Text Tuning) εισάγει μια πρωτοποριακή τεχνική στην επιστήμη της επεξεργασίας οπτικο-γλωσσικών πληροφοριών, βελτιστοποιώντας την αναπαράσταση του κειμένου ενώ κρατά τα οπτικά χαρακτηριστικά αμετάβλητα. Αυτός ο μηχανισμός ενισχύει σημαντικά τη δυνατότητα του μοντέλου να συγχρονίζει με ακρίβεια τις κειμενικές περιγραφές με το οπτικό περιεχόμενο, επιτυγχάνοντας εξειδικευμένη ευαισθησία σε εργασίες απαιτούντες λεπτομερή συσχέτιση μεταξύ κειμένου και εικόνων.

- **ViLT (Vision and Language Transformer)**: Το ViLT αποτελεί μια εξέλιξη της πολυτροπικής επεξεργασίας, η οποία επικεντρώνεται στην αποδοτική ενσωμάτωση οπτικών και κειμενικών δεδομένων χωρίς την ανάγκη για περίπλοκες προ-επεξεργασίες, βελτιστοποιώντας τόσο την απόδοση όσο και την ταχύτητα.

- **Vision Text Dual Encoder (VTDE)**: Το VTDE χρησιμοποιεί δύο ξεχωριστούς encoders, έναν για την επεξεργασία του κειμένου και έναν για την επεξεργασία της εικόνας, με σκοπό τη δημιουργία ενός κοινού χώρου χαρακτηριστικών όπου η σημασία των δύο τρόπων πληροφορίας μπορεί να συσχετιστεί αποτελεσματικά.

Αυτές οι προσεγγίσεις επιδεικνύουν την ευρύτητα και την ποικιλομορφία των τεχνικών που μπορούν να χρησιμοποιηθούν για την επίτευξη πιο προηγμένης κατανόησης και αλληλεπίδρασης. Περσσότερες λεπτομέρειες αναφέρονται στο υπόλοιπο της διπλωματικής αυτής.

1.3 Προτεινόμενο Μοντέλο Arpa

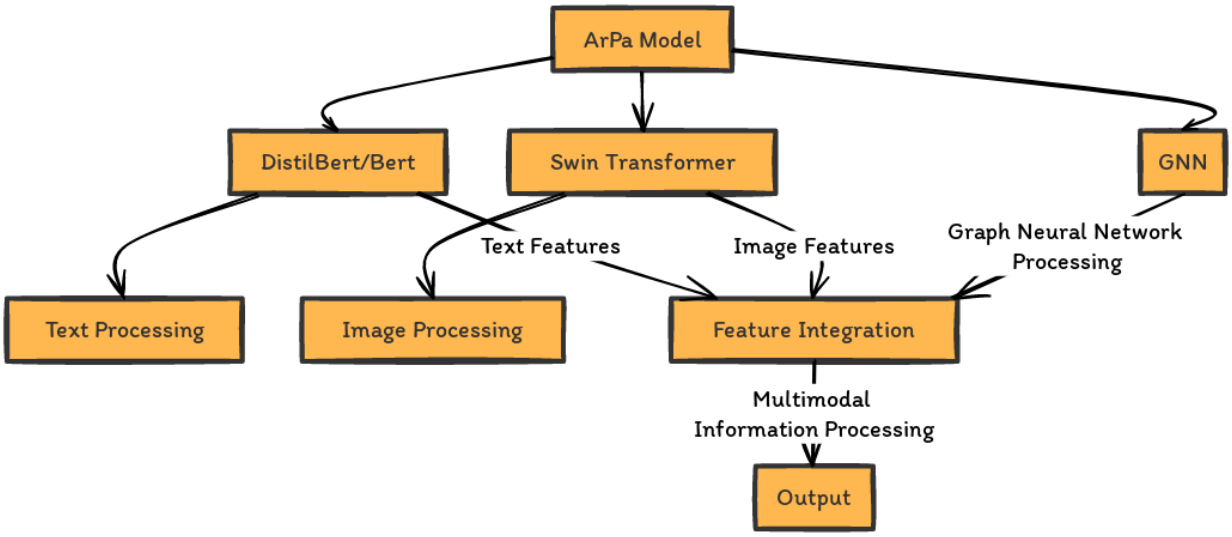

Figure 1.3.1: Αρχιτεκτονική Μοντέλου ArPa

Συγνώμη για το λάθος. Ας προχωρήσουμε με τη σωστή περιγραφή του ArPa με βάση τα στοιχεία που παρέχετε:

Το ArPa αναφέρεται σε μια πρωτοποριακή αρχιτεκτονική στον τομέα της τεχνητής νοημοσύνης, η οποία ενσωματώνει επίπεδα του DistilBERT, του Swin Transformer και των Γραφηματικών Νευρωνικών Δικτύων (GNN). Αυτή η σύνθετη αρχιτεκτονική επιδιώκει να συνδυάσει τα πλεονεκτήματα κάθε μοντέλου για να επιτύχει υψηλή απόδοση σε εργασίες που απαιτούν πολυπλοκότητα και ευρυγνωσία, όπως η αποσαφήνιση σημασιών λέξεων σε οπτικά δεδομένα.

Ο DistilBERT, ως ελαφρύτερη εκδοχή του BERT, προσφέρει τη δυνατότητα για γρήγορη και αποδοτική επεξεργασία κειμένου, διατηρώντας παράλληλα σημαντικά επίπεδα κατανόησης της γλώσσας. Ο Swin Transformer, με την ικανότητά του να χειρίζεται οπτικά δεδομένα μέσω ενός ιεραρχικού μοντέλου που επιτρέπει προσαρμοστική επεξεργασία εικόνων, προσθέτει σημαντική αξία στην ανάλυση και κατανόηση οπτικών πληροφοριών. Τέλος, η ενσωμάτωση Γραφηματικών Νευρωνικών Δικτύων (GNN) ενισχύει την ικανότητα του μοντέλου να αναλύει και να ερμηνεύει τις σχέσεις και τις δομές μέσα στα δεδομένα, προσφέροντας μια βαθιά κατανόηση των συνδέσεων μεταξύ διαφορετικών στοιχείων.

Μέσα από τη συνεργασία αυτών των τριών κομματιών, το ArPa επιδιώκει να προσφέρει μια ολοκληρωμένη λύση για την επίλυση πολύπλοκων προβλημάτων στον τομέα της τεχνητής νοημοσύνης, ενισχύοντας τις δυνατότητες της μηχανικής μάθησης σε εφαρμογές που αφορούν την αποσαφήνιση σημασιών λέξεων με οπτικά δεδομένα και πέραν αυτών.

Μερικά από τα πλεονεκτήματα που προσφέρει αυτή η αρχιτεκτονική:

- **Αξιοποίηση του Swin Transformer για Προηγμένη Επεξεργασία Εικόνων**: Η χρήση του Swin Transformer ('SwinModel') για την επεξεργασία εικόνων είναι μια κρίσιμη τεχνική επιλογή που ενισχύει σημαντικά την απόδοση της ArPa σε οπτικές εργασίες WSD. Οι Swin Transformers σχεδιάζονται για να ανιχνεύουν ιεραρχικά οπτικά χαρακτηριστικά χρησιμοποιώντας σχήματα μετατοπισμένων παραθύρων, επιτρέποντας τον αποδοτικό υπολογισμό της αυτοπροσοχής σε διάφορες κλίμακες της εικόνας. Αυτός ο σχεδιασμός επιτρέπει στην ArPa να επεξεργάζεται εικόνες με τρόπο που ανιχνεύει τόσο τις λεπτομερείς λεπτομέρειες όσο και τις ευρύτερες πληροφορίες πλαισίου, κρίσιμες για την κατανόηση περίπλοκων οπτικών σκηνών.
- Η εισαγωγή ενός στρώματος GNN αντιπροσωπεύει μια σημαντική καινοτομία στην ανάλυση πλαισίου. Αντιμετωπίζοντας τα συνενωμένα χαρακτηριστικά κειμένου και εικόνας ως κόμβους σε ένα γράφημα, και χρησιμοποιώντας ένα GNN για την ανάλυση αυτών των κόμβων στο πλαίσιο των συνδέσεων τους (δηλαδή, των σχέσεων μεταξύ διαφόρων κειμένων και εικόνων), η ArPa μπορεί να κατανοήσει το ευρύτερο πλαίσιο πέρα από τη γραμμική ακολουθία των λέξεων ή το περιεχόμενο των μεμονωμένων εικόνων. Αυτό είναι ιδιαίτερα ισχυρό για τη διασαφήνιση λέξεων που απαιτούν κατανόηση της αλληλεπίδρασης πολλαπλών στοιχείων σε ένα σύνολο δεδομένων.
- Η χρήση freezing των επιπέδων LLM και swin transformer στο pipeline της αρχιτεκτονικής μα δίνει την δυνατότητα να απόφευγουμε περιπτώσεις overfitting μιας και κρατάμε ακέραιες τις προεκπαιδευμένες ικανότητες κάθε υπο-μοντέλου.

1.4 Πειραματικό Μέρος

1.4.1 Σύνολο δεδομένων εκπαίδευσης

Το σύνολο δεδομένων Visual-WSD αναπτύχθηκε με στόχο τη βαθύτερη κατανόηση της διαδικασίας αποσαφήνισης οπτικών εννοιών, συνδυάζοντας οπτικά και κειμενικά στοιχεία σε ένα πλούσιο και πολυγλωσσικό πλαίσιο. Μέσω της ενσωμάτωσης πολύτιμων πηγών δεδομένων, όπως το Wikidata, το OmegaWiki και το BabelPic, αυτό το σύνολο δεδομένων προσφέρει μια ξεχωριστή βάση για την ανάπτυξη προηγμένων μοντέλων που είναι ικανά να αποκωδικοποιούν την περίπλοκη αλληλεπίδραση μεταξύ κειμένου και εικόνας.

Η κατασκευή του συνόλου δεδομένων περιλαμβάνει τη συλλογή και την επεξεργασία εικόνων και λεξιλογικών στοιχείων που αντιπροσωπεύουν διάφορες εννοίες, εξασφαλίζοντας μια δυναμική βάση για την πρόκληση και την ανάπτυξη μοντέλων πολυτροπικής κατανόησης. Η χρήση των πηγών αυτών διευκολύνει την πειραματική εργασία με μοντέλα όπως το WikiCLIP, το οποίο αναμένεται να επιδείξει αυξημένη απόδοση λόγω της κοινής βάσης δεδομένων στην οποία έχει εκπαιδευτεί.

Για τη δημιουργία δεδομένων εκπαίδευσης, το έργο χρησιμοποιεί τη δομή του σημασιολογικού δικτύου του BabelNet για να παράγει δεδομένα ποιότητας "silver" στα Αγγλικά. Αυτό περιλαμβάνει την επιλογή εννοιών —τόσο αμφίσημων όσο και μονοσήμαντων—από την ενότητα WordNet του BabelNet, μαζί με αντίστοιχες εικόνες. Οι συναφείς υποδείξεις προέρχονται από τα υπερώνυμα κάθε έννοιας, με πρόσθετο φιλτράρισμα για την αποκλεισμό εικόνων ανθρώπινων προσώπων μέσω της κατηγοριοποίησης του WordNet. Αυτή η προσέγγιση διασφαλίζει ότι τα δεδομένα εκπαίδευσης είναι πλούσια σε περιεχόμενο και διαφορετικά σε οπτική αναπαράσταση. Τα δεδομένα εκπαίδευσης που παρέχονται για αυτή την κοινόχρηστη εργασία αποτελούνται από ένα σύνολο δεδομένων silver με 12,869 περιπτώσεις V-WSD. Κάθε δείγμα είναι ένα 4-τουπλέτες $\langle f, c, I, i^* \in I \rangle$ όπου $|I| = 10$. Επιλέγουμε τυχαία το 10% των δεδομένων εκπαίδευσης για χρήση ως σετ ανάπτυξης.

- Δεδομένα Δοκιμής

Τα δεδομένα δοκιμής πρότυπα "gold" είναι προσεκτικά προετοιμασμένα στα Αγγλικά, τα Περσικά και τα Ιταλικά, τονίζοντας την πολύγλωσση ικανότητα του σετ δεδομένων. Η διαδικασία αρχίζει με τη συλλογή μιας λίστας αμφίσημων αισθήσεων λέξεων από το BabelNet, πλήρεις με εικόνες και ορισμούς. Οι σημειωτές, οι οποίοι έχουν εξειδίκευση στα Αγγλικά, τα Φαρσί και τα Ιταλικά, στη συνέχεια παρέχουν μία ή δύο λέξεις-κλειδιά. Αυτές οι λέξεις επιλέγονται με προσοχή για να διασφαλίζεται ότι είναι ενδεικτικές της αίσθησης της λέξης όταν συνδυάζονται με τον ορισμό και την εικόνα, χωρίς να είναι υπερβολικά αποκαλυπτικές.

1.4.2 Μετρικές απόδοσης

Για να αξιολογήσουμε αυστηρά την αποτελεσματικότητα και την ακρίβεια των μοντέλων, χρησιμοποιούμε ένα σύνολο μετρήσεων αξιολόγησης που έχουν σχεδιαστεί για να αποτυπώνουν την απόδοση των μοντέλων από διαφορετικές οπτικές γωνίες. Οι κύριες μετρήσεις που χρησιμοποιούνται για τη μέτρηση της αποτελεσματικότητας του μοντέλου στο πλαίσιο V-WSD είναι: το ποσοστό επιτυχίας (ή την ακρίβεια (accuracy) top-1) και τη μέση αμοιβαία κατάταξη (MRR). Αυτές οι μετρήσεις προσφέρουν πληροφορίες όχι μόνο για την ακρίβεια των μοντέλων αλλά και για την ικανότητά τους να εντοπίζουν με συνέπεια τη σωστή εικόνα ανάμεσα σε ένα σύνολο δυνατοτήτων.

Οι μαθηματικοί τύποι που εκφράζουν αυτές τις μετρικές είναι:

$$\text{Ακρίβεια (accuracy)} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Instances}}$$

- Μέση Αμοιβαία Κατάταξη (MRR)

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}$$

όπου N το πλήθος δεδομένων, και rank_i είναι η θέση της σωστής εικόνας στη λίστα κατάταξης υποψήφιων εικόνων του μοντέλου για την iη παρουσία.

1.4.3 Αποτελέσματα και Στατιστικά

Τα πειράματα μας χωρίζονται σε 2 κατηγορίες. Συγκεκριμένα με το αν έχουμε προεπεξεργαστεί τα δεδομένα εισόδου, εφαρμόζοντας βασικές τεχνικές όπως εμπλουτισμός με συνώνυμα, καθαρισμός από καταλήξεις λέξεων, άρθρα, αντωνυμίες κλπ.

'Ετσι καταλήγουμε στα παρακάτω:

Στατιστικά μοντέλων εκπαιδευμένα χωρίς καθαρισμό και κανονικοποίηση των δεδομένων εισόδου

Model	**Accuracy (%)**	**MRR (%)**
(Vanilla) CLIP	41.2	67.4
Wiki CLIP	62.47	71.64
Vilt	7	11.2
Vision Text Dual Encoder	15.09	21
LiT	68	74.92
ArPa	82.3	88.1852

Table 1.1: Performance of Models using unprocessed data

Αν ρίξουμε μια ματιά στον πίνακα των στατιστικών μας και στα διαγράμματα που παρέχονται στην επόμενη σελίδα, μπορούμε τόσο να επικυρώσουμε τα αποτελέσματα με βάση τη γνώση μας για την αρχιτεκτονική κάθε ενός από τα μοντέλα αλλά και να συλλέξουμε κάποιες νέες πληροφορίες. Για παράδειγμα:

- Η στατιστική υπεροχή του ArPa μπορεί να αποδοθεί ποσοτικά στα μοναδικά αρχιτεκτονικά του χαρακτηριστικά. Συγκεκριμένα, η ενσωμάτωση του DistilBert και Swin Transformer για την επεξεργασία κειμένου και εικόνων αντίστοιχα, μαζί με ένα GNN για την σύνθεση χαρακτηριστικών, παρουσιάζει μια πειστική χρήση των σχεσιακών δεδομένων.
- Το άλμα στην επίδοση από το LiT στο ArPa (από περίπου 68
- Η επίδοση του LiT, αποδεικνύει **την αποτελεσματικότητα των προσαρμοσμένων μηχανισμών προσοχής στην επεξεργασία πολυτροπικών εισόδων**. Η αρχιτεκτονική του, σχεδιασμένη για βαθύτερη ένταξη γλώσσας-εικόνας, επικυρώνει στατιστικά την προϋπόθεση ότι μια πιο στενή και λεπτομερής αλληλεπίδραση μεταξύ των τρόπων ενισχύει την επίδοση.
- Η στατιστική διαφορά μεταξύ του Vanilla CLIP και του Wiki CLIP **επισημαίνει τον αντίκτυπο του εμπλουτισμού του περιεχομένου στην επίδοση του μοντέλου**. Η περίληψη δεδομένων από τη Wikipedia στο Wiki CLIP παρέχει μια αισθητή βελτίωση τόσο στην ακρίβεια όσο και στο MRR, **επικυρώνοντας στατιστικά την αξία των εξωτερικών πηγών γνώσης στην ενίσχυση της κατανόησης και της ακρίβειας διαλεύκανσης του μοντέλου**.
- Το ViLT και το Vision Text Dual Encoder εμφανίζουν σχετικά χαμηλότερες στατιστικές επιδόσεις, υποδηλώνοντας πιθανούς αρχιτεκτονικούς περιορισμούς στην ιδανική ισορροπία ή ενσωμάτωση πολυτροπικών πληροφοριών

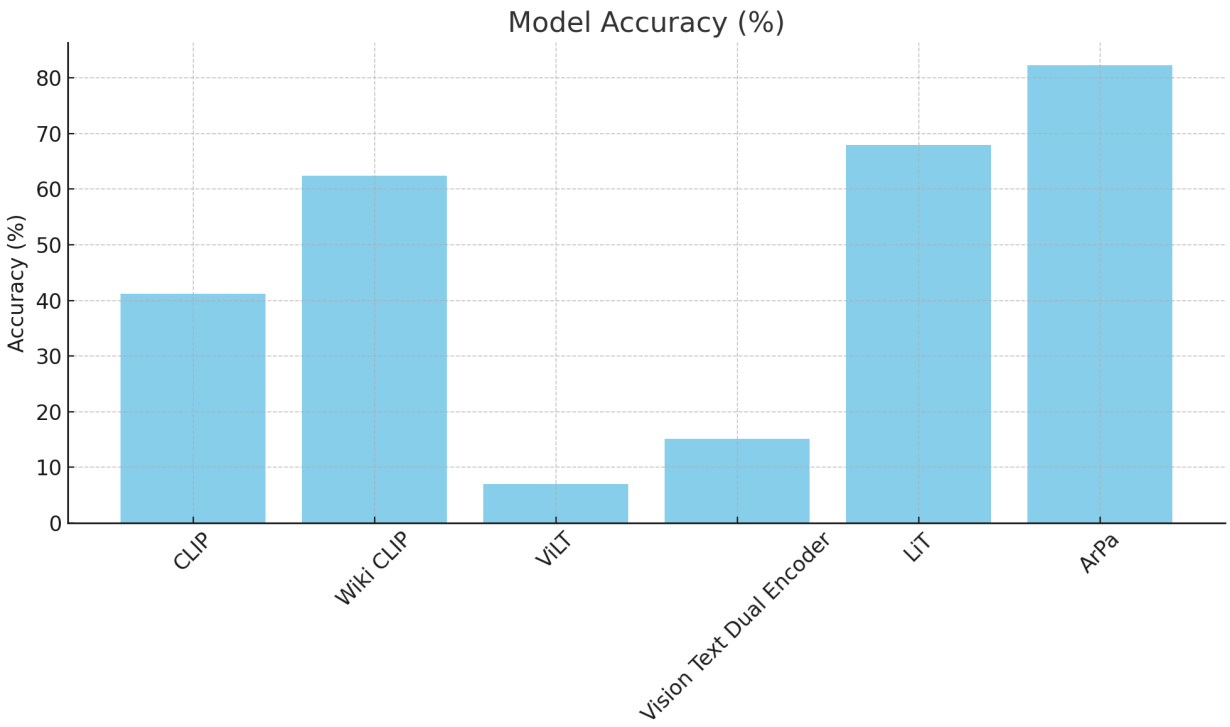

Figure 1.4.1: Ακρίβεια αρχιτεκτονικών που χρησιμοποιούν μη-προεπεργασμένα δεδομένα εισόδου

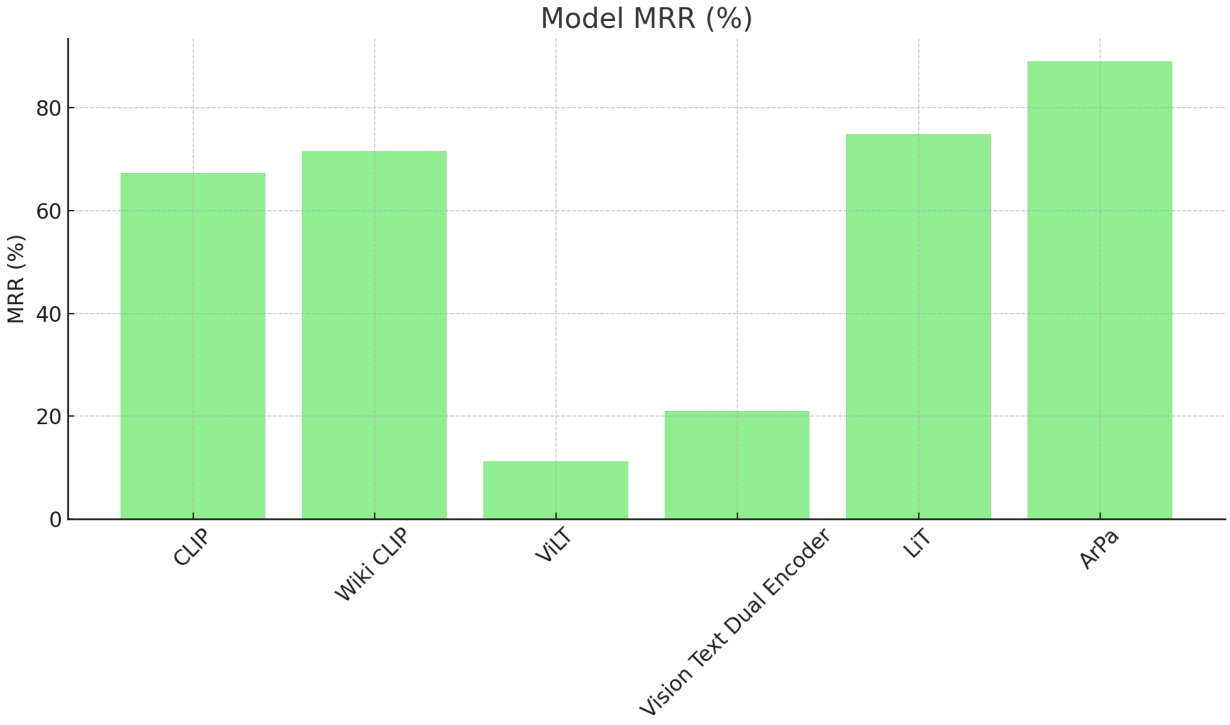

Figure 1.4.2: MRR αρχιτεκτονικών που χρησιμοποιούν μη-προεπεργασμένα δεδομένα εισόδου

