Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Large Language Models and Multimodal Retrieval for Visual Word Sense Disambiguation

Diploma Thesis

by

Anastasia Kritharoula

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2023

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Large Language Models and Multimodal Retrieval for Visual Word Sense Disambiguation

Diploma Thesis

by

Anastasia Kritharoula

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την $26^{η}$ Οκτωβρίου, 2023.

....	....	....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Μιχάλης Βαζιργιάννης
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ecole Polytechnique

Αθήνα, Οκτώβριος 2023

....

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

Περίληψη

Η Αποσαφήνιση Οπτικών Εννοιών (Visual Word Sense Disambiguation - VWSD) είναι ένα καινούριο πρόβλημα με πολλές προκλήσεις, που μπορεί να τοποθετηθεί στην τομή μεταξύ του προβλήματος αποσαφήνισης γλωσσικών εννοιών και του προβλήματος ανάκτησης εικόνων από κειμενικές περιγραφές. Με αυτή τη διατριβή θα επιχειρήσουμε ένα πρώτο ουσιαστικό βήμα προς την αναγνώριση και αντιμετώπιση του νέου αυτού προβλήματος, εφαρμόζοντας ένα ευρύ σύνολο μεθόδων. Οι πρόσφατες εξελίξεις και καινοτομίες στον τομέα των Οπτικογλωσσικών Μετασχηματιστών (VL Transformers) παρουσιάζουν υλοποιήσεις με ενθαρρυντικά αποτελέσματα, τα οποία ωστόσο υποστηρίζουμε ότι μπορούν να ενισχυθούν περαιτέρω. Για το λόγο αυτό, προτείνουμε κάποιες τεχνικές ενίσχυσης γνώσης που έχουν σκοπό να βελτιώσουν την απόδοση ανάκτησης των Οπτικογλωσσικών Μετασχηματιστών με χρήση Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) ως Βάσεις Γνώσεων. Πιο συγκεκριμένα, επιχειρούμε την ανάκτηση της γνώσης που είναι αποθηκευμένη αθόρυβα στα βάρη των Οπτικογλωσσικών Μετασχηματιστών, επερωτώντας τα με κατάλληλες φράσεις, που καλούμε *προτροπές*, σε συνθήκες μηδενικής ρύθμισης- χωρίς τη χρήση κάποιας διαδικασίας προεκπαίδευσης ή κάποιων αντιπροσωπευτικών κατευθυντήριων παραδειγμάτων. Επιπλέον, μελετάμε το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών είτε από την οπτική ενός προβλήματος Ανάκτησης Εικόνων από Εικόνες, είτε ενός προβλήματος Ανάκτησης Κειμένου από Κείμενου έτσι ώστε να διευρευνήσουμε πλήρως τις δυνατότητες των πιο σύγχρονων καινοτόμων μοντέλων που χρησιμοποιούνται για