Preprocessing of Input Data

Model	Accuracy (%)	MRR (%)
(Vanilla) CLIP	60.48	73.88
Wiki CLIP	62.47	74.92
Vilt	7.2	12
Vision Text Dual Encoder	19	22.8
LiT	71	79.81
ArPa	87.98	92.03

Table 1.2: Performance of Models using preprocessed data

Είναι εύκολο να δούμε ότι υπάρχει ορατή βελτίωση στις ακρίβειες και στα μετρικά MRR των περισσότερων μοντέλων μας. Αυτό λέει πολλά για το τι μπορούν να επιτύχουν η καθαριότητα δεδομένων και η κανονικοποίηση στον χώρο της Οπτικής Διαλεύκανσης Λέξεων. Πιο συγκεκριμένα:

- **Τόσο το Vanilla CLIP όσο και το Wiki CLIP**, βλέπουν βαθμιαίες βελτιώσεις από τα οφέλη της προεπεξεργασίας λόγω της γενικιστικής τους αρχιτεκτονικής που αξιοποιεί και τα κειμενικά και τα οπτικά δεδομένα για μάθηση. **Οι εμπλουτισμένοι κειμενικοί πλαισιοι πιθανότατα βοηθούν στην παροχή πιο λεπτομερών σημασιολογικών ενδείξεων, ενώ η προηγμένη προεπεξεργασία εικόνας διασφαλίζει ότι τα οπτικά χαρακτηριστικά είναι πιο πληροφοριακά**. Αυτές οι βελτιώσεις εξηγούν την παρατηρούμενη σταθερή επίδοση, αν και η γενική φύση των μοντέλων μπορεί να περιορίζει την ικανότητά τους να αξιοποιήσουν πλήρως τα πιο πλούσια δεδομένα σε σύγκριση με πιο εξειδικευμένα μοντέλα.
- **Η σχετικά χαμηλότερη επίδοση του ViLT (και παρόμοια του Vision Text Dual Encoder), παρά την προεπεξεργασία, μπορεί να υποδηλώνει αρχιτεκτονικούς περιορισμούς στη διαχείριση των εμπλουτισμένων πολυτροπικών δεδομένων**. Ενώ το μοντέλο είναι σχεδιασμένο να ενσωματώνει οπτικές και γλωσσικές εισόδους, η πολύπλοκη φύση των επεκταμένων πλαισίων και των βελτιωμένων χαρακτηριστικών εικόνας μπορεί να μην αξιοποιηθεί πλήρως λόγω πιθανών περιορισμών στους μηχανισμούς ενσωμάτωσής τους ή στον τρόπο που επεξεργάζεται τις ιεραρχικές πληροφορίες.
- Όσον αφορά το ArPa και το LiT, είναι σαφές ότι οι μέθοδοι προεπεξεργασίας ταιριάζουν τέλεια με τα αρχιτεκτονικά τους πλεονεκτήματα, παρέχοντας υψηλής ποιότητας, εμπλουτισμένες εισόδους που μπορούν να ενσωματώσουν και να αναλύσουν αποτελεσματικά.

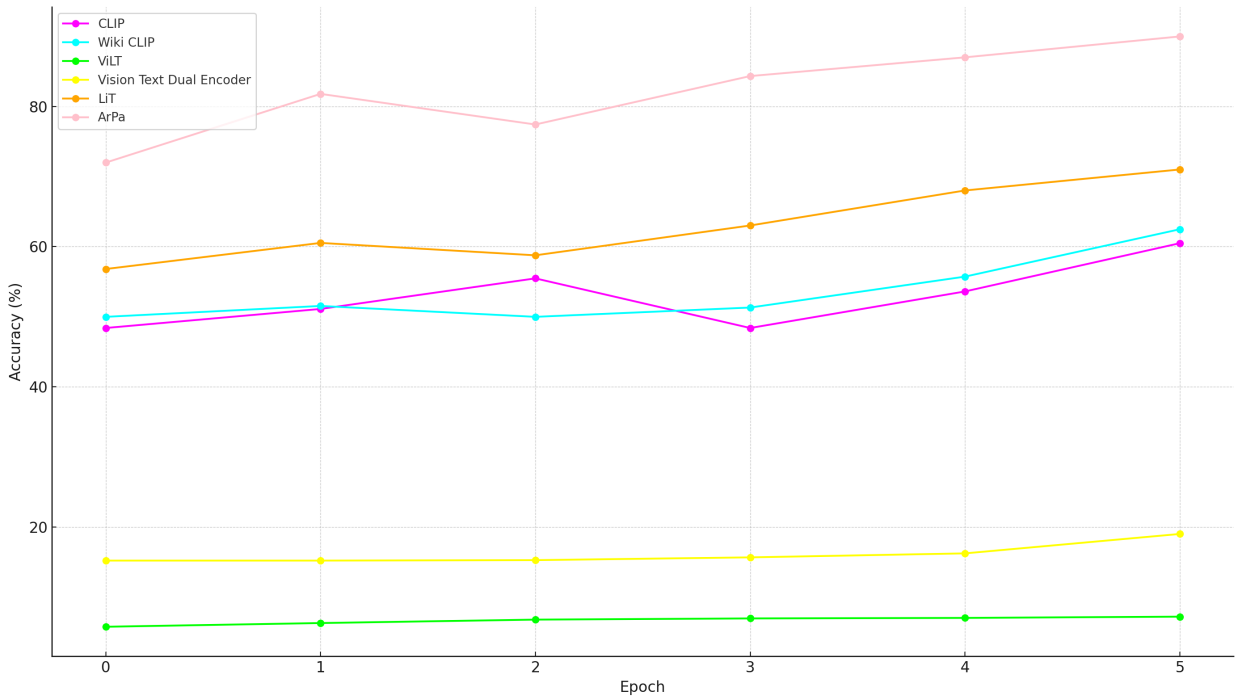

Figure 1.4.3: Ακρίβεια αρχιτεκτονικών που χρησιμοποιούν προεπεργασμένα δεδομένα εισόδου

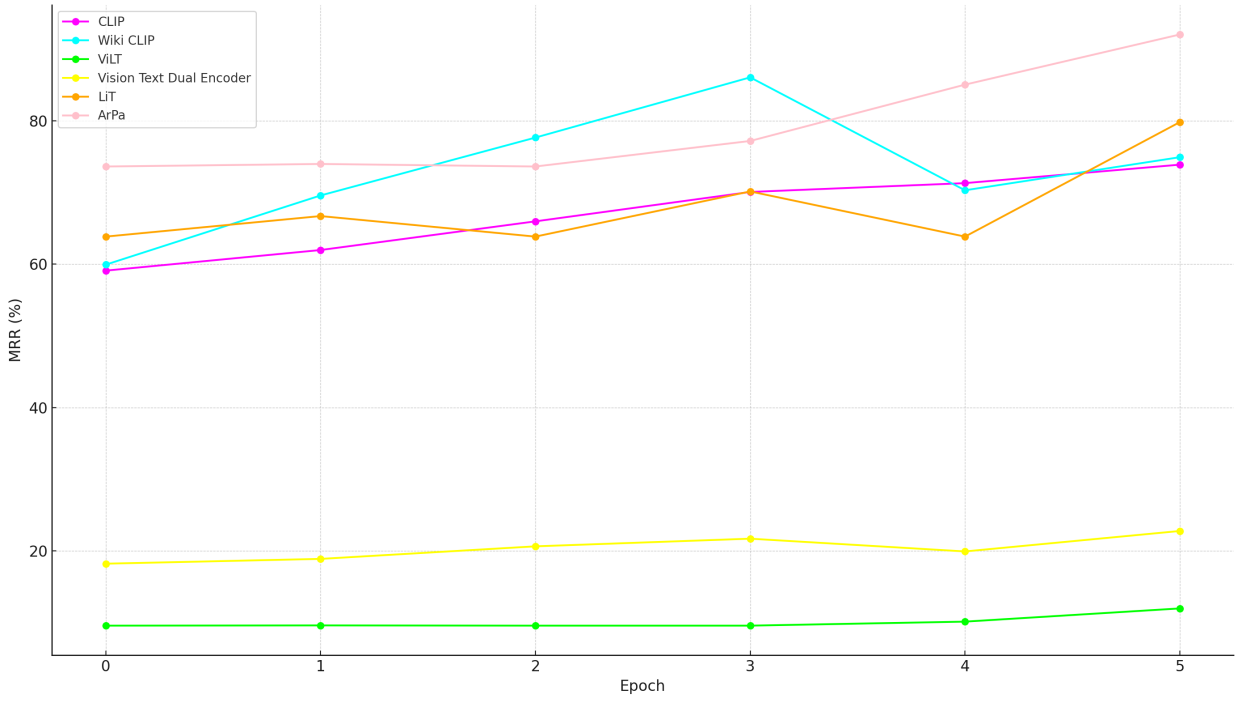

Figure 1.4.4: MRR αρχιτεκτονικών που χρησιμοποιούν μη-προεπεργασμένα δεδομένα εισόδου

1.5 Συμπεράσματα

1.6 Συζήτηση

Αυτή η διατριβή αποτελεί μια βαθιά κατάδυση στον κόσμο της Αποσαφήνισης Οπτικών Εννοιών(V-WSD), μια δύσκολη αλλά συναρπαστική τομή της υπολογιστικής όρασης και της επεξεργασίας φυσικής γλώσσας. Μέσω αυτής της εξερεύνησης, αξιολογήσαμε κριτικά μια σειρά από προηγμένα υπολογιστικά μοντέλα, με έμφαση στο καινοτόμο μοντέλο ArPa. Αυτό το ταξίδι περιελάμβανε επίσης εκτενείς αναλύσεις άλλων σημαντικών μοντέλων όπως το Vanilla CLIP, Wiki CLIP, LiT, ViLT, και το Vision Text Dual Encoder. Η προσπάθεια δεν ήταν απλώς μια ακαδημαϊκή άσκηση αλλά μια αναζήτηση για την αποκάλυψη των πολυεπίπεδων πολυπλοκοτήτων του V-WSD, τονίζοντας τους κρίσιμους ρόλους που διαδραματίζει η αρχιτεκτονική του μοντέλου και οι προηγμένες τεχνικές προεπεξεργασίας στην ενίσχυση της πολυτροπικής κατανόησης.

Μία από τις πιο ενδιαφέρουσες ανακαλύψεις που προέκυψαν από αυτή την έρευνα είναι η αναμφισβήτητη επιρροή του αρχιτεκτονικού πλαισίου ενός μοντέλου στην ικανότητά του να επεξεργάζεται και να ενσωματώνει πολύπλευρα πολυτροπικά δεδομένα. Το μοντέλο ArPa αποτελεί μαρτυρία αρχιτεκτονικής πρωτοπορίας, συνδυάζοντας τις ακλόνητες ικανότητες του DistilBert και Swin Transformer με ένα Γραφικό Νευρωνικό Δίκτυο (GNN) για μια ασύγκριτη σύνθεση συγχώνευσης δεδομένων. Αυτή η αρχιτεκτονική συνέργεια προώθησε το μοντέλο ArPa σε νέα ύψη επιδόσεων, θέτοντας ένα σημείο αναφοράς για μελλοντικές προσπάθειες στον τομέα της πολυτροπικής αλληλεπίδρασης και της γνωστικής κατανόησης από μηχανές.

Η έμφαση στη βελτίωση των μεθόδων προεπεξεργασίας, συμπεριλαμβανομένης της επέκτασης του γλωσσικού πλαισίου και της προηγμένης επεξεργασίας εικόνων, αναδείχθηκε ως γωνιακός λίθος της στρατηγικής μας για την βελτιστοποίηση των δεδομένων για την κατανάλωση από το μοντέλο. Αυτή η λεπτομερής προσέγγιση στην προετοιμασία δεδομένων βελτίωσε σημαντικά την ικανότητα των μοντέλων να διαπλέουν με επιδεξιότητα τα περίπλοκα τοπία του V-WSD, επιτυγχάνοντας υψηλότερη ακρίβεια και αποδοτικότητα.

Η αυστηρή πειραματική διαδικασία και η αναλυτική εξέταση που πραγματοποιήθηκε σε αυτή τη διατριβή προσέφεραν πολύτιμες διορατικότητες και μεθοδολογίες, συμβάλλοντας σημαντικά στην ευρύτερη συζήτηση για την τεχνητή νοημοσύνη και την πολυτροπική μάθηση. Αυτή η εργασία διαφώτισε τον περίπλοκο χορό μεταξύ των οπτικών και κειμενικών δεδομένων, παρέχοντας συγκεκριμένες, βασισμένες σε αποδείξεις στρατηγικές για την πλοήγηση και την υπέρβαση των προκλήσεων που είναι έμφυτες στο V-WSD.

Κλείνοντας αυτή την έρευνα, είναι προφανές ότι το ταξίδι δεν μας έχει μόνο διευρύνει την κατανόηση της Οπτικής Διαλεύκανσης Αισθήσεων Λέξεων αλλά έχει επίσης φωτίσει το μονοπάτι για μελλοντική εξερεύνηση στην αρμονική ενσωμάτωση οπτικών και κειμενικών πληροφοριών. Αναγνωρίζοντας και διατυπώνοντας τους κύριους παράγοντες της επίδοσης των μοντέλων στις εργασίες V-WSD, αυτή η διατριβή καθιερώνει ένα θεμέλιο σχέδιο για την εξέλιξη πιο λεπτεπίλεπτων, συμπαθητικών και ευαίσθητων στο πλαίσιο συστημάτων τεχνητής νοημοσύνης. Βρισκόμενοι στο όριο νέων ανακαλύψεων, οι διορατικότητες που αποστάζονται από αυτή τη διατριβή μας προσκαλούν να φανταστούμε ένα μέλλον όπου η αρμονική σύμπλευση της γλώσσας και της όρασης υπερβαίνει τα τρέχοντα όρια, προάγοντας μια εποχή τεχνολογικής προόδου και νέων μορφών αλληλεπίδρασης. Έτσι, αυτή η διατριβή δεν είναι απλώς η κορύφωση της έρευνας αλλά ένας φάρος που μας οδηγεί προς το ανεκμετάλλευτο δυναμικό της πολυτροπικής τεχνητής νοημοσύνης.

1.7 Μελλοντική Εργασία

Κλείνοντας αυτή τη διατριβή, θα θέλαμε να προτείνουμε μερικές κατευθύνσεις για να βελτιωθεί περαιτέρω αυτή η εργασία ή να εμπνεύσει διαφορετικές και ενδιαφέρουσες προσεγγίσεις. Μελλοντικές έρευνες θα πρέπει να στοχεύουν στην ανάπτυξη ακόμη πιο προηγμένων πολυτροπικών μοντέλων που μπορούν να ενσωματώσουν άρρηκτα τις οπτικές και κειμενικές πληροφορίες με μεγαλύτερους βαθμούς ακρίβειας και λεπτομέρειας. Εμπνευσμένοι από την επιτυχία του μοντέλου ArPa, οι έρευνες σε νέες αρχιτεκτονικές που εκμεταλλεύονται τις τελευταίες προόδους στη μηχανική μάθηση—όπως βαθύτερα Γραφικά Νευρωνικά Δίκτυα, ενισχυμένα μοντέλα μετασχηματιστών και καινοτόμους μηχανισμούς προσοχής—θα είναι κρίσιμες. Αυτά τα μοντέλα δεν θα πρέπει μόνο να εξελίσσονται σε εργασίες V-WSD αλλά και να δείχνουν προσαρμοστικότητα και γενικευσιμότητα σε διάφορους τομείς και σύνολα δεδομένων.

Η ενσωμάτωση μιας ευρύτερης ποικιλίας εξωτερικών πηγών γνώσης, πέρα από τη Wikipedia και το WordNet, θα μπορούσε να παρέχει στα μοντέλα μια πιο πλούσια εννοιολογική κατανόηση και να τα ενεργοποιήσει να αντιμετωπίζουν πιο σύνθετες προκλήσεις διαλεύκανσης. Μελλοντικές εργασίες θα μπορούσαν να εξερευνήσουν την ενσωμάτωση των εγκυκλοπαιδικών πληροφοριών ειδικού τομέα, πολιτιστικών βάσεων δεδομένων, ακόμη και πολυμέσων περιεχομένου ως μέρος της φάσης προεπεξεργασίας, εμπλουτίζοντας τη βάση γνώσεων των μοντέλων και ενισχύοντας το ερμηνευτικό τους βάθος. Εξερεύνηση Μη Επιβλεπόμενων και Ημι-Επιβλεπόμενων Μεθόδων Μάθησης

Δεδομένης της συχνά περιορισμένης διαθεσιμότητας unlabeled δεδομένων σε συγκεκριμένους τομείς, μελλοντικές έρευνες θα μπορούσαν να ωφεληθούν από την εξερεύνηση μη επιβλεπόμενων και ημι-επιβλεπόμενων μεθόδων μάθησης. Αυτές οι μέθοδοι, που μπορούν να εκμεταλλευτούν μεγάλες ποσότητες unlabeled δεδομένων, θα μπορούσαν να ανοίξουν νέες διαδρομές για την εκπαίδευση και την εξέλιξη των μοντέλων, ιδιαίτερα σε περιοχές όπου η χειροκίνητη επισημείωση είναι ανεφάρμοστη.

Τέλος, η βελτίωση των στρατηγικών μεταφοράς μάθησης για να διευκολυνθεί η άνευ ραφής προσαρμογή των μοντέλων σε νέες εργασίες και τομείς αποτελεί άλλη μια υποσχόμενη κατεύθυνση για μελλοντικές έρευνες. Μέσω της βελτίωσης αυτών των στρατηγικών, οι ερευνητές μπορούν να αυξήσουν την αποδοτικότητα και την αποτελεσματικότητα των διαδικασιών εκπαίδευσης των μοντέλων, επιτρέποντας ταχύτερη ανάπτυξη και ευρύτερη εφαρμογή των τεχνολογιών V-WSD.

Ουσιαστικά, το μέλλον της Αποσαφήνισης Οπτικών Εννοιών και της πολυτροπικής μάθησης είναι γεμάτο ευκαιρίες για επαναστατικές έρευνες και μετασχηματιστικές εφαρμογές. Το θεμέλιο που έθεσε αυτή η διατριβή λειτουργεί ως βήμα για μελλοντικές προσπάθειες, προσκαλώντας τους επιστήμονες και τους επαγγελματίες να επεκταθούν σε ανεξερεύνητα εδάφη της τεχνητής νοημοσύνης.

Chapter 2

Introduction

This chapter is designed to provide the reader with the fundamental knowledge, necessary to understand the rest of the research work. This chapter will lay the groundwork by introducing the principles of visual word disambiguation, graph neural networks, and the basics of artificial intelligence. This section essentially forms the backbone of the thesis, linking these various concepts together in the context of stock market prediction.

The field of natural language processing (NLP) has witnessed remarkable advancements in recent years, driven by the development of sophisticated computational models capable of understanding and generating human language with unprecedented accuracy. Among these developments, the challenge of visual word disambiguation stands out as a critical frontier. This involves the process of accurately identifying the meaning of words that have multiple interpretations based on their visual context. For example, the word **bark** in the following example has different interpretations in different contexts:

- I hope her dog doesn't **bark** when I knock on the door.
- The tree **bark** is rough to the touch.
- I love eating pretzels covered with almond **bark**

As the digital world becomes increasingly visual, with an explosion of images and videos accompanied by textual data, the ability to effectively disambiguate words within these visual contexts becomes imperative. This thesis presents a comprehensive exploration of visual word disambiguation, proposing a novel hybrid approach that leverages the strengths of Large Language Models (LLMs), Transformers, and introduces the innovative Hybrid ArPa Model.

The significance of tackling visual word disambiguation cannot be overstated. In an era where visual content dominates digital communication, from social media platforms to educational resources, the integration of text and imagery has become ubiquitous. However, the inherent ambiguity in language poses a significant challenge, as many words bear multiple meanings depending on their context. Traditional text-based disambiguation techniques have made substantial progress, yet they often fall short when applied to complex visual-textual data. This limitation underscores the necessity for more advanced models capable of interpreting the nuanced interplay between visual elements and textual information.

Enter the realm of Large Language Models and Transformers, technologies at the forefront of NLP. These models have revolutionized our approach to language understanding, offering deep insights into linguistic structures and context. LLMs, with their vast knowledge bases and sophisticated understanding of language nuances, provide a solid foundation for disambiguation tasks. Meanwhile, Transformers, known for their ability to handle sequential data and capture long-distance dependencies, have become indispensable in processing the intricate relationships between words and their visual contexts. However, despite their strengths, these models still face challenges in fully grasping the subtleties of visual word disambiguation, necessitating the development of more specialized approaches.

This thesis introduces the Hybrid ArPa Model, an innovative advancement designed to bridge the gap in visual word disambiguation. The model synergizes the contextual understanding capabilities of LLMs with the structural processing strengths of Transformers and gnn's. At its core, the Hybrid ArPa Model incorporates architectural innovations and processing strategies that enhance the interpretation of visual-textual data. By integrating these elements, the model achieves a superior level of accuracy in disambiguating words within diverse visual environments.

The development of the Hybrid ArPa Model was motivated by the recognition that the complexity of visual word disambiguation requires a multifaceted approach. This model is distinguished by its hybrid nature, combining elements from different technological paradigms to address the multifarious challenges posed by visual-textual data. The integration of a specifically designed visual processing module within the model allows for a more nuanced analysis of images and videos, facilitating a deeper understanding of the context surrounding ambiguous words.

This thesis is structured to provide a thorough understanding of visual word disambiguation, beginning with a comprehensive review of the current state of research in the field. It then delves into the theoretical underpinnings of Large Language Models and Transformers, elucidating their roles in advancing NLP. The core chapters are dedicated to the detailed exposition of the Hybrid ArPa Model, including its conceptual foundation, architectural design, and implementation. Through empirical studies and comparative analysis, the thesis demonstrates the model's efficacy in disambiguating words across a variety of visual contexts, highlighting its potential to transform the landscape of visual language processing. The outline of this thesis is as follows:

- We will firstly provide all the background needed in basic Machine Learning algorithms and concepts of visual word disambiguation, as well as transformers, Large Language Models in order to be able to explain and justify the idea of the use of Graph Neural Networks in an architectural pipeline. After doing so, we will provide a thorough description of GNN variants relevant to this work.
- Explore different models and strategies used to tackle V-wsd tasks
- Lastly, we will propose our **ArPa** architecture for visual word disambiguation based on a pipeline of LLMs, transformers and GNNs. Alongside that we will highlight the performance of our model using different hyperparameters and compare these results with the more conventional methods described in previous sections and draw conclusions based on their expressiveness and evaluation on quantitative and qualitative results.

Chapter 3

Background

This chapter aims to equip the reader with the essential understanding required for grasping the subsequent sections of this research. It establishes a foundation by presenting the key concepts of visual word disambiguation, graph neural networks, large language models, transformers and the foundational elements of data processing. Essentially, this part serves as the core of the thesis, weaving these diverse ideas into a cohesive narrative focused on Visual Word Sense Disambiguation.

In the rapidly evolving field of natural language processing (NLP), the challenge of Visual Word Sense Disambiguation (VWSD) has emerged as a critical area of research, addressing the complexity of understanding words in the context of visual cues. VWSD tackles the intricate task of discerning the meaning of words based on accompanying visual information, a nuanced process that mirrors the human ability to interpret text and images in a cohesive manner.

The primary methods used for Visual Word Sense Disambiguation (VWSD) involve leveraging multimodal data, particularly the integration of textual context and visual cues, to accurately interpret the meaning of words in images. One approach utilizes similarity functions to compare text strings and the alignment between images and text descriptions, computing pairwise similarities among the image-context, image-gloss, and context-gloss. This method aims to identify the best match between a candidate image and word sense, based on a weighted average of these similarities, adjusting the approach through hyperparameters to optimize performance.

This thesis sets out to explore the innovative integration of Graph Neural Networks (GNNs) into VWSD, aiming to enhance the interpretation of textual content within visual contexts. By harnessing the representational power of GNNs, this work seeks to capture the intricate relationships between textual and visual elements, thereby advancing the state-of-the-art in multimodal understanding.

The incorporation of Graph Neural Networks into the realm of VWSD represents a pioneering approach to bridging the gap between textual and visual data. GNNs, known for their efficacy in modeling complex relational data, offer a promising framework for encapsulating the dynamic interplay between words and their associated visual contexts. This thesis delves into the development of novel GNN architectures and methodologies tailored for VWSD, with the goal of achieving a more nuanced and contextually aware disambiguation of words in visual environments. Through a series of experiments and evaluations, this research endeavors to demonstrate the viability and effectiveness of GNNs in enhancing visual word sense disambiguation, setting a new benchmark for multimodal NLP applications.

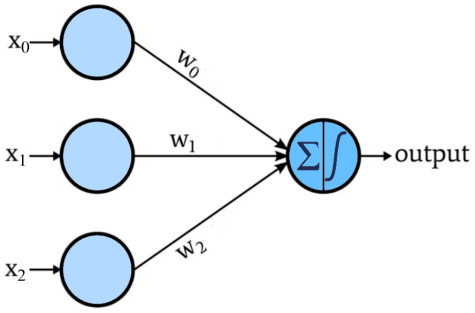

Figure 3.1.1: Illustration of a Perceptron: A Basic Neural Network with a Single Neuron [9]

3.1 Training a Neural Network

This section delves into Neural Networks, a specific area of machine learning that draws inspiration from the functionality of human neural structures. Here, we discuss the fundamental aspects of how such networks learn and the potential obstacles encountered.

3.1.1 Foundational Principles

Initially, we delve into the workings of a basic neural network to lay the groundwork necessary for understanding more complex models tailored for graph-structured data. The concepts introduced primarily pertain to supervised learning tasks, yet they hold relevance across various methodologies explored in this dissertation. The insights provided herein largely stem from [6].

Within the realm of supervised learning, a model leverages a predefined set of N instances from a dataset $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, along with their corresponding labels, to derive a function $f : X \to Y$. This function maps inputs X to outputs Y, with its adjustable parameters frequently referred to as weights. The evaluation of f involves the use of a designated function that gauges training error, known as the loss function L. The training objective centers around optimizing the weights of f to minimize L.

A neural network's neuron output isn't merely the result of input weights' sum x_i; rather, an **activation function** is utilized to modify this sum into an output. The selection of an activation function is crucial for network efficacy and can be either linear or nonlinear, with common examples provided below.

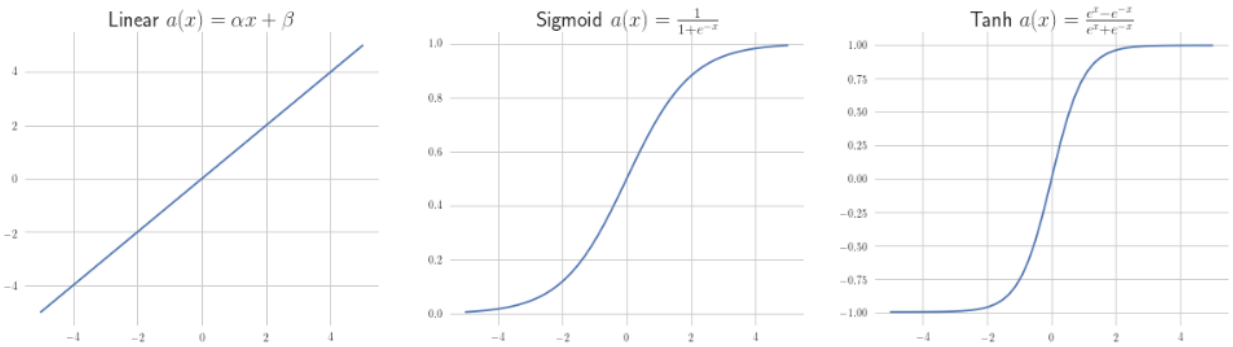

Figure 3.1.2: Various Activation Function Examples.

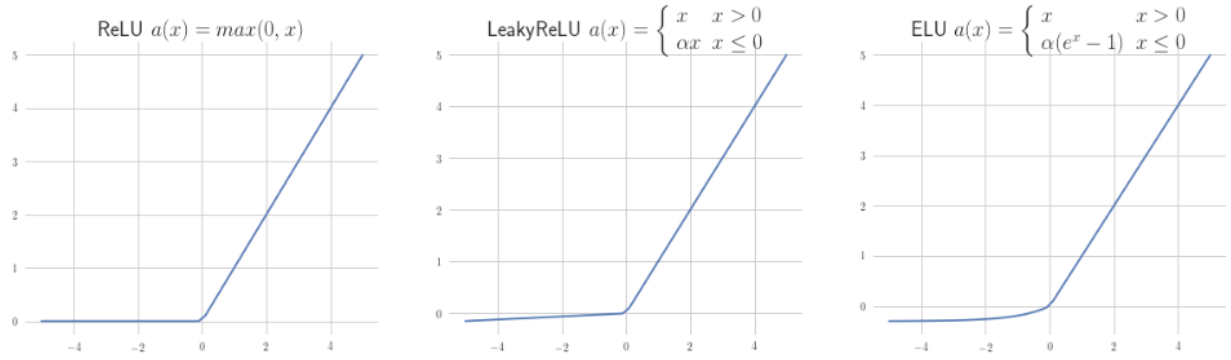

Figure 3.1.3: ReLU Activation Function and Its Variants.

Linear activation functions have notable drawbacks, such as rendering backpropagation infeasible due to their constant derivative, essentially maintaining the weighted sum unaltered. Contrarily, nonlinear activation functions are preferred as they enable the construction of deep networks through layer stacking for complex mappings. Despite their popularity for certain applications, functions like Sigmoid and tanh encounter vanishing gradients issues [22], leading to training instability.

ReLU and its variations stand out as the predominant choice for activation functions, addressing the limitations of prior functions and facilitating computationally efficient training. Variants like LeakyReLU and ELU, depicted in Figure 3.1.3, address the dying ReLU problem, which results in inactive neurons [21].

The **loss function**, or cost function, assesses the model's prediction accuracy by mapping $Y \times Y$ to a non-negative real figure, or $L(y_i, f(x_i))$ for the i_{th} instance. Training involves multiple iterations (epochs) until an objective is met or a maximum iteration count is reached. Ideally, the parameters that minimize this function are identified.

The selection of a cost function is task-dependent. Notable loss functions are summarized below, with custom functions often devised for intricate tasks.

Mean Squared Error serves as a basic yet frequently utilized cost function for regression tasks.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{n} \tag{3.1.1}$$

Mean Absolute Error, or *L1 loss*, similar to MSE, discounts error direction and is more outlier-resistant.

$$MAE = \frac{\sum_{i=1}^{n}|y_i - f(x_i)|}{n} \tag{3.1.2}$$

Cross Entropy Loss, or *Negative Log Likelihood*, is primarily used in classification tasks, penalizing confident but incorrect predictions. For non-binary classification across M classes, it is defined as:

$$NLL = -\sum_{c=1}^{M} p_{y_i,c} log(p_{f(x_i),c}) \tag{3.1.3}$$

Optimization hinges on **gradient-based** methods, with gradients offering insights on modifying inputs to achieve desired outputs [6]. Gradient Descent, a predominant gradient method, iteratively adjusts parameters opposite to the gradient direction.

$$\theta' = \theta - \epsilon \nabla_\theta L(\theta) \tag{3.1.4}$$

Various optimizers extend or draw inspiration from Gradient Descent, such as Stochastic Gradient Descent (SGD) and others like AdaGrad, RMSProp, AdaDelta, and Adam, each with unique parameter updating mechanisms.

The computationally intensive nature of analytical gradient computation led to the development of the **back-propagation** algorithm [14], which efficiently applies the chain rule for gradient calculation through a systematic operation order.

3.1.2 Generalization and Overfitting

A model's success hinges on its performance with new, unseen data, demonstrating its generalization capability. Good generalization requires low errors during both training and testing phases.

Models lacking generalization may exhibit underfitting or overfitting. Underfitting, indicated by high training error, suggests inadequate learning, while overfitting, marked by a significant train-test error disparity, reveals over-learned training data, including noise.

Training a neural network involves balancing train and test error optimization, with regularization strategies like L_1 and L_2 norms imposing penalties on complex models to favor simplicity. Dropout, another generalization technique, randomly omits layer outputs during training to enhance model robustness by introducing noise.

3.2 Embedding Overview

Throughout this dissertation, a key concept we'll frequently reference is *embeddings*. Defined as the transformation of high-dimensional vectors into a more compact, lower-dimensional space, embeddings aim to spatially co-locate semantically similar entities, thereby encapsulating the essence of the input data. This process of dimensionality reduction significantly simplifies machine learning tasks, especially when the original data is represented as sparse, high-dimensional vectors [5].

Particularly prevalent in the domain of natural language processing (NLP), embeddings serve as foundational elements for representing words or sentences. These are typically manifested as real-valued vectors that spatially arrange words with akin meanings closer together within the vector space [23]. The creation of such embeddings is facilitated through various techniques, including but not limited to, dimensionality reduction of word co-occurrence matrices, probabilistic models, and neural network methodologies. Noteworthy methods encompass Word2vec [10], GloVe [12], BERT [13], and Principal Component Analysis (PCA).

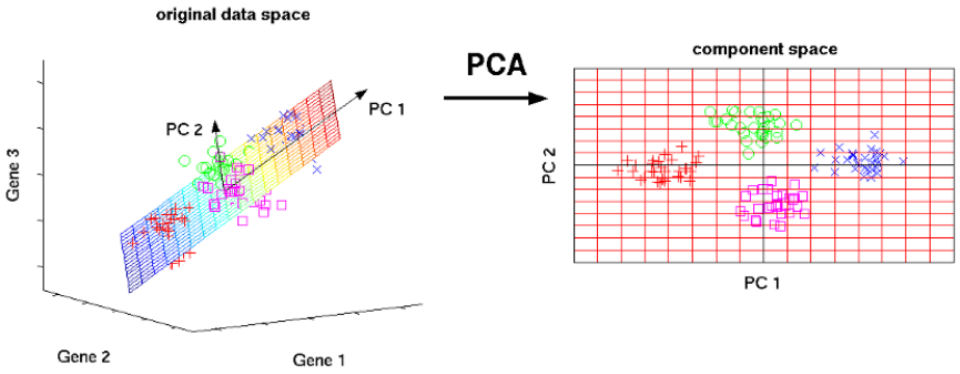

Figure 3.2.1: Embedding Visualization via PCA. [8]

Below, we delve into several prominent techniques for generating embeddings:

3.2.1 Principal Component Analysis

Employed for its dimensionality reduction capabilities, PCA identifies and merges highly correlated dimensions within the input data into singular dimensions. This method is characterized by an orthogonal linear transformation that reallocates data to a newly defined coordinate system, where the new axes are derived as eigenvectors from the covariance matrix's largest eigenvalues, ensuring these vectors are orthogonal to maximize data representation accuracy and independence.

Despite its straightforward application, PCA may not adequately capture non-linear data correlations, potentially resulting in the loss of pertinent information [20].

3.2.2 Autoencoder

Distinguished by its neural network foundation, the *Autoencoder* acts as a non-linear extension of PCA, capable of addressing PCA's inherent limitations. In its essence, an autoencoder could replicate any orthogonal basis in a manner that is not fixed. Basically. this architecture gives the input to a hidden layer (or layers) and lets the output be exactly the same as shape as the input. The goal would be to reproduce the input after multiplying the input with these hidden layers. So basically we compress the input and then decompress it. Or rather, we encode the input then decode it, hence the name autoencoder. Auto because it requires only the input to encode and decode it. And encoder is for the compression/encoding part.

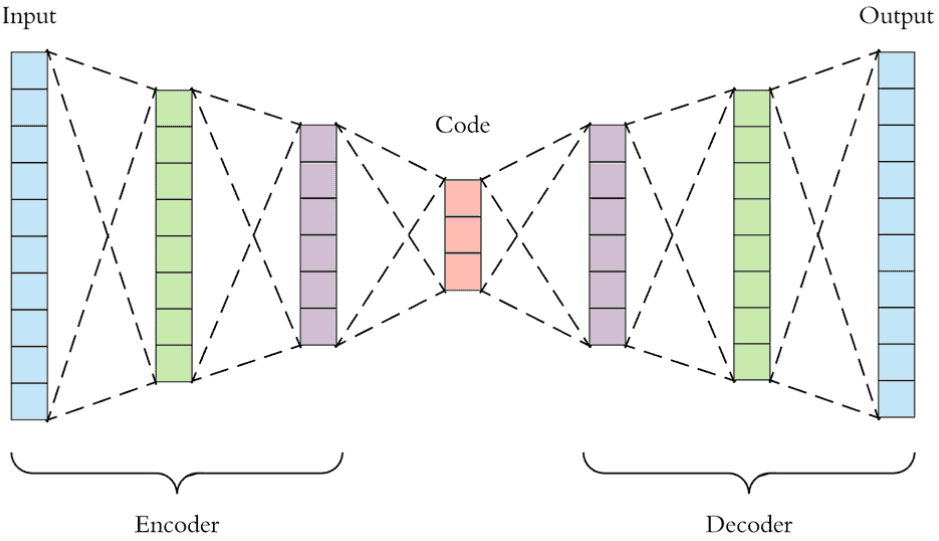

Figure 3.2.2: Autoencoder Structure.

Structured around a bottleneck design, an autoencoder comprises an *Encoder*, which condenses high-dimensional data into a dense, lower-dimensional representation, and a *Decoder* that aims to reconstruct the original data from this compressed form [7]. These components are optimized through a specialized loss function, the reconstruction loss, evaluating the fidelity of the decoded output to the original input.

Variants of the autoencoder architecture, such as undercomplete, sparse, contrastive, and variational autoencoders, introduce modifications primarily in the construction of their loss functions, thereby catering to a broad spectrum of embedding and data generation applications.

3.3 Graph Neural Networks

Given that the structure of graphs naturally emerges all around us, neural networks that operate directly on this type of data were invented. Graphs are non-Euclidean data and therefore GNNs can be

grouped in the broader category of Geometric Learning. Graph Neural Networks (GNNs) are known for their expressive power and have recently gained popularity due to their increasing capabilities in various applications such as recommendation systems and molecular fingerprinting. GNNs were created because most conventional Machine or Deep Learning algorithms are specifically designed to cover a certain type of data, such as images or text, but not graphs. Most data representations can be generalized to graphs, but the opposite is not true. In general, graphs are more complex, having a non-fixed number of unordered nodes within neighborhoods of variable size, and therefore existing models cannot handle them. Moreover, most common algorithms assume instance independence. This does not hold when performing tasks at the node level where a graph is the input of the neural network and the instances are its nodes. Finally, traditional Convolutional Neural Networks operate on images or more generally regular grids. The lack of locality in the traditional sense in graph data, their arbitrary size, and their invariance to permutations make it difficult to perform regular convolution.

Graph Neural Networks (GNNs) have emerged as a powerful class of neural networks for processing data represented in graph form. Graphs naturally represent a multitude of systems across various domains, such as social networks, biological networks, transportation networks, and knowledge graphs. GNNs leverage the structure of these graphs to perform node-level, edge-level, and graph-level predictions.

Understanding Graph Neural Networks (GNNs)

GNNs extend traditional neural network architectures to handle the irregular structure of graph data. They do this by focusing on the relationships and interactions among nodes (entities) and edges (relationships). The core idea behind GNNs is to learn a representation (embedding) for each node that captures not only its own attributes but also the information from its neighbors.

Message Passing Framework

The fundamental operation in GNNs is the message passing mechanism, where nodes aggregate information from their neighbors and update their own representations. This can be formalized as:

$$h_v^{(k+1)} = \text{UPDATE}^{(k)}\left(h_v^{(k)}, \text{AGGREGATE}^{(k)}\left(\{h_u^{(k)} : u \in \mathcal{N}(v)\}\right)\right)$$

where $h_v^{(k)}$ is the feature vector of node v at the k-th iteration, $\mathcal{N}(v)$ represents the neighbors of v, and UPDATE and AGGREGATE are functions that, respectively, update and aggregate neighbor information.

3.3.1 Graph Convolutional Networks (GCNs)

GCNs (Graph Convolutional Networks) are the gnns category that we will use during our experiments in this research. Before introducing them though in more detail, we should first mention ConvGNNs.
GCNs [16] are a specific subset of ConvGNNs (Convolutional Graph Neural Networks), exemplifying one of the pioneering approaches within the broader framework of graph convolutional techniques designed to analyze and interpret graph-structured data.

Convolutional Graph Neural Networks (ConvGNNs), drawing inspiration from traditional Convolutional Neural Networks (CNNs), utilize the graph convolution operation tailored for graph data. The general process followed by most of these models involves identifying a node's neighborhood, aggregating information from adjacent nodes, and ultimately forming a representation for each node. Similar to the way CNNs operate in conventional deep learning, ConvGNNs employ several layers of graph convolution to progressively refine node features, enhancing the level of detail captured as one progresses towards the output layer. ConvGNNs constitute critical components in the development of more sophisticated models. Furthermore, ConvGNNs are categorized into spatial or spectral types based on their approach to convolution: spatial ConvGNNs apply convolutions directly on the graph structure, considering the spatial relationships between nodes, whereas spectral ConvGNNs analyze graphs through the lens of signal processing in the frequency domain.

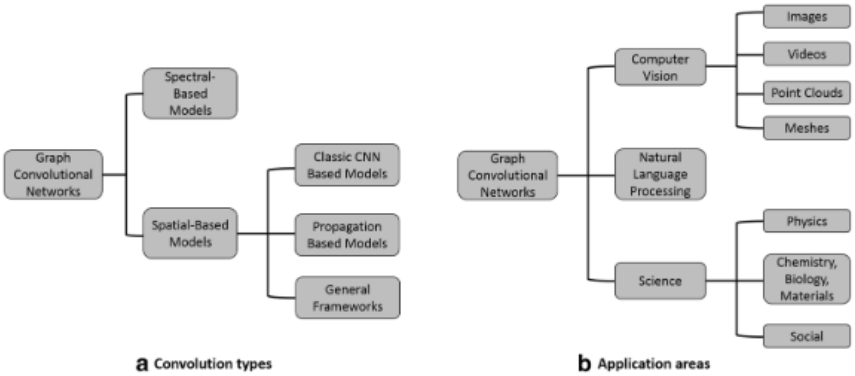

Figure 3.3.1: An overview of graph convolutional networks [16]

GCNs, introduced by Kipf & Welling, are among the most popular ConvGNNs. They simplify the convolution operation on graphs by using a first-order approximation of spectral graph convolutions. The basic architecture of Graph Convolutional Networks (GCNs) integrates node features and graph topology by aggregating neighbor information through a series of convolutional layers, allowing for the efficient learning of node embeddings. This process leverages a layer-wise propagation rule that combines local graph structure and node-level attributes, enabling tasks such as node classification and link prediction with improved accuracy.

The **Graph Convolutional Network (GCN)** introduces the idea of using a first-order approximation of ChebNet in order to mitigate overfitting. In practice, it assumes $K = 1$ and $\lambda_{max} = 2$. Following the same direction, the model imposes the constraint $\theta = \theta_0 = -\theta_1$. After enforcing these constraints, the convolution operation is:

$$x *_G g_\theta = \theta(I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x$$

Since it was empirically found that the term $I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ causes numerical instability, a *renormalization* trick was used. The term $D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = \bar{A}$ was replaced by $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} = \hat{A}$ where $\tilde{A} = I_n + \bar{A}$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. All of the above can be described with this compact equation:

$$H = X *_G g_\theta = f(\hat{A} X \Theta)$$

where f is an activation function, and multiple inputs and outputs are allowed due to the use of matrices.

GCN is a special case of spectral approach since it can also be interpreted as spatial. In the following equation, we can see how the aggregation of information within the neighborhood will occur. In this case, the node itself is also considered as its own neighbor, within one step:

$$h_v = f(\Theta^T(\sum_{u \in N(u) \cup v} \hat{A}_{v,u} x_u)) \quad \forall u \in V$$

This model is often used as part of more complex architectures in the literature due to its simplicity and good experimental performance.

3.3.2 GraphSAGE

GraphSAGE extends the idea of GCNs to efficiently generate embeddings by sampling and aggregating features from a node's local neighborhood. Unlike GCNs, which require the entire graph to be processed

at once, GraphSAGE learns a function to generate embeddings by sampling a fixed-size neighborhood and can thus be more scalable.

3.3.3 GAT (Graph Attention Networks)

GATs, proposed by Vaswani et al., introduce an attention mechanism into the aggregation step, allowing nodes to weigh their neighbors' contributions differently. This is particularly useful for graphs with highly variable node degree. Its goal is to learn the relative weights between neighboring nodes and thus differs from previous approaches like GCN and GraphSAGE because the concept of neighborhood is not predetermined or uniform.

The aggregation function is defined as:

$$h_v^{(k)} = \sigma\left(\sum_{u \in N(u) \cup v} \alpha_{vu}^{(k)} W^{(k)} h_u^{(k-1)}\right)$$

where the attention weights for each node v can be defined as:

$$\alpha_{vu}^{(k)} = \text{softmax}\left(\text{LeakyReLU}\left(a^T [W^{(k)} h_v^{(k-1)} \,||\, W^{(k)} h_u^{(k-1)}]\right)\right)$$

The variable a represents a set of learnable parameters. The representation of the hidden layers is initialized with the features of each node, and the softmax function ensures that the attention weights sum to one.

The above mechanism is called *self-attention*, but GAT additionally uses *multi-head attention* to stabilize learning and make the model more expressive. The exact equations are found in Velickovic et al.

GAT is efficient since the node-neighbor pairs can be computed concurrently. Additionally, the sizes of its neighborhoods are irrelevant, and it can be easily applied to inductive learning problems.

3.3.4 GNNs in the PyTorch Library

PyTorch, a popular deep learning framework, has an extension library called PyTorch Geometric (PyG) that provides efficient implementations of various GNN models, including GCNs, GATs, and GraphSAGE. PyG is optimized for both performance and usability, offering:

- Sparse GPU Acceleration: By leveraging sparse matrix operations, PyG efficiently handles large graphs on the GPU, significantly speeding up computation compared to dense matrix operations.

- Mini-Batch Processing: PyG supports mini-batch processing for GNNs, which is challenging due to the irregular structure of graph data. This is achieved through clever batching and unbatching techniques

that maintain the graph structure.

- Easy Model Definition: PyG abstracts away the complexity of defining convolutional layers on graphs, allowing researchers and practitioners to implement new GNN models with minimal code.

- Comprehensive Model Zoo: PyG comes with a wide range of pre-implemented GNN models and layers, making it easy to experiment with different architectures or benchmark new ideas against established models.

Finally, GNNs find applications across diverse domains, showcasing their versatility and efficacy. From social network analysis for community detection to drug discovery in bioinformatics, recommendation systems, and fraud detection, GNNs have proven instrumental in extracting meaningful insights from complex graph-structured data.

3.4 Transformers

In recent years, Transformers have revolutionized the field of Natural Language Processing (NLP) by achieving remarkable performance in various language-related tasks such as machine translation, text generation, and sentiment analysis. Originally introduced by Vaswani et al. in the seminal paper "Attention is All You Need" [17], Transformers have become the cornerstone of state-of-the-art NLP models, supplanting traditional recurrent and convolutional architectures. This section provides a comprehensive overview of Transformers, delving into their architecture, mechanisms, and recent advancements, with a focus on Swin Transformers.

3.4.1 Architecture of Transformers

Transformers are a class of deep learning models that have revolutionized the field of natural language processing (NLP) and beyond, due to their ability to efficiently handle long-range dependencies in data. The original Transformer model was introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017. Since then, Transformers have been adapted and expanded upon for various applications including computer vision, speech recognition, and more.

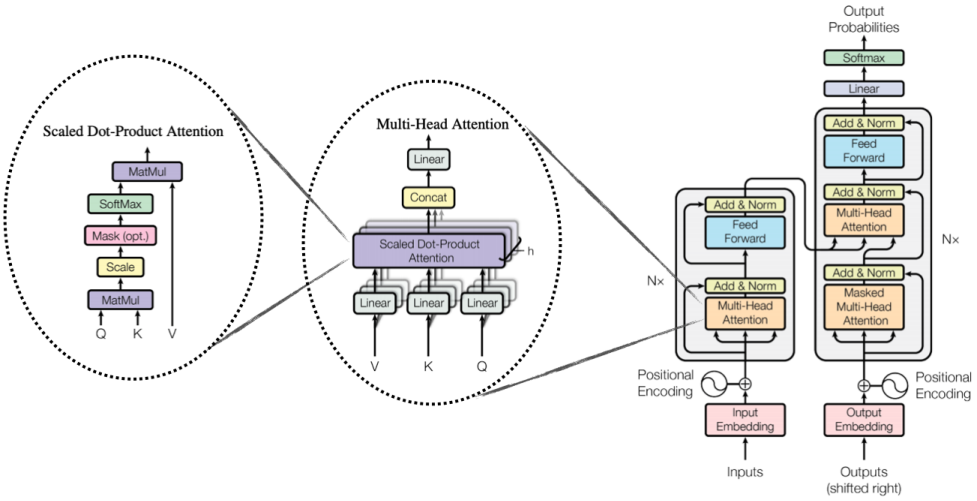

Figure 3.4.1: The Transformer - model architecture [17]

Core Components of Transformers

The Transformer architecture is based on the self-attention mechanism, which allows the model to weigh the importance of different parts of the input data differently. This is particularly useful in NLP, where the context of a word can significantly alter its meaning.

Self-Attention Mechanism

The self-attention mechanism in Transformers computes the attention scores for a sequence by considering how each element of the sequence (e.g., each word in a sentence) relates to every other element. This is achieved through a set of queries (Q), keys (K), and values (V), which are derived from the input embeddings through linear transformations. The attention scores are calculated using the following formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where d_k is the dimensionality of the keys.

Multi-Head Attention

To capture different aspects of the data, Transformers use multi-head attention, which runs several attention mechanisms in parallel. The output of each head is concatenated and linearly transformed into the desired dimensionality. This allows the model to focus on different positions and features within the input.

Positional Encoding

Since Transformers do not inherently process sequential data like RNNs or LSTMs, positional encodings are added to the input embeddings to provide some information about the position of each element in the sequence. These encodings can be fixed or learnable parameters.

Transformer Architecture

The Transformer model consists of an encoder and a decoder, each comprising multiple identical layers. Each layer in the encoder contains two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. In addition to these two sub-layers, each layer in the decoder includes a third sub-layer that performs multi-head attention over the encoder's output.

3.4.2 Training Transformers

Transformers are typically trained using the self-supervised learning paradigm, where the model learns to predict masked words in a sentence (Masked Language Model, MLM) or to generate the next word in a sequence (Next Sentence Prediction, NSP). The training objective is to minimize the cross-entropy loss between the predicted and actual words.

3.4.3 Vision Transformers (ViT)

The Vision Transformer (ViT) marks a significant departure from traditional convolutional neural networks (CNNs) in the field of computer vision, applying the transformer architecture, originally designed for natural language processing tasks, to image analysis. This innovative approach has demonstrated remarkable success, challenging the dominance of CNNs in various vision tasks. The core idea behind ViT is to treat an image as a sequence of fixed-size patches and apply the self-attention mechanism to capture global dependencies between these patches.

Architecture Overview

A ViT model begins by dividing an input image into a grid of non-overlapping patches. These patches are then flattened and linearly embedded into tokens with a dimension of D. A special token, known as the "class token" (CLS), is prepended to this sequence of embedded patches. Positional embeddings are added to the patch embeddings to retain positional information, which is crucial since the transformer architecture does not inherently understand the order of the input sequence. The resultant sequence of vectors serves as the input to the transformer encoder.

The transformer encoder consists of alternating layers of multi-head self-attention (MHSA) and multilayer perceptrons (MLP), with LayerNorm (LN) applied before each block, and residual connections applied after each block. The self-attention mechanism in MHSA allows the model to weigh the importance of different patches when processing each patch, enabling it to capture global interactions across the entire image.

Mathematically, the self-attention mechanism can be described as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V represent the queries, keys, and values matrices, respectively, derived from the input embeddings, and d_k is the dimension of the keys.

Each head in the MHSA layer independently computes self-attention, and their outputs are concatenated and linearly transformed. The MLP blocks contain two linear layers with a non-linearity (such as GELU) in between.

Despite these advantages, ViTs require large-scale datasets and significant computational resources for training from scratch. They are also more prone to overfitting on smaller datasets compared to CNNs. However, pre-trained ViT models can be fine-tuned on smaller datasets, mitigating this issue and enabling their application across a broad spectrum of vision tasks.

3.4.4 Swin Transformers

Swin Transformers, introduced by Liu et al. in the paper "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" [**liu2021swin**], extend the Transformer architecture to the domain of computer vision. Unlike traditional convolutional neural networks (CNNs) that operate on fixed-size local windows, Swin Transformers use a hierarchical architecture with shifted windows to capture long-range dependencies,reducing computational complexity compared to standard Transformers.

The key innovation in Swin Transformers lies in the use of shifted windows, which enable the model to process images at multiple scales while maintaining computational efficiency. By stacking multiple layers of shifted window-based self-attention mechanisms, Swin Transformers achieve state-of-the-art performance on various vision tasks, including image classification, object detection, and semantic segmentation.

Key Features of Swin Transformers:

- Shifted Window Partitioning: Swin Transformers divide the input image into non-overlapping windows and compute self-attention within each window. The windows are then shifted, and the process is repeated, allowing for cross-window connections and reducing computational cost.
- Hierarchical Representation: By gradually merging windows and reducing their number while increasing the dimensionality of representations, Swin Transformers build a hierarchical feature representation that is beneficial for various vision tasks.
- Applications: Swin Transformers have shown impressive performance on a range of tasks, including image classification, object detection, and semantic segmentation.

The Swin Transformer computes self-attention within local windows to efficiently process images. This local window self-attention mechanism can be formulated similarly to the standard self-attention in Transformers, but with a crucial modification to restrict attention computation within each window. Given a window of size $M \times M$, the self-attention within this window can be expressed as follows:

1. **Partitioning the Input**: The input image is divided into non-overlapping windows of size $M \times M$. Each window is treated as a sequence of tokens (pixels or patch embeddings).

2. **Computing Q, K, V**: For each window, we compute the queries Q, keys K, and values V matrices by applying linear transformations to the input representations within the window:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

where X is the input representation matrix for tokens within the window, and W_Q, W_K, and W_V are the parameter matrices for queries, keys, and values, respectively.

3. **Calculating Attention Scores**: The self-attention scores within a window are computed as the dot product of queries and keys, followed by a scaling factor and softmax normalization:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

where A is the attention matrix, and d_k is the dimensionality of the keys.

4. **Output Representation**: The output for each position in the window is then obtained by weighting the values V by the attention scores:

$$O = AV$$

where O represents the output matrix of the self-attention operation within the window.

This localized computation of self-attention allows Swin Transformers to efficiently handle high-resolution images by reducing the quadratic complexity with respect to the input size, which is typical in global self-attention mechanisms. Additionally, Swin Transformers employ a shifted windowing scheme in subsequent layers to enable cross-window connections and enhance the model's ability to capture long-range dependencies.

3.4.5 Advantages and Challenges

Transformers offer several advantages over traditional sequence models, including parallelization, scalability to longer sequences, and superior performance on various NLP tasks. However, they also pose challenges such as increased computational requirements and difficulty in capturing positional information. However one may wonder ... Why choose Swin Transformers over ViT in a research project on visual wod disambiguation? Well, choosing a Swin Transformer over a Vision Transformer (ViT) hinges on the **Swin Transformer's hierarchical processing of images, which captures multi-scale features more effectively**, and its computational efficiency through a shifted windowing scheme that reduces the complexity of self-attention computations. On the other hand, **ViTs require large-scale datasets and significant computational resources for training from scratch**. They are also more prone to overfitting on smaller datasets compared to CNNs. However, pre-trained ViT models can be fine-tuned on smaller datasets, mitigating this issue and enabling their application across a broad spectrum of vision tasks. This makes the Swin Transformer not only more scalable and adaptable to various vision tasks, including image classification, object detection, and semantic segmentation, but also more suited to handling large images and dense prediction tasks where understanding spatial hierarchies is crucial. Its design allows for easier integration into existing CNN architectures, offering a versatile and efficient solution for applications requiring detailed understanding of both local and global image features.

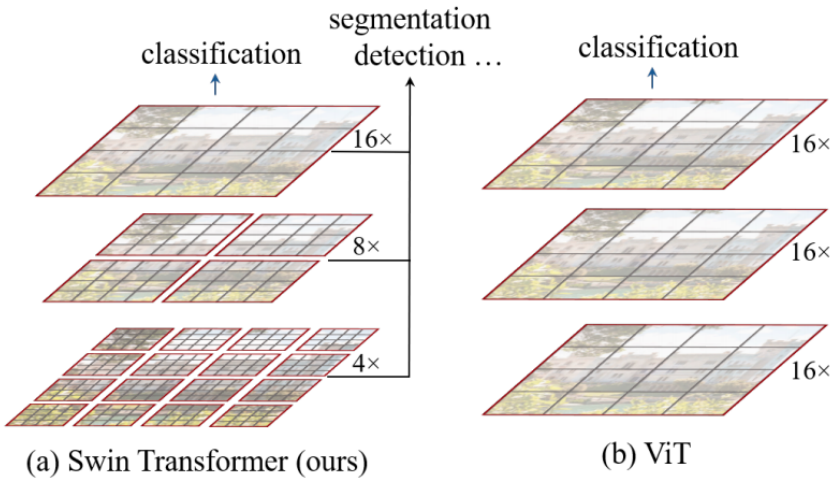

Figure 3.4.2: Swin Transformer vs ViT

3.5 MultiModal Models

Since we researched all possible methods and models that can be incorporated in Visual Word Disambiguation tasks, we had to experiment with using Multimodal models.
Multimodal models represent a significant advancement in the field of artificial intelligence by their ability to process and interpret multiple types of data simultaneously, most notably textual and visual information. These models are designed to understand the intricacies of language and the complexities of images or videos, enabling them to perform tasks that require a comprehensive understanding of both worlds.

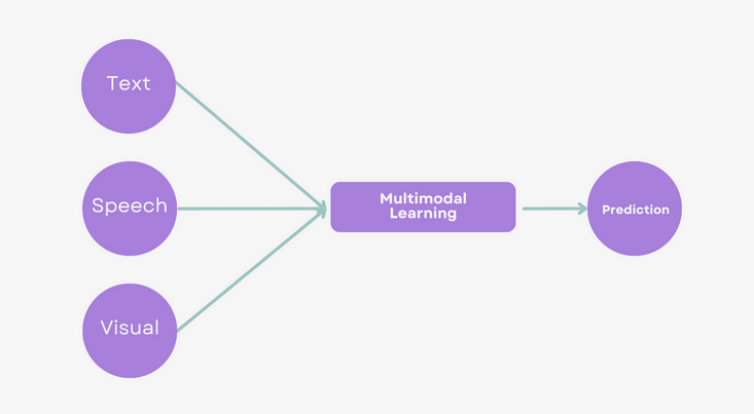

Figure 3.5.1: Multimodal Learning

3.5.1 Fundamentals of MultiModal Learning

The essence of multimodal learning lies in integrating data from different modalities - such as text, images, audio, and video - to improve the AI's understanding and decision-making capabilities. This approach allows the models to capture a richer representation of the real world, as humans do, by utilizing the complementary nature of different data types.

- Key Components of Multimodal Models

- Cross-Modal Understanding: Multimodal models are adept at correlating and translating concepts across different forms of data. For instance, they can associate the textual description of an object with its visual representation, enhancing tasks like image captioning or text-based image retrieval.
- Attention Mechanisms: These models often employ sophisticated attention mechanisms that allow them to focus on relevant parts of the data. For example, in a complex scene, the model can learn to pay more attention to the elements that are most relevant to the task at hand, whether they are in the text or the image.
- Fusion Strategies: An essential aspect of multimodal models is their ability to fuse information from different sources. This fusion can happen at various stages of processing, from early integration, where raw data from each modality is combined, to late integration, where high-level features or decisions are merged.

We can't talk about multimodals though without mentioning CLIP [2]. CLIP (Contrastive Language–Image Pre-training) builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning. The idea of zero-data learning dates back over a decade8 but until recently was mostly studied in computer vision as a way of generalizing to unseen object categories.9,10 A critical insight was to leverage natural language as a flexible prediction space to enable generalization and transfer. Essentially how CLIP works is that it pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a dog" and predict the class of the caption CLIP estimates best pairs with a given image.

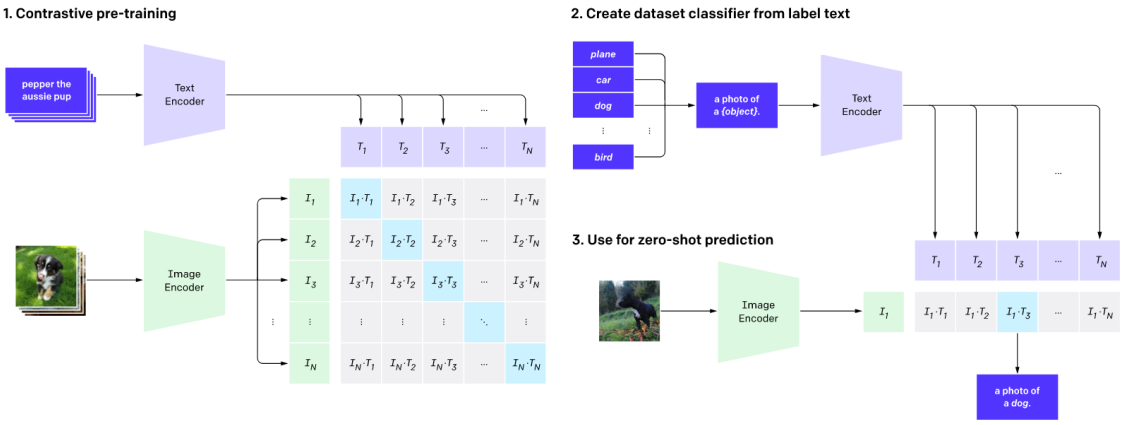

Figure 3.5.2: CLIP example

In the context of multimodal models, CLIP (sometimes referred as Vanilla CLIP) and Wiki-CLIP stand out as innovative approaches to bridging the gap between textual and visual data. These models leverage the strengths of neural networks to process and relate information across different modalities, enhancing the machine's understanding and capabilities in tasks that involve both images and text.

3.5.2 CLIP

CLIP, developed by OpenAI, is a groundbreaking model that has significantly impacted the field of artificial intelligence by demonstrating an exceptional ability to understand images in the context of natural language descriptions. Its architecture consists of two primary components:

- **Image Encoder**: This part of the model processes visual inputs, converting images into a high-dimensional vector space where similar visual content is represented by nearby points.
- **Text Encoder**: Parallel to the image encoder, the text encoder processes textual inputs, mapping sentences or phrases into the same high-dimensional space as the images.

The key innovation of CLIP lies in its training objective, which aligns the representations of images and texts in a shared embedding space. By using a contrastive learning approach, CLIP is trained on a vast dataset of image-text pairs, learning to associate images with their corresponding textual descriptions. This enables CLIP to perform a wide array of tasks, from zero-shot classification to complex reasoning tasks, by simply framing the tasks in the form of natural language queries.

3.5.3 Wiki-CLIP

Building upon the foundation laid by "Vanilla" CLIP, Wiki-CLIP represents an extension or variation of the original model that incorporates knowledge from Wikipedia or similar comprehensive text sources. This adaptation is aimed at enriching the model's understanding of concepts, entities, and contexts that are prevalent in human knowledge bases but might not be adequately covered in the datasets typically used for training standard CLIP models.

The integration of Wikipedia information can take various forms, such as:

- **Enhanced Text Encoder**: By incorporating Wikipedia articles or summaries into the training data, the text encoder of Wiki-CLIP can develop a deeper understanding of a broader range of concepts, including those that are less common in everyday language.
- **Knowledge-Augmented Training**: Wiki-CLIP might also involve modifying the training process to explicitly align visual representations with textual descriptions that are informed by Wikipedia knowledge, thereby enabling the model to make more informed associations between text and images.
- **Contextual Understanding**: The use of Wikipedia data helps the model grasp context better, allowing it to disambiguate between visually similar but conceptually different objects or scenes based on textual descriptions.

CLIP and Wiki-CLIP are at the forefront of multimodal AI research, showcasing the power of integrating visual and textual data. Their ability to understand and relate complex information across modalities opens up new horizons for AI applications, making technology more intuitive and accessible for users.

3.6 Visual Word Disambiguation

It's worth dedicating a section to the general sector of visual word disambiguation, since it's an ML branch that it's relatively new in the field of data science.
Visual word disambiguation is an advanced task within the intersection of computational linguistics and computer vision, aimed at resolving the ambiguity of words based on visual contexts. The task of Word Sense Disambiguation (WSD) consists of associating words in context with their most suitable entry in a pre-defined sense inventory. The de-facto sense inventory for English in WSD is WordNet. For example, given the word **mouse** and the following sentence:

"A mouse consists of an object held in one's hand, with one or more buttons."

we would assign "mouse" with its electronic device sense.

This task gains importance in the multimodal data domain, where textual and visual information coexists, such as in social media, online articles, and digital marketing. The core challenge lies in accurately determining the intended meaning of a word that has multiple senses, using associated visual cues alongside textual context. Word Sense Disambiguation (WSD) in general involves determining the correct meaning of a word based on its usage within a given context, typically referencing a specific sense from an established inventory. Recent advancements in this field (as noted by Bevilacqua et al[1]) have been largely driven by improvements in language models, though the application of WSD has predominantly been within purely textual domains. Traditionally, WSD relies on sense inventories derived from lexical databases like WordNet [11].
Visual word disambiguation extends beyond traditional text-based word sense disambiguation (WSD) by incorporating visual information to clarify ambiguities. Words with multiple meanings, known as polysemes, can be correctly interpreted when the textual context is supplemented with relevant visual information. For instance, the word "bat" could refer to a flying mammal or a sports equipment, and an accompanying image would immediately clarify the ambiguity.

3.6.1 Approaches and Methodologies

Efforts in visual word disambiguation involve a blend of techniques from both natural language processing (NLP) and computer vision:

- **Multimodal Embeddings**: A common approach is to generate embeddings that encapsulate features from both text and images. Techniques such as concatenation of pretrained word and image embeddings, cross-modal attention mechanisms, and joint embedding spaces are still explored.
- **Attention Mechanisms**: Many solutions incorporate attention-based models, like Transformers, to dynamically weigh the importance of different elements in the text and visual inputs, allowing

the model to focus on relevant features for disambiguation. In visual word disambiguation, attention mechanisms can dynamically prioritize the aspects of both text and image that are most helpful for resolving ambiguity.

– **Pretrained Multimodal Models**: The use of pretrained multimodal models such as CLIP (Contrastive Language–Image Pre-training) and VisualBERT, which are already capable of understanding associations between text and images.

– **Data Augmentation and Fusion Techniques**: To enrich the training data and enhance model robustness, participants employed data augmentation strategies for both text and images. Fusion techniques were also pivotal in effectively combining textual and visual features, including early fusion, late fusion, and hybrid approaches.

3.6.2 SemEval 2023 Task 1: Visual Word Disambiguation

The 2023 edition of SemEval (Semantic Evaluation) included a task specifically dedicated to visual word disambiguation, highlighting the growing interest and challenges in this area. Task 1 focused on leveraging multimodal data to resolve word sense ambiguities, inviting participants to develop models capable of integrating textual and visual cues effectively. This task actually triggered our thesis and the different experimentations.

- Task Overview

Participants were provided with datasets containing text-image pairs, where target words in the text were polysemous. The task was to predict the correct sense of these target words, guided by both the textual context and the associated images.

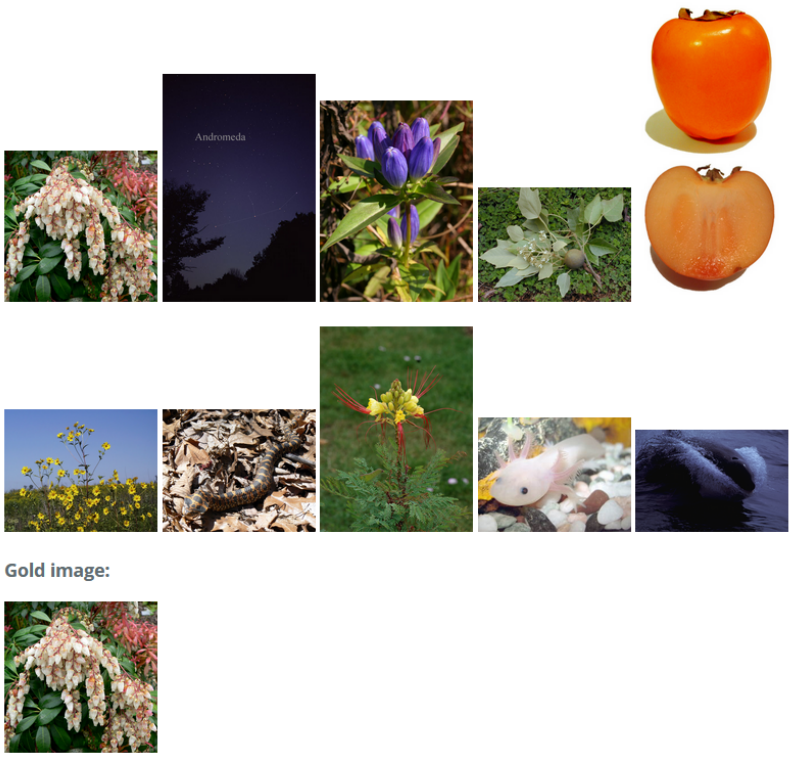

Figure 3.6.1: Given the full phrase andromeda tree containing the ambiguous target word andromeda, and the following ten candidate images, the task is to select the corresponding one. In this case, the correct image is the first one on the left, as shown above

- Highlights

- The task attracted a wide range of submissions, showcasing innovative approaches in multimodal learning and attention mechanisms.
- A common theme among successful submissions was the effective use of pretrained models, both in NLP (such as BERT and its variants) and computer vision (such as ResNet), and their adaptation to the multimodal disambiguation context.
- The task underscored the importance of robust fusion techniques and the challenge of dealing with diverse and sometimes noisy multimodal data.

3.6.3 Application of Advanced Architectures in Visual Word Disambiguation

The task of visual word disambiguation presents a unique set of challenges that stem from the need to accurately interpret the context in which a word is used, particularly when the word has multiple meanings. This complexity is magnified when dealing with visual data, as the interpretation must consider not only the textual cues but also the visual context surrounding the word. The application of state-of-the-art neural network architectures, such as Transformers, Graph Neural Networks (GNNs), and Large Language Models (LLMs) like BERT, offers promising solutions to these challenges. Each of these architectures brings a unique set of capabilities that are particularly suited to the task of visual word disambiguation, as outlined below.

- **Transformers**: This architecture utilizes self-attention mechanisms to weigh the influence of different parts of the input data, making it exceptionally well-suited for understanding the nuanced context of words within sentences. In the realm of visual word disambiguation, **Transformers can analyze the textual annotations or descriptions associated with images, effectively grasping the subtle contextual clues that hint at the specific meaning of ambiguous words** in a visual context.
- **Graph Neural Networks**: GNNs leverage the power of graph structures to model complex relationships and interactions between data points. This is particularly relevant in scenarios where the relationship between visual elements and textual descriptors is not linear or straightforward. GNNs can be used to construct a graph that represents the visual elements as nodes and their relationships as edges, incorporating textual annotations to provide context. By doing so, **GNNs facilitate a deep understanding of the interplay between visual elements and their textual descriptions, aiding in the accurate disambiguation of words based on visual context**.
- **Large Language Models**: LLMs such as BERT have demonstrated remarkable capabilities in capturing the intricacies of language, including syntax, semantics, and even some aspects of common sense knowledge. **These models are pretrained on vast amounts of text, enabling them to develop a deep understanding of language and context. When applied to visual word disambiguation, LLMs can analyze the textual content related to images, discerning the appropriate meaning of ambiguous words based on their context**.

The synergy of these advanced architectures—Transformers, GNNs, and LLMs like BERT—provides a robust framework for tackling the complexities of visual word disambiguation. Each architecture contributes a unique set of strengths: Transformers offer unparalleled efficiency in handling sequential data, GNNs excel in modeling complex relationships within data, and LLMs provide a deep understanding of language and context. When combined, these technologies enable the development of systems capable of accurately interpreting the meaning of words within a rich visual context, paving the way for significant advancements in the field of visual information processing and understanding.

Chapter 4

Strategies and Techniques in Visual Word Sense Disambiguation: A Multimodal Approach

In the realm of natural language processing and computer vision, Visual Word Sense Disambiguation (WSD) represents a fascinating intersection where the dual challenges of understanding textual and visual information converge. This chapter delves into the sophisticated array of strategies and techniques employed to navigate the complexities of Visual WSD, spotlighting the **multimodal approach** as a cornerstone for innovation in this field. At the heart of these efforts lie groundbreaking models such as **CLIP** (Contrastive Language–Image Pre-training), which have pioneered new pathways in bridging the semantic gap between visual content and textual descriptions. Moreover, we explore the pivotal role of **Large Language Models (LLMs)** in enhancing our ability to interpret and contextualize textual data within visual frameworks.

As the demand for nuanced understanding and interpretation grows, the importance of **advanced regularization techniques** and the strategic use of **embeddings** come to the fore. Regularization methods stand as critical tools in preventing overfitting, ensuring that models remain robust and generalizable across diverse datasets. Simultaneously, embeddings serve as the linchpin in translating high-dimensional data into a more accessible, lower-dimensional space, enabling models to grasp the subtle nuances that differentiate meanings in varied contexts. These embeddings, whether derived from text or images, are instrumental in capturing the essence of the data, facilitating a deeper semantic understanding that is essential for accurate word sense disambiguation.

Throughout this chapter, we will dissect these methodologies, examining how they contribute individually and synergistically to tackle Visual WSD tasks. By weaving together insights from the latest research and practical applications, we aim to illuminate the intricate tapestry of techniques that underpin current successes and future advancements in the field. As we navigate through the discussion on CLIP, LLMs, regularization, and embeddings, our objective is to provide a comprehensive overview that not only enriches the reader's understanding but also inspires further exploration and innovation in Visual Word Sense Disambiguation.

4.1 Data Preprocessing - Regularization

Normalization and regularization are crucial processes in the preparation and training of machine learning models, including those designed for Visual Word Sense Disambiguation (VWSD). These techniques enhance the model's ability to generalize from the training data to unseen data, thereby improving its performance on real-world tasks. In this context, let's explore the normalization/regularization techniques within the VWSD domain.

Normalization involves scaling input features to a standard range or distribution. This is vital in VWSD because the model processes heterogeneous data types, such as images and text, which may have different value ranges or distributions. By normalizing the features, we ensure that no single type of input disproportionately influences the model's learning process. Below we list the steps we implemented to regularize our data.

- Retrieve possible meanings (synsets) for a given word, which can be used for understanding different meanings in the context. This is already implemented in Python by importing wordnet from the nltk.corpus.
- Generate lexical substitutions by replacing a focus word with synonyms from its synsets, potentially increasing the robustness of the model by exposing it to varied lexical contexts.

```
    def expand_sentence_contexts(focus_word, base_sentence) -> tuple:
        word_synsets = wn.synsets(focus_word)
        if not word_synsets:
            return 1, [base_sentence]
        all_expanded_sentences = [base_sentence]
        count_synonyms = [len(synset.lemma_names()) for synset in word_synsets]
        count_hypernyms = [[len(hypernym.lemma_names()) for hypernym in (synset.
    hypernyms() + synset.instance_hypernyms())] for synset in word_synsets]
        max_count_synonyms = max(count_synonyms)
        max_count_hypernyms = 0
        total_max_synset = max(max_count_synonyms, max_count_hypernyms)
        for idx, synset in enumerate(word_synsets):
            variations_for_context = [base_sentence.replace(focus_word, synonym.
    replace('_', ' ')) for synonym in synset.lemma_names()] * math.ceil(
    total_max_synset / count_synonyms[idx])
            variations_for_context = variations_for_context[:total_max_synset]
            assert len(variations_for_context) == total_max_synset
            all_expanded_sentences.extend(variations_for_context)
        return len(all_expanded_sentences), all_expanded_sentences


```

Listing 4.1: How to generate synsets

- Calculate the dot product similarity between image and text embeddings, a measure that could be utilized to align multimodal features.

```
    def dot_prod_sim(img_e, txt_e) -> torch.Tensor:
      dot_similarity = (img_e @ txt_e).T
      return dot_similarity

```

Listing 4.2: Python example of dot product similarity

– Finally, an essential component in the preprocessing pipeline for a Visual Word Sense Disambiguation (VWSD) system, is our custom **SwinImageProcessor** (especially when employing models like Swin Transformer that are sensitive to input specifications). This custom class, SwinImageProcessor, which we will denote as I, is designed to standardize the input images, ensuring they are in the correct format and distribution for optimal model performance.

When the processor is called, it checks if the input is a list of images, allowing batch processing, and then individually processes each image. The processing involves resizing the image to the predetermined size using bilinear interpolation, a method that helps in retaining the image quality. Let's denote the resizing operation as $R(I, s)$, where s is the target size.

The bilinear interpolation function for resizing can be represented as:

$$R(I, s) = I_{resized}(x', y') = \sum_{i,j} I(i, j) \cdot r(x', i) \cdot r(y', j)$$

where (x', y') are the coordinates in the resized image, (i, j) are the coordinates in the original image, and $r(\cdot)$ is the bilinear interpolation kernel.

After resizing, the image is normalized by adjusting its pixel values to have a specified mean and standard deviation, which is a common practice to remove the effects of lighting variations and contrast differences in the dataset. This adjusts the pixel values of the image such that the resulting pixel intensity values have a specified mean and standard deviation. If mean is the vector of mean values for each channel and std is the vector of standard deviations, the normalization operation $N(I)$ can be mathematically described as:

$$N(I) = \frac{I - \text{mean}}{\text{std}}$$

where the subtraction and division are element-wise operations applied per channel. This operation ensures that the pixel value distribution of the image closely aligns with that of the dataset on which the Swin Transformer model was originally trained. Combining these two operations, the output of the 'SwinImageProcessor' for an image I can be represented as:

$$O(I) = N(R(I, s))$$

This sequence of preprocessing operations ensures that the images fed into the Swin Transformer model are of a consistent size and value distribution, which is critical for the model to learn effectively and generalize well to new, unseen data.The SwinImageProcessor thus serves as a critical step in preparing images for feature extraction by Swin Transformer models, aligning the visual data with the expected input structure and scaling. This tailored processing is vital to ensure that the complex interplay between text and image modalities in VWSD tasks is accurately captured by the multimodal learning system.

```
class SwinImageProcessor():
    def __init__(self, size=224, mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]):
        super().__init__()
        self.mean = mean
        self.std = std

    def __call__(self, images, return_tensors=None):
        if not isinstance(images, list):
            images = [images]

        processed_images = []
        for image in images:
            # Your image processing code here, e.g., resizing, normalization, etc.
            processed_image = self.process_image(image)
            processed_images.append(processed_image)

        return processed_images

    def process_image(self, image):
        # Example: Resize the image to the specified size and normalize
        image = image.resize((self.max_size, self.max_size), Image.BILINEAR)
        image = self.normalize(image)

        return image

    def normalize(self, image):
        # Example: Normalize the image
        image = (image - self.mean) / self.std
        return image
```

Listing 4.3: Swin Image Processor (instead of using AutoImageProcessor)

After applying these practices to our input data we can then apply our model of choice to train on the input data and use any of the methods mentioned in the following sections.

4.2 Multimodal Learning Architectures

A commonly spotted methodology was the integration of pre-built models from Hugging Face, a leading repository of state-of-the-art machine learning models. Specifically, two models: **the Vision-and-Language Transformer (ViLT) and the Vision Text Dual Encoder**. These models are designed to navigate the complex interplay between visual and linguistic information, encoding them into a unified representation that facilitates a deeper understanding of multimodal content.

4.2.1 Vision-and-Language Transformer (ViLT)

The Vision-and-Language Transformer (ViLT) stands out as a pivotal model in our exploration. ViLT is engineered to process both visual and textual inputs simultaneously, leveraging the transformer architecture renowned for its effectiveness in capturing long-range dependencies. At its core, ViLT utilizes the Transformer model, a deep learning mechanism known for its efficiency in handling sequential data, which has been remarkably successful in natural language processing (NLP). By design, ViLT treats images and text on equal footing, directly embedding pixel values and textual tokens without relying on region proposals or object detection pre-processing steps. Unlike previous vision-language models that often rely on heavy pre-processing of images through object detection models, ViLT directly inputs image pixels and text into a single Transformer model. This approach simplifies the architecture and reduces computational costs. The model is trained on tasks that require understanding the interplay between text and images, such as image-caption matching, visual question answering, and image-text retrieval. By doing so, ViLT learns to embed both visual and textual inputs into a shared representation space, enabling it to perform tasks that require comprehension of both modalities. This direct and efficient method allows ViLT to achieve competitive performance on various benchmarks with less computational overhead compared to its predecessors.

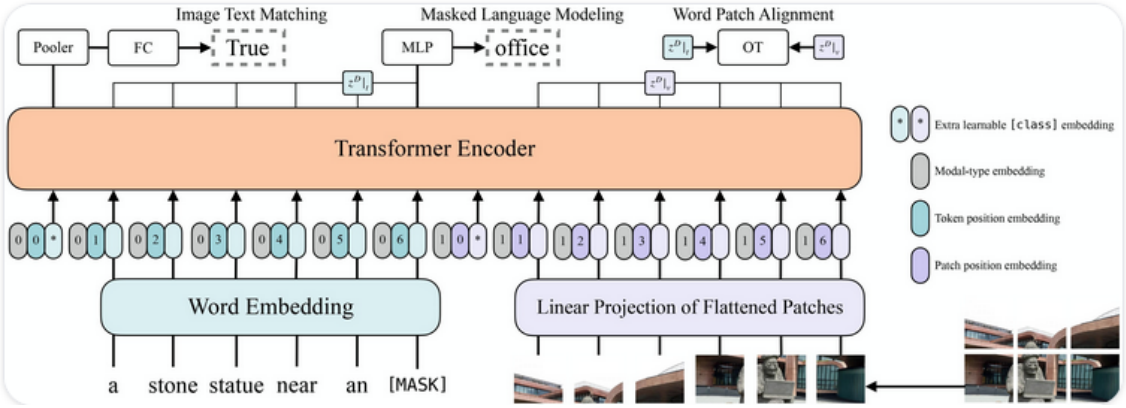

Figure 4.2.1: ViLT architecture

4.2.2 Vision Text Dual Encoder

Parallel to our exploration with ViLT, we experimented with **the Vision Text Dual Encoder**, a model characterized by its dual-encoder architecture. This innovative model projects visual and textual inputs into a shared 512-dimensional vector space, facilitating a direct comparison between the two modalities. For the visual component, we utilized the Vision Transformer (ViT) to encode images. ViT applies the principles of the transformer architecture directly to patches of the image, allowing it to capture rich, hierarchical visual information. For textual encoding, we adopted BERT (Bidirectional Encoder Representations from Transformers), specifically its uncased variant, to process textual input. BERT's deep understanding of language nuances and context made it an ideal candidate for generating robust textual embeddings.

```
1
2 processor = VisionTextDualEncoderProcessor(image_processor, tokenizer)
3 model = VisionTextDualEncoderModel.from_vision_text_pretrained(
4     "google/vit-base-patch16-224", "bert-base-uncased"
5 )
```

Listing 4.4: Vision Text Dual Encoder from Hugging Face

By combining ViLT's holistic approach to multimodal data processing with the precision of the Vision Text Dual Encoder's separate but synergistic encodings, we aimed to achieve a more nuanced understanding of the interrelations between text and images. This involved fine-tuning these models on our dataset, adjusting parameters, and experimenting with various encoding strategies to optimize performance.

The adoption of these two ready-made Hugging Face multimodal models, reveals inherent limitations in their architectures that can hinder optimal performance in certain tasks. Both models integrate separate encodings for vision and text into a unified 512- dimensional vector space, employing Vision Transformer (ViT) for image processing and Uncased BERT for textual analysis. While this design facilitates the handling of multimodal inputs, it inherently assumes a linear and direct correspondence between the visual and textual domains. This assumption can be overly simplistic for complex tasks requiring nuanced understanding and integration of multimodal information. For instance, the ViLT's reliance on transformer architectures without region-specific feature extraction can lead to inadequate spatial reasoning, crucial for tasks like object detection or spatial layout comprehension. Similarly, the Vision Text Dual Encoder's approach, though innovative in merging vision and text encodings, might struggle with capturing the depth of semantic relationships between text and image elements due to the fixed dimensionality and independent processing paths.

4.2.3 LiT : Zero-Shot Transfer with Locked-image text Tuning

LiT [24], short for Locked-image text Tuning, represents a significant advancement in the field of machine learning, particularly in tasks that involve both visual and textual data. This approach introduces an innovative technique for zero-shot transfer learning, where a model trained on one task is applied to a different, unseen task without any additional fine-tuning. The essence of LiT lies in its unique method of leveraging the pre-trained capabilities of large-scale image and text models, essentially "locking" the image representation while tuning the text components to achieve a harmonious understanding between the two modalities.

In the traditional transfer learning paradigm, models often undergo extensive re-training or fine-tuning on datasets specific to the target task. However, LiT challenges this norm by keeping the visual representation fixed and only adjusting the textual part of the model. This methodology is grounded in the insight that high-quality image representations, once learned, possess a versatile and robust understanding of visual content that can generalize across different tasks. By focusing on the textual aspect, LiT efficiently adapts to new tasks, leveraging the inherent strength of the image representations to comprehend and interpret visual data in the context of the new textual information it encounters.

The concept of "Locked-image Tuning" (LiT) emerges from the broader context of training models to understand and integrate visual and textual information. This training strategy is a subset of contrastive tuning methods specifically tailored for image-text data. In these methods, the goal is to fine-tune a pre-trained model so that it can effectively match or relate images with their corresponding textual descriptions, or vice versa, in a shared representation space. The "two letters" notation is a concise way to describe the setup of the model's components during this fine-tuning phase, particularly focusing on how the image and text "towers" (or processing streams) are initialized and whether they are allowed to update during training.

- **L** stands for "locked," indicating that these variables (parameters) are kept fixed during the training process. They are initialized from a pre-trained model and do not change, meaning the model retains the knowledge it acquired during pre-training without further adaptation in these areas.
- **U** signifies "unlocked and initialized from a pre-trained model," meaning these parts of the model are both initialized with weights from a pre-trained model and allowed to update during the training process. This allows for fine-tuning and adaptation based on the new task-specific data.
- **u** (lowercase) indicates "unlocked and randomly initialized," denoting that these model components start without pre-trained weights and are fully trained from scratch on the task-specific data, learning entirely new representations.

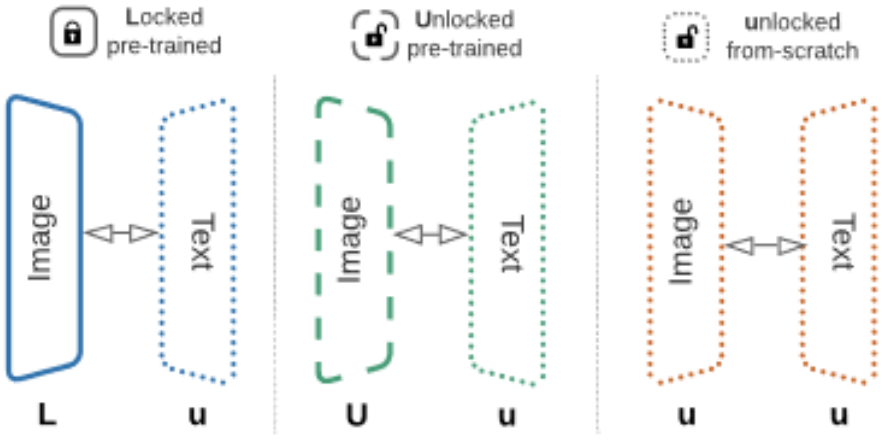

Figure 4.2.2: LiT architecture

In the context of LiT, the notation **Lu** specifies a particular configuration where the image tower is "locked" (L) and the text tower is "unlocked and randomly initialized" (u). This setup implies that the image processing component of the model does not adjust its parameters during fine-tuning, relying entirely on the knowledge it gained during pre-training. In contrast, the text-processing component starts without any pre-learned knowledge and learns from the ground up how to relate to the fixed representations provided by the image tower.

This design choice for LiT aims to exploit the rich, pre-trained visual representations while allowing the model to develop a tailored understanding of how text can be matched or associated with these images. By keeping the image representations fixed, LiT focuses the model's capacity on learning the textual nuances and their associations with the visual content, enabling effective zero-shot transfer capabilities. Zero-shot transfer refers to the model's ability to apply what it has learned to new tasks or datasets it was not explicitly trained on, a valuable property for generalizing across diverse visual and textual domains.

4.3 Contrastive Language–Image Pre-training (CLIP)

The advent of CLIP [2] (Contrastive Language–Image Pre-training) by OpenAI has ushered in a novel paradigm for tackling visual word sense disambiguation (WSD) tasks, leveraging its groundbreaking capability to understand images in the context of natural language descriptions. CLIP, by training on a diverse range of internet- collected images and text, learns to associate words and phrases with visual concepts, making it uniquely suited for visual WSD, where the goal is to resolve ambiguity in visual content using textual cues. This dual understanding allows CLIP to excel in tasks requiring nuanced interpretation of both visual and linguistic elements, setting a new standard for performance in visual WSD tasks.

After researching the published papers from SemEval 2023, the most common applications of CLIP was through (Vanilla) CLIP and Wiki enhanced CLIP.

4.3.1 CLIP

The core of this approach leverages the CLIP (Contrastive Language–Image Pre-training) model, which operates on the premise of aligning text and image embeddings in a shared high-dimensional space. The model's objective function during pre-training is designed to maximize the cosine similarity between correct image-text pairs while minimizing it for mismatched pairs. Formally, for a given text T and image I, the similarity score S computed by CLIP can be represented as:

$$S(T, I) = \frac{E_T(T) \cdot E_I(I)}{\|E_T(T)\|\|E_I(I)\|}$$

where $E_T(T)$ and $E_I(I)$ denote the text and image embeddings produced by CLIP, respectively, and $\cdot$ represents the dot product between these embeddings.

CLIP, as seen above, operates by simultaneously training two separate encoders: one for images and another for text. The goal during this pre-training phase is to enable these encoders to accurately predict the pairing of images and texts as found in the dataset. This capability forms the basis of CLIP's application as a zero-shot classifier, a model that can classify images it has never seen during training into categories.

To achieve zero-shot classification, CLIP utilizes a clever approach that involves translating the classes of a dataset into descriptive captions. For instance, if the dataset contains images of dogs among other objects, one such caption could be "a photo of a dog". CLIP then evaluates these captions against a given image to determine which caption, or class, best matches the image according to the model's prediction.

This process involves generating text embeddings for each caption and an image embedding for the input image. CLIP then compares these embeddings to calculate which caption's embedding is most

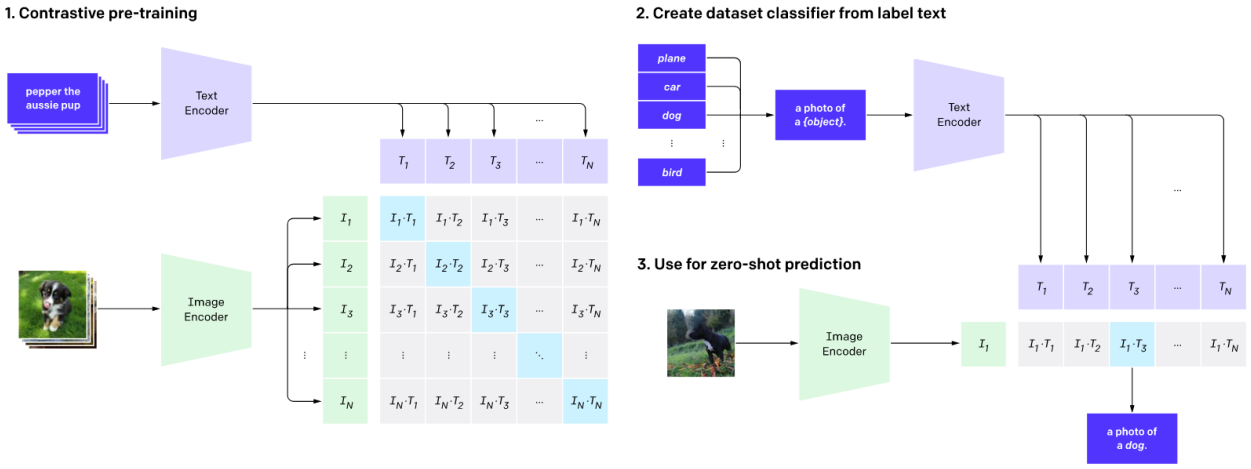

Figure 4.3.1: CLIP example

similar to the image's embedding. The class of the caption that has the highest similarity score with the image is predicted to be the class of the image. This method allows CLIP to classify images into categories it was not explicitly trained to recognize, showcasing its flexibility and power as a zero-shot classifier.

4.3.2 Wikipedia-Enhanced CLIP

Recognizing the limitations of minimal context in accurately disambiguating words, we explored the augmentation of textual descriptions using Wikipedia, the largest and most comprehensive online encyclopedia. This augmentation strategy involves enriching the textual context of each target word with relevant information extracted from Wikipedia, aiming to provide a broader semantic foundation for the CLIP model's decision-making process.

```
openai.api_key = "API_KEY"
tokenizer = CLIPTokenizer.from_pretrained("openai/clip-vit-base-patch32")
wiki_wiki = wikipediaapi.Wikipedia('en')
page1 = wiki_wiki.page(sentence)
```

Listing 4.5: Wikipedia-Enhanced CLIP

To enhance the textual context provided to CLIP, we retrieve additional information from Wikipedia. The process involves generating an augmented textual description T' for each target word by appending relevant Wikipedia summary content to the original text T. This enriched context aims to provide a comprehensive semantic understanding that aids in the disambiguation process.

The Wiki-enhanced CLIP approach proceeds in two steps. First, for each target word, a relevant Wikipedia summary or article excerpt is retrieved, focusing on content that closely relates to the word's context within the V-WSD task. This enriched textual description, embodying a more detailed exposition of the word and its nuances, is then presented to CLIP along with the candidate images. By computing similarity scores in this enriched embedding space, CLIP can leverage the additional contextual clues to make more informed and accurate disambiguation decisions.

4.3.3 CLIP observations

CLIP models, by leveraging a broad spectrum of visual concepts learned directly through natural language, exhibit a level of flexibility and generality that surpasses traditional ImageNet models. Their ability to perform a variety of tasks without specific training—known as zero-shot capability— has been demonstrated across more than 30 distinct datasets. These tasks range from fine-grained object classification and geo-localization to action recognition in videos and Optical Character Recognition

(OCR). Although CLIP's performance on zero-shot OCR tasks shows variability, its semantic OCR representations prove to be highly valuable. For instance, when the SST-2 NLP dataset is converted into image form, a linear classifier using CLIP's embeddings performs comparably to a Continuous Bag of Words (CBoW) model processing the actual text. Moreover, CLIP shows promise in identifying hateful memes without relying on the original text, a task where OCR capabilities play a critical role and illustrate a behavior not found in standard ImageNet-based models. This document includes visualizations of unbiased, randomly selected predictions from each zero-shot classifier to illustrate these findings.

Further evidence of CLIP's superior performance comes from a conventional evaluation of representation learning, utilizing linear probes. Here, the highest-performing CLIP model outshines the leading public ImageNet model, Noisy Student EfficientNet-L2, in 20 out of 26 different transfer datasets tested, showcasing its robustness and wide applicability across diverse visual understanding tasks.

While CLIP is generally adept at recognizing common objects, it encounters difficulties with more abstract or intricate tasks, such as enumerating objects within an image or estimating the proximity of the nearest vehicle in a photograph. In these scenarios, CLIP's zero-shot capabilities only marginally outperform random chance. Similarly, CLIP **faces challenges in highly specific classification tasks, like distinguishing between various car models, aircraft types, or flower species**, where it lags behind models tailored to these particular tasks.

Furthermore, CLIP's ability to generalize to images outside its pre-training scope is limited. Additionally, it's worth noting that CLIP's zero-shot classifiers exhibit sensitivity to the specific wording or phrasing used, often necessitating a process of trial and improvement known as "prompt engineering" to achieve optimal results.

Chapter 5

Introduction to ArPa Model

The ArPa architecture, as seen visualised in the diagram below, integrates Bert (or DistilBert for faster and more efficient results) and Swin Transformer models before funneling the processed data into a Graph Neural Network (GNN), forming a comprehensive pipeline particularly suited for tasks like Visual Word Sense Disambiguation (WSD). This pipeline is ingeniously designed to harness the strengths of each model to process textual and visual inputs separately, then combine their features to exploit the relational context between different elements (words and visual cues) in a graph structure. Let's delve into how each component contributes to handling the complexities of visual wsd.

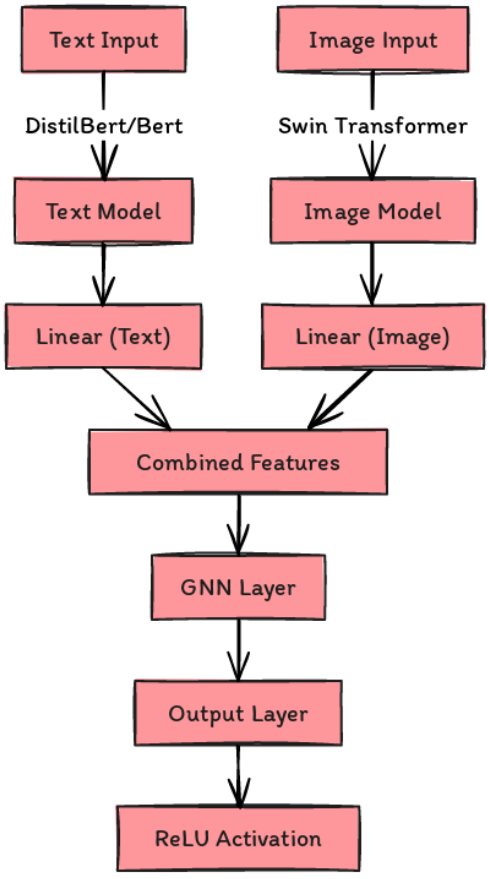

Figure 5.0.1: The ArPa model - Generic Diagram of architectural pipeline

5.1 Architectural Pipeline of ArPa

5.1.1 Custom GNN layer

In the realm of graph neural networks (GNNs), creating custom layers tailored to specific tasks can significantly enhance the model's ability to learn complex patterns and relationships within graph-structured data. The code snippet below outlines the definition of such a custom GNN layer, which intriguingly combines node features from different modalities (text and images in our case) with the graph's structural information. Let's dissect this custom GNN layer to understand the unique role each component plays in processing graph data.

- Initialization of the CustomGNNLayer

- **GCNConv Layer**: At the heart lies the GCNConv (Graph Convolutional Network Convolution) layer, a pivotal component that embodies the essence of graph convolutions. By accepting an 'input_dim' parameter (the size of the node features) and a 'hidden_dim' parameter (the size of the features after transformation), this layer acts as a sophisticated sieve, distilling complex node features into a more refined representation.
- **ReLU Activation Function**: Following the GCNConv layer, we encounter the ReLU (Rectified Linear Unit) activation function. This function is like the gatekeeper of neural networks, allowing only positive values to pass through unaltered while setting all negative values to zero. It introduces non-linearity into the model, enabling it to learn and represent more complex relationships between the nodes in the graph.

- The Forward Pass

- **Preparing the Edge Index**: The method begins by transforming the adjacency matrix into an edge index, a compact representation required by the GCN layer. This step is akin to mapping the roads between cities on a map, delineating how each node (city) is connected to others.
- **Reshaping Node Features**: Before diving into the graph convolution, the node features ('x') are reshaped to ensure they align with the expected input format.
- **Graph Convolution**: With everything in place, the node features and the edge index are fed into the GCN layer. This is where the alchemy happens—node features are filtered through the lens of the graph's structure, melding the information from a node's neighbors with its own features to produce a new, enriched representation.
- **Activation**: Finally, the ReLU activation function is applied to the output of the GCN layer, ensuring the model captures non-linear patterns and relationships within the data.

```
1
2  class CustomGNNLayer(nn.Module):
3      def __init__(self, input_dim, hidden_dim):
4          super(CustomGNNLayer, self).__init__()
5
6          # GCN layer
7          self.gcn = GCNConv(input_dim, hidden_dim)
8          self.relu = nn.ReLU()
9
10     def forward(self, x, adjacency_matrix):
11         """
12         :param x: Node features (concatenated text and image features)
13         :param adjacency_matrix: Graph adjacency matrix
14         :return: GNN output
15         """
16         # Assuming adjacency_matrix is in COO format
17         row, col = adjacency_matrix.nonzero(as_tuple=True)
18         edge_index = torch.stack([row, col], dim=0)
19
20         # Ensure the node features have the correct shape
21         x = x.view(-1, x.size(-1))
22
23         # Pass features through GCN layer
```

```
        x = self.gcn(x, edge_index)

        x = self.relu(x)
        return x
```

Listing 5.1: GNN layer of ArPa

The result of the forward pass through this custom GNN layer is a set of updated node features, which have been transformed by considering both the intrinsic characteristics of the nodes and the structural information of the graph.**In the ArPa architecture, Graph Neural Networks (GNNs) play a crucial role by exploiting the relational information inherent in the data, crucial for tasks such as Visual Word Sense Disambiguation (WSD), but it will become more evident in the following sections.**

5.1.2 ArPa Model Architectural Pipeline Specifics

The ArPa model, utilizes a multi-stage processing pipeline to handle multimodal inputs (text and images) leveraging different types of architectural layers to produce a comprehensive understanding of the data. Here's an overview of each component and its purpose:

- **DistilBert Layer**: This layer processes textual input using a DistilBert model, which is a streamlined version of the BERT model, retaining most of its original performance while being smaller and faster. The DistilBert layer is responsible for extracting contextualized text embeddings that capture the semantics and syntactic relationships within the textual data. *The user can easily exchange DistilBert for Bert as DistilBERT is a smaller and faster version of BERT, designed to be more efficient.*
- **Swin Transformer Layer**: For visual inputs, the Swin Transformer model is utilized. Swin Transformers are designed to compute image representations through a hierarchical vision transformer architecture that allows for modeling at various scales and is effective at capturing both local and global visual features.
- **Linear Layers (fc_text and fc_image)**: After the initial feature extraction from text and images, these features are processed by separate linear layers which transform the high-dimensional embeddings into a lower-dimensional space, making the data more manageable and preparing it for fusion. The linear layers also serve to align the feature dimensions, facilitating effective combination.
- **Feature Combination**: The outputs of the linear layers(fc_text and fc_image) are then concatenated to form a single, unified representation that encapsulates both textual and visual information, ensuring that the model has a holistic view of the multimodal data.
- **GNN Layer**: The concatenated features are fed into a custom Graph Neural Network (GNN) layer. The GNN excels at modeling the relationships and interactions between the different elements represented in the features, effectively learning how the context of a word and its visual surroundings interact to influence meaning. This step is pivotal in WSD as it mirrors how humans use contextual cues for interpretation.
- **Output Layer**: Finally, the processed and enriched feature set is passed through an output linear layer, which is tasked with making the final predictions. In the context of WSD, this layer would be responsible for classifying the sense of a word based on the learned multimodal embeddings.

The combination of these layers allows the ArPa model to perform sophisticated WSD by considering the rich and complex interdependencies between text and images. The architecture is designed to be both deep, to capture multiple levels of abstraction, and wide, to integrate different types of data, which is ideal for handling the intricacies of real-world multimodal scenarios.

```
Class ArPa
    Initialize(gnn_hidden_dim)
        super(ArPa, self).initialize()

        text_model <- Load BertModel
        #NOTE: For efficiency reasons we can
        #      use DistilBert Model which is a smaller
        #      and faster version of BERT, designed to
        #      be more efficient

        image_model <- Load SwinModel
        #NOTE: Also can be substitued by a different kind
        #      of vision transformer. But due to Swin Transformers
        #      hierarchical process of visual data, we thought it
        #      would be a better fit.

        fc_text <- Linear layer (size_in to size_o)
        fc_image <- Linear layer (size_intermid to size_o)

        gnn_layer <- CustomGNNLayer (input_dim, hidden_dim)
        #NOTE: we used GCN so that we take advantage of
        #      convolutional neural networks hierarchy

        fc_output <- Linear layer (2*size_in to output size)
        relu <- ReLU()

        #NOTE: also can be changed to a differet activation
        #      function but it seemed to be perform better
        #      with our datasets distribution

    Function forward(input_ids, attention_mask, images)
        text_outputs <- text_model(input_ids, attention_mask)[0][:, 0, :]
        text_features <- fc_text(text_outputs)

        images <- Reshape images for Swin Transformer
        image_outputs <- image_model(images)

        # Reshape image_outputs for batch and image dimensions

        #Pass through linear layer
        image_features <- fc_image(image_outputs)

        If shape mismatch between text_features and image_features
            Handle mismatch

        #Concatenate text_features and image_features
        #and pass them through the GNN layer
        #NOTE: see formula below that explains the
        #      process of concatenation

        #Concatenate gnn output with the previous results
        x <- fc_output(x)
        return x
```

Listing 5.2: ArPa class pseudocode

The mathematical equation for concatenating image and text features and then passing them through a Graph Neural Network (GNN) layer can be represented as follows:

Let $\mathbf{T}$ represent the matrix of text features, where each row $\mathbf{t}_i$ corresponds to the feature vector of the i-th text instance, and $\mathbf{I}$ represent the matrix of image features, where each row $\mathbf{i}_j$ corresponds to the feature vector of the j-th image instance. Assuming that $\mathbf{t}_i$ and $\mathbf{i}_j$ are aligned such that they are features of the same instance (i.e., they belong to the same data point and $i = j$), the concatenation of these features for each instance can be represented as:

$$\mathbf{x}_k = [\mathbf{t}_k; \mathbf{i}_k]$$

where $\mathbf{x}_k$ is the concatenated feature vector for the k-th instance, and $[;]$ denotes the concatenation operation.

The concatenated feature vectors are then passed through a GNN layer. The output of the GNN layer for each node k can be represented as:

$$\mathbf{h}_k^{(l+1)} = \text{GNN}(\mathbf{x}_k, \{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(k)\})$$

where $\mathbf{h}_k^{(l+1)}$ is the feature vector of node k at layer $l+1$, $\mathcal{N}(k)$ represents the set of neighbors of node k in the graph, and $\{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(k)\}$ represents the set of feature vectors of the neighbors of node k at layer l. The initial feature vectors $(\mathbf{h}_k^{(0)})$ are the concatenated vectors $\mathbf{x}_k$.

This GNN operation aggregates information from the neighbors of each node to update its feature representation, effectively incorporating both local and global information from the graph structure into the feature representation of each instance (both textual and visual as seen below). To make this statement more evident, one only has to think that since we did the same thing using ViT-Net with the help of neural trees, now we expand this data structure to a GNN which has all the functionalities of a Neural Tree but is far more scalable.

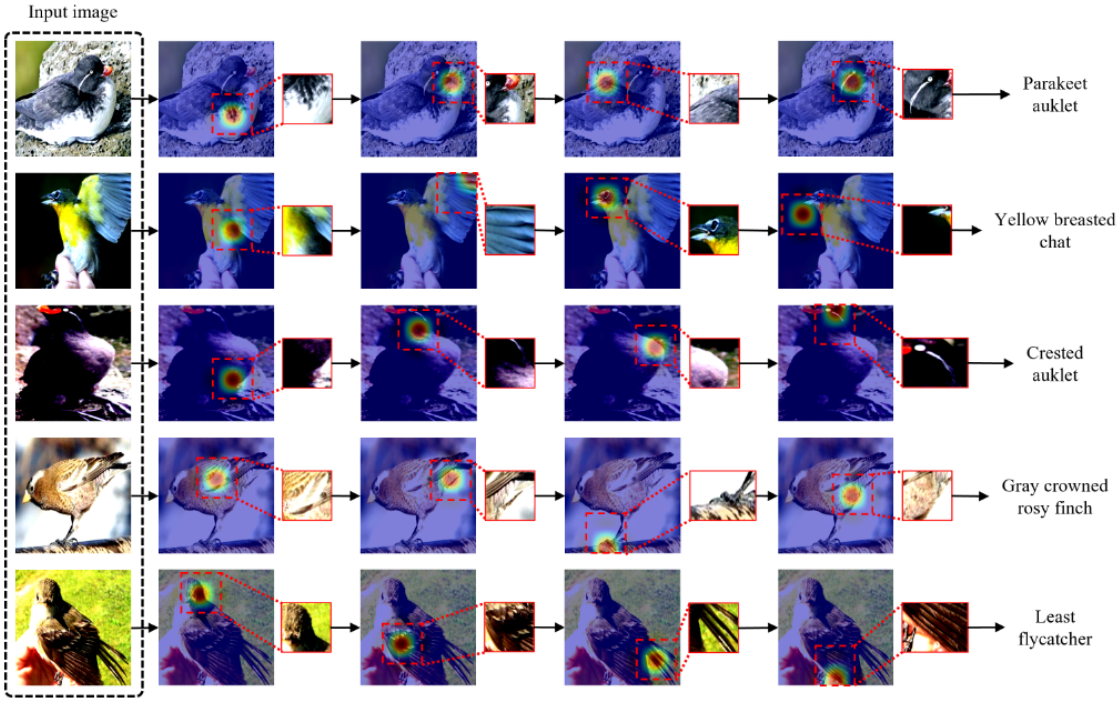

Figure 5.1.1: How the information from the visual contents is extracted

Above we can see the proposed architecture methodology (the vision parts) unveiled through visualized local interpretations, the sequential decision-making process by examining randomly selected images. This technique successfully identified various features of the subjects presented, such as tails, beaks, wings, feathers, claws, and eyes, with precision. The ability of ArPa to recognize and differentiate these characteristics demonstrates its potential in detailed image analysis, providing valuable insights into the specific elements that constitute the overall image composition. This part of ArPa works the same way as Vit-Net [15]

5.2 Advantages of ArPa

First of all, ArPa was inspired based on the idea of an architectural pipeline proposed by the authors of ViT-Net architecture [15]. Vit-Net has demonstrated a state-of-the-art performance in image classification, aiming to accurately classify fine-grained objects with similar inter-class correlations and different intra-class correlations. Although Vit-Net has applications that refer only to image classification, it proposes an intriguing architecture, using visual transformers and Neural tree decoders to achieve image classification. Yet, when it comes to the complexities of Visual Word Sense Disambiguation, an adapted version of this architecture might struggle. This is because the task requires the integration and interpretation of both textual and visual data—a challenge that a Neural Tree Decoder alone may not be well-equipped to address, potentially leading to difficulties in understanding how the model arrives at its conclusions.

Taking under consideration the limitations of previously mentioned tactics used in visual word disambiguation tasks, we came up with ArPa. The ArPa model, with its integrated architecture, presents several advantages that stem from the model's capability to process and synthesize complex multimodal data, combining the rich contextual information from both textual and visual domains to enhance disambiguation accuracy. The following section explores these benefits in detail:

- **Contextual Comprehension and Feature Integration**: ArPa leverages the contextual understanding provided by **DistilBert to interpret text, allowing it to grasp the subtleties of language and the varied meanings words can convey in different contexts**. When combined with the visual perception capabilities of the **Swin Transformer, which extracts and processes image features, the model achieves a level of multimodal understanding that mirrors human-like interpretation of visual-textual content.** This dual-pathway processing ensures that the semantic relationships within the text are not considered in isolation but are instead enriched by relevant visual cues, leading to a more nuanced and accurate word sense prediction.
- **Graph-Based Relational Learning**: The GNN component of the ArPa architecture adds another layer of advantage by modeling the relationships between words and visual elements as a graph. This approach allows ArPa to capture and utilize the relational dynamics within the data, which is paramount in scenarios where the meaning of a word can be significantly influenced by its relationship to other words and associated images. Indeed, one of the limitations of using neural tree decoders (NTDs), such as the Neural Tree Decoder (NeT) used in Vit-Net, especially in the context of Visual Word Sense Disambiguation (WSD), is their sensitivity to the design choice of tree depth. **A tree that is too shallow may fail to capture the necessary hierarchical features of the data, while a tree that is too deep does not necessarily enhance performance and may indeed lead to overfitting**. This means that the model could become too specialized to the training data, losing its ability to generalize to unseen data—a critical aspect for WSD tasks, where the ability to handle a wide variety of contexts is essential. On the other hand, **Graph Neural Networks (GNNs)**, do not suffer from these specific depth-related issues. **They are less prone to overfitting in the context of rich, relational data commonly found in Visual WSD** tasks
- **Enhanced Disambiguation Accuracy**: By integrating features from text and image sources before processing them through the graph-based framework, ArPa can discern the correct meaning of ambiguous words with greater accuracy. This is especially important in cases where visual information plays a crucial role in disambiguation, such as in instances where the same word may have different meanings in different visual contexts. The ArPa model's ability to correlate and fuse multimodal features directly addresses this challenge, leading to more reliable sense predictions.
- **Scalability and Adaptability**: Finally, the modular nature of ArPa's architecture lends itself to scalability and adaptability. Each component—textual, visual, and relational—can be independently fine-tuned or replaced with more advanced versions as the field evolves, ensuring that the model remains state-of-the-art. This adaptability not only extends the model's lifespan but also allows for customization to specific WSD tasks or datasets, enhancing its applicability across different domains and applications.

5.3 What makes this model stand out?

The technical sophistication of ArPa's pipeline for Visual Word Sense Disambiguation (WSD) emerges from its unique integration of multimodal data processing, the strategic freezing of pre-trained layers, and the employment of a Graph Neural Network (GNN) for contextual analysis.

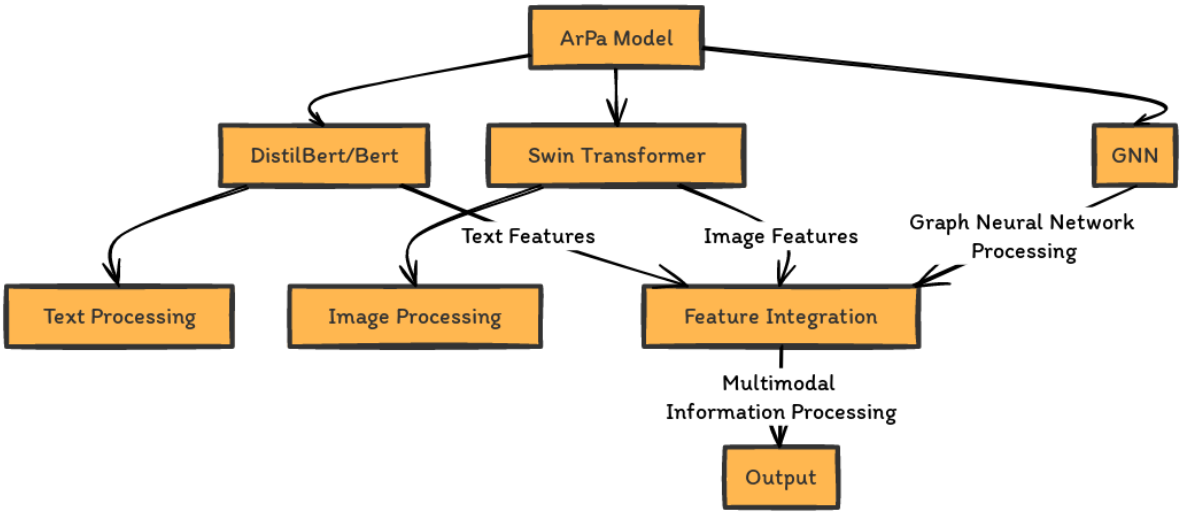

Figure 5.3.1: Detailed ArPa Model Structure

Here's a detailed examination of why ArPa's design is particularly effective for visual WSD:

- **Integration of Pre-trained Models and LLMs for Text and Image Processing**: ArPa employs 'DistilBertModel' for text and 'SwinModel' for images, tapping into the robust pre-existing knowledge encoded in these models. By initializing these components with weights from pre-trained models, ArPa leverages deep semantic and visual understandings without starting from scratch. This approach significantly enhances its capability to discern nuanced differences in word senses based on textual context and visual cues.
- **Layer Freezing to Maintain Pre-trained Knowledge**: The decision to freeze the parameters of both the text and image models ('DistilBertModel' and 'SwinModel') ensures that the nuanced, pre-trained capabilities of these models are preserved during the training process. This technique **prevents the overfitting of the model** to the training data and maintains the generalizability of the model, which is crucial for the zero-shot or few-shot learning scenarios often encountered in WSD tasks.
- **Leveraging Swin Transformer for Advanced Image Processing**: The use of the Swin Transformer ('SwinModel') for image processing is a key technical choice that significantly enhances ArPa's performance on visual WSD tasks. Swin Transformers are designed to capture hierarchical visual features by leveraging shifted windowing schemes, allowing for efficient computation of self-attention across different scales of the image. This design enables ArPa to process images in a way that **captures both fine-grained details and broader contextual information, crucial for understanding complex visual scenes**. By integrating the Swin Transformer, ArPa benefits from its ability to dynamically adjust to the varying scales and complexities of visual data, ensuring that the model can accurately interpret visual cues in a way that traditional convolutional networks might not. This advanced image processing capability allows ArPa to more effectively match images with their textual descriptions, significantly improving its ability to disambiguate words based on visual context.
- **Employment of a Graph Neural Network (GNN)**: The introduction of a GNN layer represents a significant innovation in contextual analysis. **By treating the concatenated text and image features as nodes in a graph, and employing a GNN to analyze these nodes in**

the context of their connections(i.e., the relationships between different pieces of text and images), **ArPa can understand the broader context beyond the linear sequence of words or the content of individual images.** *This is particularly powerful for disambiguating words that require an understanding of the interplay between multiple elements in a dataset.*

- **Dynamic Handling of Text and Image Feature Mismatches**: The pipeline includes a mechanism to handle mismatches in the number of text and image samples processed. This ensures that the model can dynamically adjust to varying amounts of textual and visual data, maintaining its performance even when the input data is unbalanced. Such flexibility is vital in real-world applications where the data may not always be perfectly aligned.
- **Effective Use of Concatenation and Fully Connected Layers for Classification**: Finally, the concatenation of GNN outputs with the original features, followed by processing through a fully connected output layer ('fc_output'), enables the model to **make nuanced distinctions between different senses of a word** based on a comprehensive analysis of both text and image features. This step is crucial for translating the complex, multimodal understanding developed by the model into precise, disambiguated outputs.

Together, these technical elements make ArPa's pipeline not just innovative but also highly effective for Visual WSD. The strategic choices in its design—from leveraging pre-trained knowledge and preserving it through layer freezing, to integrating advanced GNN techniques for context analysis ensure that ArPa is well-suited to the challenges of interpreting and disambiguating words based on visual context.

Chapter 6

Experiments

The primary objective of this chapter is to present a comprehensive and systematic examination of various multimodal models, like ArPa, Vision-and-Language Transformer (ViLT), Vision Text Dual Encoder, and the Contrastive Language–Image Pre-training (CLIP) model. By leveraging the Visual-WSD dataset, enriched with data from Wikidata, OmegaWiki, and BabelPic, we aim to assess these models' ability to accurately disambiguate words in contexts where visual cues play a pivotal role. Furthermore, we strive to explore the models' performance across different languages and conceptual categories, highlighting their generalizability and adaptability to diverse linguistic and cultural contexts.

Although this chapters tries to prove through statistical results and visuals that the ArPa model can perform better than the other methods that were presented in the SemEval contest 2023, we also explore how we could possibly optimize these methods by creating a pipeline with multiple models or using different preprocessing practices.

Drawing from the experimental outcomes and comparative analysis, we will distill key insights that shed light on the current state of multimodal models in the realm of V-WSD. This discussion will also touch upon the practical implications of our findings, offering guidance for researchers and practitioners in selecting, fine-tuning, and deploying these models in various applications. Moreover, we will identify promising directions for future research, inspired by the challenges and limitations uncovered through our experiments.

Figure 6.0.1: The task is to select the image that best represents the meaning of the focus word (e.g., bat) in the context (e.g., "baseball bat.")

6.1 Preliminaries

6.1.1 Dataset

The Visual-WSD dataset, through its intricate construction process and careful curation of data sources, offers a unique resource for advancing research in the field of Word Sense Disambiguation. By integrating visual elements with textual contexts and spanning multiple languages, it sets the stage for developing more sophisticated models capable of understanding the complex interplay between text and imagery in representing word senses. This dataset not only challenges existing models but also encourages the exploration of new methodologies in multimodal language understanding.
In more detail:

- Construction of the Visual-WSD Dataset

The development and assembly of the Visual Word Sense Disambiguation (Visual-WSD) dataset represent a critical step in advancing the understanding of how words are represented across different modalities, specifically textual and visual contexts. This dataset is built on the foundation of several key resources: Wikidata, OmegaWiki, and BabelPic, each serving a unique role in the dataset's construction.

- Data Sources and Integration

Wikidata and OmegaWiki serve as the primary lexical resources, with Wikidata aiming to create a free, collaborative knowledge base and OmegaWiki functioning as an open dictionary. These platforms offer a rich tapestry of concepts or named entities, each linked to related images, providing a visual dimension to the textual information. **BabelPic** further enriches this dataset by associating images with BabelNet's extensive multilingual dictionary entries, particularly focusing on abstract concepts not typically represented visually. BabelNet bridges these distinct resources, creating a cohesive dataset by linking the visual and textual data from Wikidata, OmegaWiki, and BabelPic. The use of said data sources is the reason we experimented with WikiCLIP as it would be expected to perform better since it was trained on the same data source.

- Dataset Construction

The construction process of the Visual-WSD dataset is meticulously designed to capture the nuanced relationship between words and their corresponding visual representations. Each dataset entry comprises a target word, contextual trigger words, and a set of ten images. These images include one that accurately reflects the target word's intended meaning, while the remainder depict alternative meanings, similar terms within the same domain, or are randomly selected from the underlying resources. This diverse set of images is crucial for developing models capable of disambiguating words in a visually rich context.

- Training Data

For the creation of training data, the project utilizes BabelNet's semantic network structure to generate "silver" quality data in English. This involves selecting senses—both ambiguous and monosemous—from BabelNet's WordNet section, along with corresponding images. Contextual cues are derived from hypernyms of each concept, with additional filtering to exclude images of human faces by leveraging WordNet's categorization. This approach ensures that the training data is both rich in context and diverse in visual representation. The training data provided for this shared task consists of a silver dataset with 12,869 V-WSD instances.
Each sample is a 4-tuple $\langle f, c, I, i* \in I \rangle$ where $|I| = 10$. The contexts are generally very short, often just a single word in addition to the focus word. We randomly select 10% of the training data for use as a development set.

- Testing Data

The **gold** standard testing data is meticulously prepared in English, Farsi, and Italian, underscoring the dataset's multilingual capability. The process starts with compiling a list of ambiguous word senses from BabelNet, complete with images and definitions. Annotators, proficient in English, Farsi and Italian, then provide one or two trigger words. These words are carefully chosen to ensure they are indicative

of the word sense when paired with the definition and image, without being overtly revealing. This delicate balance aims to present a genuine challenge in text disambiguation, free from dataset artifacts and biases.

- Language Distribution in the Visual Word Sense Disambiguation Dataset

In the exploration of the Visual Word Sense Disambiguation (V-WSD) dataset, an intriguing aspect that emerges is the diversity of languages intertwined within the context provided for each focus word. This linguistic variety introduces a layer of complexity to the task, potentially impacting the performance of models trained predominantly on English data. To gauge the extent and implications of this multilingual context, we embarked on a detailed analysis of the language distribution within the dataset.

Our investigation began with a methodical review of a subset of the dataset. We meticulously selected 100 instances from the training set, aiming to manually ascertain the language of each focus word and its contextual framing. This process involved identifying the primary language of the focus word and examining the context in which it was used, paying special attention to non-English words that might influence the model's understanding and disambiguation capabilities.

The findings from our analysis revealed a notable linguistic diversity:

- **English** dominated the dataset with 82% of the instances incorporating English focus words, exemplified by terms like "waxflower" and "wildflower."
- **Latin**, reflecting its historical significance in the naming of biological taxa, accounted for 15% of the instances, with focus words such as "shorea" in "shorea genus," denoting a genus of rainforest trees.
- **German** and **French** were less prevalent, constituting 2% and 1% of the instances, respectively. An example from German includes "truppenübungsplatz" (meaning "military training area"), categorized under "workplace," and from French, "brumaire," a month in the French Revolutionary Calendar, indicating the dataset's foray into culturally and historically specific lexicons.

Language	**%**	**Example**
English	82%	waxflower, wildflower
Latin	15%	shorea genus
German	2%	truppenübungsplatz, workplace
French	1%	brumaire, month

Table 6.1: Language Distribution in the Training Set Sample

The language distribution within the V-WSD dataset underscores the complexity and richness of the task at hand. The presence of non-English words and phrases, particularly those derived from Latin in scientific contexts, highlights the necessity for V-WSD models to possess a degree of linguistic flexibility and cultural awareness. Models that can effectively navigate this multilingual terrain are likely to demonstrate superior disambiguation capabilities, especially in domains where specialized vocabulary and cross-lingual references are prevalent.
Although SemEval-2023 Task 1 was published as a multilingual task which included Italian and Farsi in the test set, we only deal with English examples as examples in foreign languages can be first translated in English and processed in the same way.

- Mitigating Model Bias

To construct a fair and challenging dataset, negative sample images are selected with the intention to minimize model bias towards unwanted artifacts. This involves incorporating both random images and those related to terms within the same domain but not directly tied to the target word. The use of BabelDomains extends the categorization to encompass a broader range of concepts, ensuring a comprehensive and balanced dataset.

- Dataset Statistics

The dataset is segmented into trial, training, and test sets, with the English dataset boasting the largest volume of data, over 12,000 silver training instances and 463 gold test annotations. For Farsi and Italian, the dataset focuses solely on providing gold test data, with 200 and 305 annotations, respectively. This distribution highlights the extensive effort invested in creating a dataset that not only spans multiple languages but also thoroughly tests the models' ability to perform Visual-WSD tasks effectively.

6.1.2 Evaluation Metrics

To rigorously assess the effectiveness and precision of models, we employ a set of evaluation metrics designed to capture the models' performance from different angles. This section elaborates on the primary metrics used to gauge model efficacy in the V-WSD context: the hit rate (or top-1 accuracy) and the mean reciprocal rank. These metrics offer insights not only into the accuracy of the models but also into their ability to consistently identify the correct image among a set of possibilities.

- Hit Rate (Top-1 Accuracy)

The hit rate stands as the primary measure of a model's accuracy in the V-WSD task. It is defined as the proportion of instances where the model correctly identifies the intended image out of a selection of candidate images. Mathematically, the hit rate can be expressed as follows:

$$\text{Hit Rate} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Instances}}$$

This metric encapsulates the model's ability to pinpoint the exact image that matches the given word in context, providing a straightforward assessment of its precision. A higher hit rate indicates superior performance, reflecting the model's adeptness at interpreting the nuanced interplay between textual cues and visual content.

- Mean Reciprocal Rank (MRR)

While the hit rate offers a direct measure of accuracy, the Mean Reciprocal Rank (MRR) [19] provides a nuanced view of the model's performance across all ranking positions. The MRR is particularly informative in scenarios where multiple images may closely relate to the target word but only one is deemed correct. It calculates the average inverse rank of the ground-truth image across all instances, as described by the formula:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}$$

where N is the total number of instances, and rank_i is the position of the correct image in the model's ranked list of candidate images for the ith instance. The MRR thus not only considers the cases where the model ranks the correct image first but also rewards models that consistently rank the correct image highly, even if not in the top position. This metric highlights the model's overall effectiveness in discerning relevant visual information from less pertinent details.

- Integrating Evaluation Metrics for Comprehensive Analysis

Together, the hit rate and MRR furnish a comprehensive evaluation framework for V-WSD systems. By analyzing both metrics in tandem, we can gain a holistic understanding of a model's performance, from its precision in correctly identifying images to its consistency in ranking relevant images highly. This dual-metric approach enables a balanced assessment, ensuring that models are rewarded for both accuracy and the ability to recognize and prioritize the most relevant visual representations in varied contexts.

6.2 Model Experiments

In our exploration of Visual Word Sense Disambiguation (V-WSD), we aim to conduct a comprehensive set of experiments to evaluate the performance of several state-of-the-art models under different conditions. The focus of our investigation will include Vanilla CLIP, Wiki-Enhanced CLIP, Vision-and-Language Transformer (ViLT),Vision Text Dual Encoder and the ArPa model. Each of these models will be subjected to two distinct experimental conditions: one without any preprocessing of the data and the other with specific preprocessing techniques applied. This section outlines the structure, objectives, and anticipated outcomes of these experiments.

In summary, the models under investigation are:

- **Vanilla CLIP**: A baseline model leveraging the Contrastive Language–Image Pre-training (CLIP) without any modifications or enhancements.
- **Wiki-Enhanced CLIP**: An adaptation of the CLIP model that incorporates additional contextual information from Wikipedia to enrich the textual input.
- **ViLT**: The Vision-and-Language Transformer (ViLT) model that processes joint representations of image and text data in a unified transformer architecture.
- **Vision Text Dual Encoder**: This model projects visual and textual inputs into a shared embedding space, promising an intriguing balance between modalities.
- **Language-Image Transformer (LiT)**: A model designed to enhance the interaction between language and visual elements, potentially offering improved performance on tasks requiring deep semantic understanding.
- **ArPa Model**: Our proposed model that consists of a pipeline of Bert LLM, Swin Transformer and GCN layers, serving as a potentially innovative approach to V-WSD.

Each model will be tested using the dataset given to us by Semeval 2023 - Task 1. The other parameter we are going to test our models twice, is whether we use preprocessing or not to our data. More specifically:

- **Without Preprocessing**: In this condition, models will directly utilize the raw data from the Visual-WSD dataset. This approach aims to evaluate the models' inherent capabilities and how well they perform with unmodified input, providing a baseline for comparison.
- **With Preprocessing**: Here, the data will undergo specific preprocessing steps before being fed into the models. Preprocessing may include normalization, augmentation, or encoding strategies designed to enhance the models' understanding and interpretation of both textual and visual information. The preprocessing techniques will be tailored to address known challenges in V-WSD, such as handling abstract concepts or improving the models' ability to discern fine-grained differences between images.

The primary objective of these experiments is to assess how different models respond to the challenges of V-WSD under varying conditions. By comparing the performance of each model with and without preprocessing, we aim to identify which models are most sensitive to the quality and format of their input data. Additionally, these experiments will help uncover the strengths and weaknesses of each model in handling the complexities of multimodal disambiguation.

We anticipate that preprocessing will generally improve model performance by providing cleaner, more informative inputs. However, the degree of improvement and the effectiveness of preprocessing techniques are expected to vary among the models. For instance, Wiki-Enhanced CLIP might show significant benefits from textual preprocessing due to its reliance on rich textual context, whereas the impact on ViLT could be different given its integrated processing of image and text data.

6.3 Results

6.3.1 Comparisons

In this section we will first present the accuracy score and MRR that each model scores for each of of the following categories, some visual examples and general comments of the performance results. Furthermore, each model was trained for 10 epochs using V100 GPU.

Without Preprocessing/Regularization of Input Data

Model	Accuracy (%)	MRR (%)
(Vanilla) CLIP	41.2	67.4
Wiki CLIP	62.47	71.64
Vilt	7	11.2
Vision Text Dual Encoder	15.09	21
LiT	68	74.92
ArPa	82.3	88.1852

Table 6.2: Performance of Models using unprocessed data

If we take a look at the table of our statistics and the plots provided on the next page, we can both validate the results based on our knowledge of each of the models architecture but also gather some new information. For example:

- The statistical superiority of ArPa can be quantitatively attributed to its unique architectural features. Specifically, the integration of DistilBert and Swin Transformer for processing text and images, respectively, alongside a GNN for feature fusion, showcases a compelling use of relational data.
- The jump in performance from LiT to ArPa (from approximately 68% to 82.3% in accuracy) underscores the statistical significance of incorporating a GNN layer into the architecture, **suggesting that capturing relational structures between multimodal features substantially enhances disambiguation capabilities**.
- LiT's performance, demonstrates **the effectiveness of tailored attention mechanisms in processing multimodal inputs**. Its architecture, designed for deeper language-image integration, statistically validates the premise that closer and more nuanced interplay between modalities boosts performance.
- The statistical difference between Vanilla CLIP and Wiki CLIP **highlights the impact of contextual enrichment on model performance**. The inclusion of Wikipedia data in Wiki CLIP provides a tangible boost in both accuracy and MRR, **statistically validating the value of external knowledge sources in enhancing model comprehension and disambiguation precision**.
- ViLT and Vision Text Dual Encoder display relatively lower performance statistics, indicating possible architectural constraints in optimally balancing or integrating multimodal information. The statistical analysis suggests that while transformer models are potent, their direct application to V-WSD tasks without additional contextual or relational processing layers may limit their effectiveness.

Although most of our model's performance statistics are encouraging, we can't stop wondering if preprocessing our input data, using the methods described in 4.1, will improve the statistics. However, before jumping into this section, let's take a look at the plots of each model's Accuracy and MRR and visualize some of these models outputs .

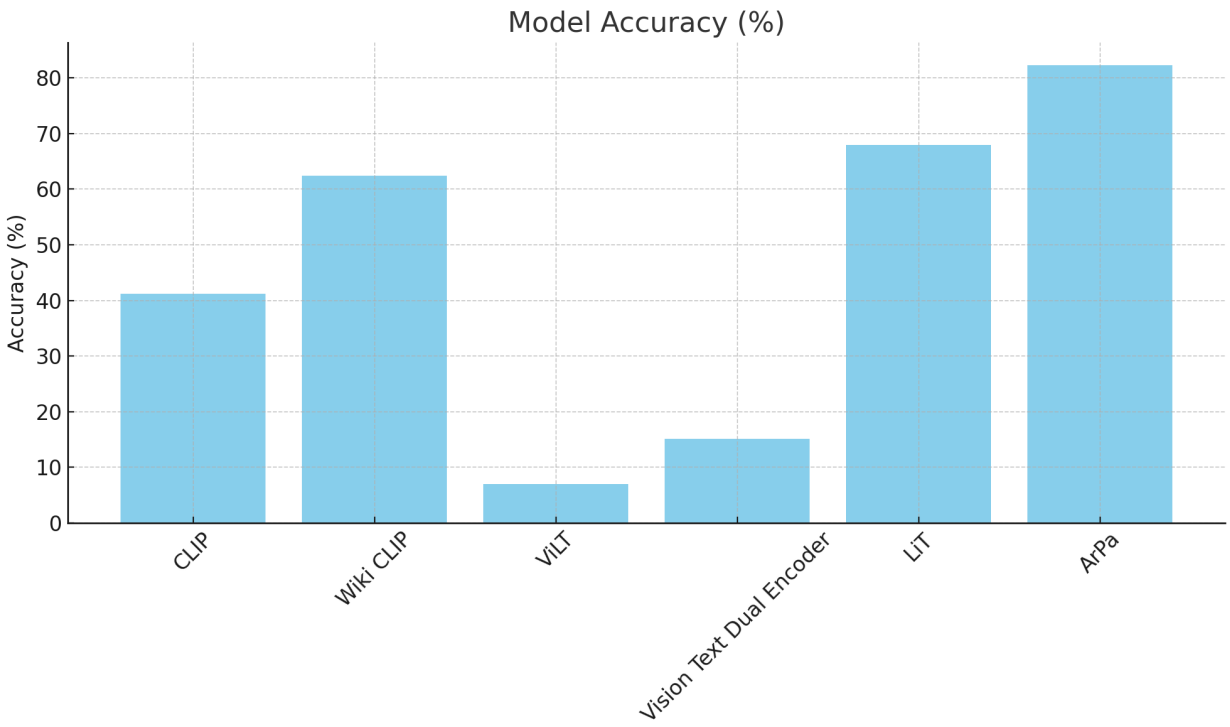

Figure 6.3.1: Model Accuracy using un-preprocessed data

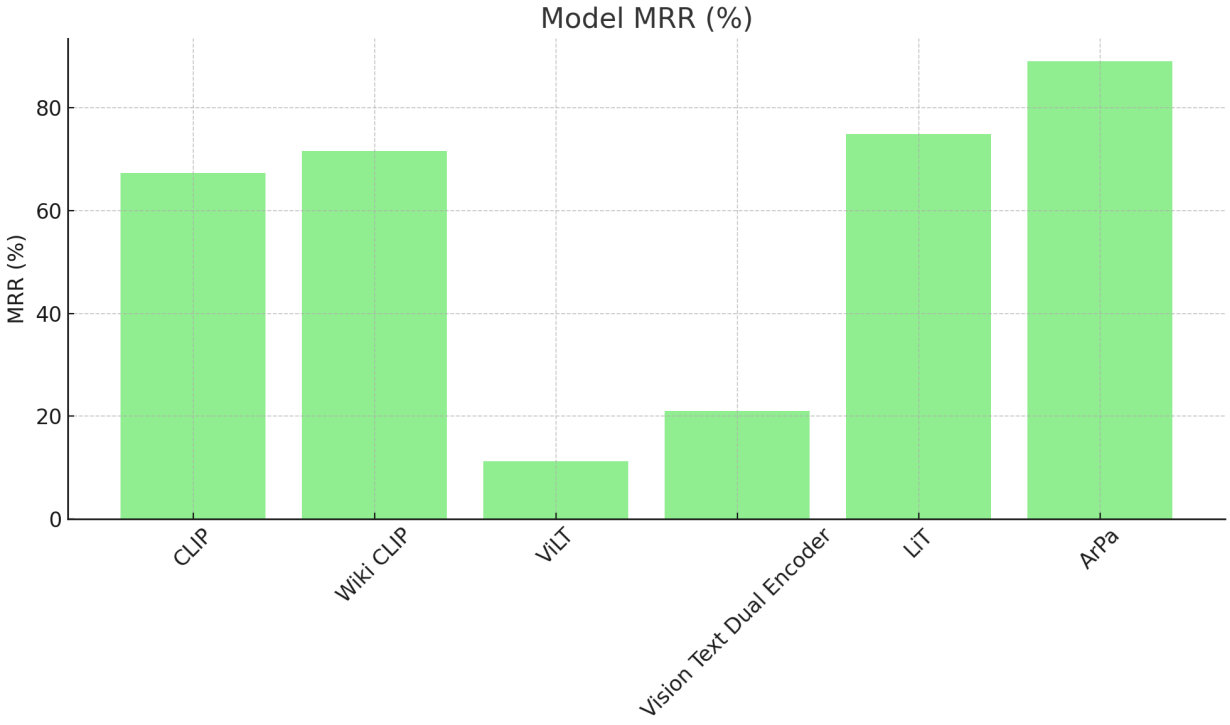

Figure 6.3.2: Model MRR using un-preprocessed data

Preprocessing of Input Data

Model	Accuracy (%)	MRR (%)
(Vanilla) CLIP	60.48	73.88
Wiki CLIP	62.47	74.92
Vilt	7.2	12
Vision Text Dual Encoder	19	22.8
LiT	71	79.81
ArPa	87.98	92.03

Table 6.3: Performance of Models using preprocessed data

Our results here for some of the models, have also been replicated by another publication of the SemEval 2023 Task 1 [27], therefore verifying our experiments. It's easy to see that there is visible improvement in most of our model's accuracies and MRR metrics. This speaks volumes for what data cleaning and regularization can achieve in the realm of Visual Word Disambiguation. More specifically:

- **Both Vanilla CLIP and Wiki CLIP**, see incremental improvements from the preprocessing benefits from the preprocessing techniques due to its generalist architecture that leverages both textual and image data for learning. **The enriched textual contexts likely help in providing more nuanced semantic cues, while advanced image preprocessing ensures that visual features are more informative**. These enhancements explain the observed solid performance, though the models general-purpose nature may limit its ability to fully exploit the richer data compared to more specialized models.
- **ViLT's (and similarly the Vision Text Dual Encoder) relatively lower performance, despite preprocessing, might indicate architectural limitations in handling the enriched multimodal data**. While the model is designed to integrate visual and linguistic inputs, the complex nature of the expanded contexts and the refined image features may not be fully capitalized due to possible constraints in its integration mechanisms or the way it processes hierarchical information.
- As for ArPa and LiT it's clear that the preprocessing methods align perfectly with their architectural strengths, providing high-quality, enriched inputs that it can effectively integrate and analyze.

While all models benefit to some extent from enriched inputs, those with architectures specifically designed to integrate and analyze complex multimodal data—such as ArPa and LiT—show the most substantial improvements. This analysis highlights the synergy between preprocessing techniques and model architectures, suggesting pathways for future enhancements in multimodal understanding.

At the the next two pages you can observe the plots showing the accuracy and MRR statistics for each model throughout the training of 10 epochs. Finally on the last page, we used ArPa to visualize the process and result of a Visual Word Disambiguation task

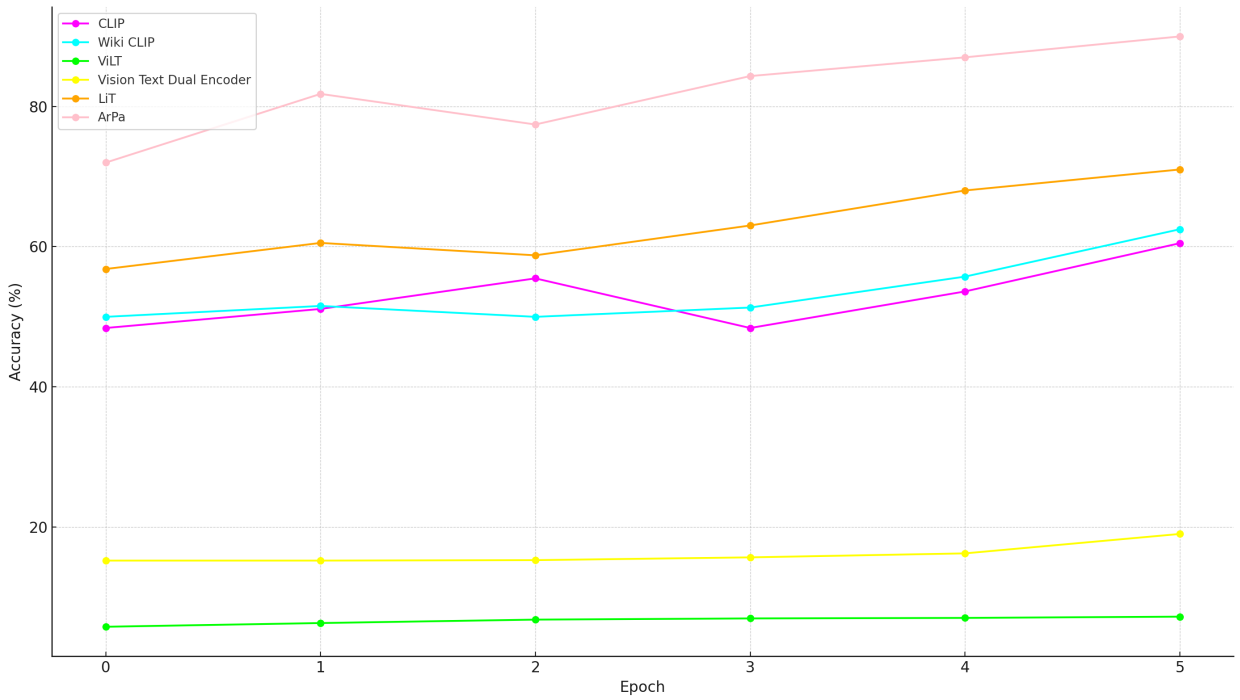

Figure 6.3.3: Model Accuracy using preprocessed data

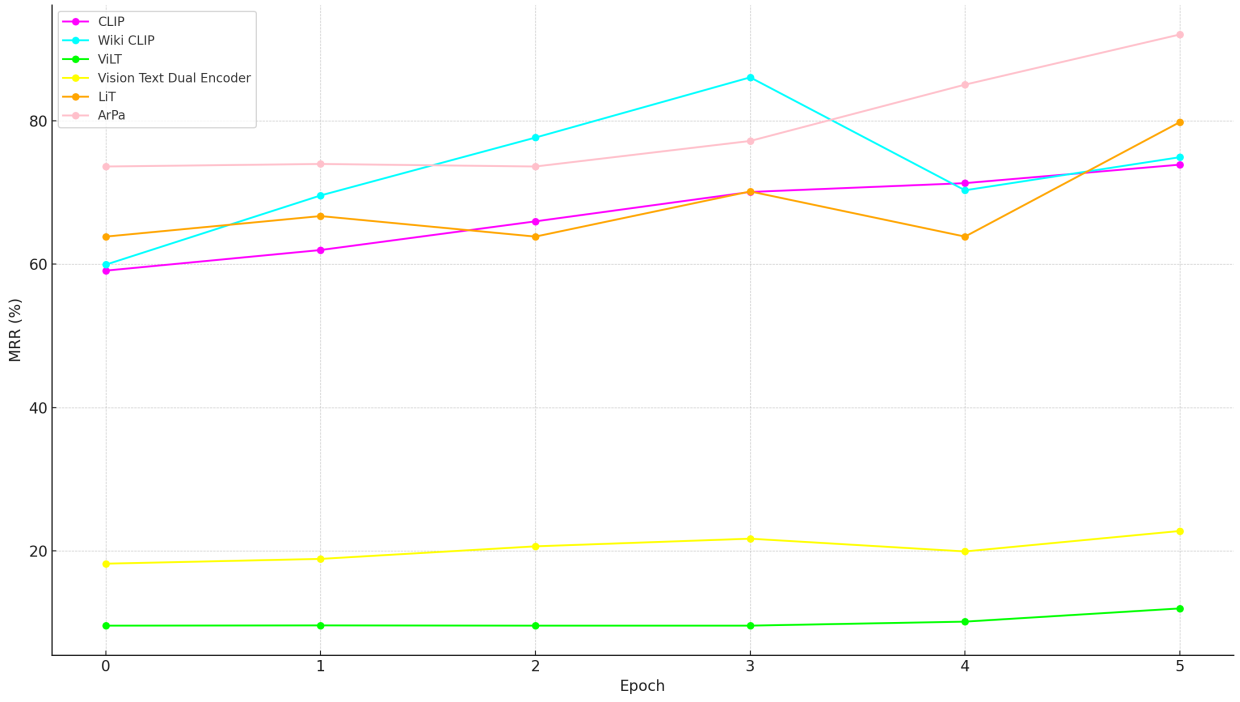

Figure 6.3.4: Model MRR using preprocessed data

Visualization of example of a V-wsd task using our model ArPa.

Our task in this case is to select the image that best represents the meaning of the focus word **"Adalia"**. This word has multiple meanings. It's the name of a specific species of a ladybug, it's a Youtube persona known as ***Adalia Rose Williams*** or it can be easily confused with the the fifth-most populous city in Turkey, pronounced as ***Attalia*** (in this case because both Antalya and Attalia have multiple letters in common, it's easy to confuse)

Figure 6.3.5: wrong image

Figure 6.3.6: wrong image

Figure 6.3.7: correct image

Our model given the **target word adalia** and the **full phrase: biology adalia**, predicted with accuracy rate of 80.14% that the correct picture is that of the ladybug!

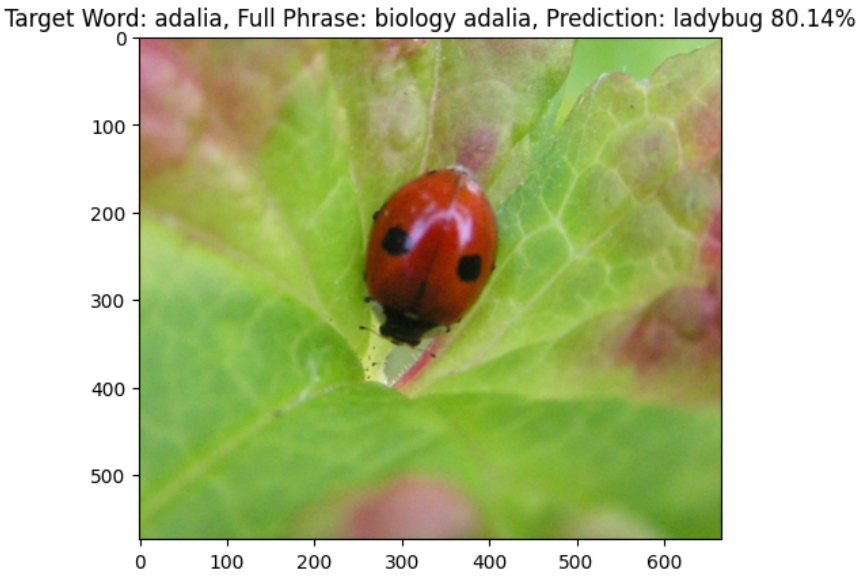

Figure 6.3.8: Example of a v-wsd task using our proposed model, ArPa

For reference LiT also chose the correct picture with accuracy of 74% , the clip models guessed correctly as well, while the ViLT and the Vision Text Dual Encoder guessed poorly and chose Antalya ...

6.3.2 Optimizations

Having considered the results of the previous testing for every model, there are some small optimizations that we could try and incorporate to any of our models, especially the ones that performed poorly.

For example, in order to help ViLT and Vision Text Dual Encoder gain better intuition of the textual and visual context, we could implement **Hierarchical Training** [25]. To implement hierarchical attention within a model's framework, we begin by structuring our input data hierarchically. This structuring might involve segmenting images into regions or categorizing objects, thereby defining various levels of hierarchy. Subsequently, the model's architecture is modified to integrate hierarchical attention mechanisms, which involves incorporating attention layers tailored to different hierarchy levels. These specialized layers focus on key features at each level, effectively grasping both the micro and macro relationships present in the data.

As the model trains, the hierarchical attention mechanism adeptly combines and prioritizes information across different levels, fostering a comprehensive understanding of hierarchical relationships. This approach significantly enhances model's proficiency in managing complex hierarchical data, thereby improving its capabilities in tasks like object recognition, scene comprehension, and image segmentation.

Also, another strategy that could prove fruitful is the implementation of **Progressive Training** [4]. Progressive training adopts an incremental learning approach, where the model's complexity and capability are systematically advanced. Initially, the model is trained on a relatively straightforward task or dataset. This initial phase lays the foundation for more intricate challenges introduced in subsequent stages. As the model progresses, it encounters increasingly complex tasks or datasets, compelling it to acquire new features and representations while retaining previously learned knowledge.

This methodical escalation in training complexity allows the model to incrementally expand its knowledge base and refine its representations. The overarching goal of progressive training is to enhance the model's generalization and transfer learning capabilities, ensuring it is well-equipped to discern and interpret complex patterns and features across a diverse array of domains and tasks.

Chapter 7

Conclusion

7.1 Discussion

This thesis represents a deep dive into the realm of Visual Word Sense Disambiguation (V-WSD), a challenging yet fascinating intersection of computer vision and natural language processing. Through this exploration, we critically evaluated a suite of cutting-edge computational models, with a spotlight on the innovative ArPa model. This journey also encompassed comprehensive analyses of other significant models like Vanilla CLIP, Wiki CLIP, LiT, ViLT, and the Vision Text Dual Encoder. The endeavor was not merely an academic exercise but a quest to unravel the layered complexities of V-WSD, spotlighting the critical roles played by model architecture and advanced preprocessing techniques in augmenting multimodal comprehension.

One of the most salient insights from this research is the unmistakable influence of a model's architectural framework on its proficiency in processing and integrating multifaceted multimodal data. The ArPa model stands as a testament to architectural ingenuity, melding DistilBert and Swin Transformer's robust capabilities with a Graph Neural Network (GNN) for an unmatched synthesis of data fusion. This architectural synergy propelled the ArPa model to new heights of performance, setting a benchmark for future endeavors in the domain of multimodal interaction and cognitive machine understanding.

The emphasis on refining preprocessing methodologies, including linguistic context expansion and advanced image processing, emerged as a cornerstone of our strategy to optimize data for model ingestion. This nuanced approach to data preparation markedly enhanced the models' ability to adeptly maneuver through the complex terrains of V-WSD, achieving higher accuracy and efficiency.

The rigorous experimentation and analytic scrutiny undertaken in this thesis have yielded invaluable insights and methodologies, significantly contributing to the broader discourse on artificial intelligence and multimodal learning. This work elucidated the intricate dance between visual and textual data, providing concrete, evidence-based strategies to navigate and surmount the challenges inherent in V-WSD.

In drawing this research to a close, it's evident that the journey has not only expanded our understanding of Visual Word Sense Disambiguation but also lit the path for future exploration in the seamless integration of visual and textual information. By identifying and articulating the key drivers of model performance within V-WSD tasks, this thesis lays down a foundational blueprint for the evolution of more nuanced, empathetic, and contextually aware artificial intelligence systems. Standing at the cusp of new discoveries, the insights distilled from this thesis invite us to envision a future wherein the harmonious blending of language and vision transcends current limitations, fostering an era of technological advancement and novel forms of interaction. This thesis, therefore, is not just a culmination of research but a beacon guiding us toward the untapped potential of multimodal artificial intelligence.

7.2 Future Work

In closing this thesis we would like to suggest a few directions to further improve on this work or inspire different interesting approaches. Future research should aim to develop even more sophisticated multimodal models that can seamlessly integrate visual and textual information with higher degrees of accuracy and nuance. Inspired by the success of the ArPa model, investigations into novel architectures that leverage the latest advancements in machine learning—such as deeper Graph Neural Networks, enhanced transformer models, and innovative attention mechanisms—will be crucial. These models should not only excel in V-WSD tasks but also demonstrate adaptability and generalizability across diverse domains and datasets.

Integrating a wider variety of external knowledge sources, beyond Wikipedia and WordNet, could provide models with a richer contextual understanding and enable them to tackle more complex disambiguation challenges. Future work could explore the incorporation of domain-specific encyclopedias, cultural databases, and even multimedia content as part of the preprocessing phase, enriching the models' knowledge base and enhancing their interpretative depth.

Given the often limited availability of labeled data in specific domains, future research could benefit from exploring unsupervised and semi-supervised learning approaches. These methods, capable of leveraging large amounts of unlabeled data, could unlock new pathways for model training and refinement, particularly in areas where manual annotation is impractical.

Finaly, improving transfer learning strategies to facilitate the seamless adaptation of models to new tasks and domains represents another promising direction for future research. By refining these strategies, researchers can enhance the efficiency and effectiveness of model training processes, enabling quicker deployment and broader applicability of V-WSD technologies.

In essence, the future of Visual Word Sense Disambiguation and multimodal learning is brimming with opportunities for groundbreaking research and transformative applications. The foundation laid by this thesis serves as a springboard for future endeavors, inviting scholars and practitioners alike to venture into uncharted territories of artificial intelligence.

Chapter 8

Bibliography

[1] Agostina Calabrese Michele Bevilacqua, R. N. "EViLBERT: Learning Task-Agnostic Multimodal Sense Embeddings". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (2020).

[2] Alec Radford Chris Hallacy, A. R. et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *arXiv preprint arXiv:2103.00020v1* (2021).

[3] Bronstein, M. M. et al. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[4] Changlin Li Bohan Zhuang, G. W. and Xiaodan Liang Xiaojun Chang, Y. Y. "Automated Progressive Learning for Efficient Training of Vision Transformers". In: *arXiv preprint arXiv:2203.14509* (2022).

[5] Developers, G. *Foundational Courses - Embeddings*. [Online; accessed 23-September-2022]. 2022.

[6] Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

[7] Hinton, G. E. and Salakhutdinov, R. R. "Reducing the dimensionality of data with neural networks". In: *science* 313.5786 (2006), pp. 504–507.

[8] James Im. *Introduction to PCA (Principal Component Analysis) - Medium*. [Online; accessed 23-September-2022]. 2018.

[9] Keim, R. *How to Train a Basic Perceptron Neural Network*. en. [online]. 2019. URL:

[10] Mikolov, T. et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[11] Miller, G. A. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[12] Pennington, J., Socher, R., and Manning, C. D. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL:

[13] Pires, T., Schlinger, E., and Garrette, D. "How multilingual is multilingual BERT?" In: *arXiv preprint arXiv:1906.01502* (2019).

[14] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[15] Sangwon Kim Jaeyeal Nam, B. C. K. "ViT-NeT: Interpretable Vision Transformers with Neural Tree Decoder". In: *Proceedings of the 39th International Conference on Machine Learning* (2022).

[16] Si Zhang Hanghang Tong1, J. X. and Maciejewski, R. "Graph convolutional networks: a comprehensive review". In: *SpringerOpen* (2019).

[17] Vaswani, A. et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[18] Veličković, P. et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[19] Wikipedia contributors. *Mean reciprocal rank — Wikipedia, The Free Encyclopedia*. [Online; accessed 27 August 2022]. 2022.

[20] Wikipedia contributors. *Principal component analysis — Wikipedia, The Free Encyclopedia*. [Online; accessed 23-September-2022]. 2022.

[21] Wikipedia contributors. *Rectifier (neural networks) — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-September-2022]. 2022.

[22] Wikipedia contributors. *Vanishing gradient problem — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-September-2022]. 2022.

[23] Wikipedia contributors. *Word embedding — Wikipedia, The Free Encyclopedia*. [Online; accessed 23-September-2022]. 2022.

[24] Xiaohua Zhai Xiao Wang, B. M., Andreas Steiner Daniel Keysers, A. K., and Beyer, L. "Learning Transferable Visual Models From Natural Language Supervision". In: *arXiv preprint arXiv:2111.07991* (2021).

[25] Yamin Sepehri Pedram Pad, A. C. Y. and Pascal Frossard, L. A. D. "Hierarchical Training of Deep Neural Networks Using Early Exiting". In: *arXiv preprint arXiv:2303.02384* (2023).

[26] Zhou, J. et al. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81.

[27] Zhuohao Yin, X. H. "HKUST at SemEval-2023 Task 1: Visual Word Sense Disambiguation with Context Augmentation and Visual Assistance". In: *arXiv preprint arXiv:2311.18273* (2023).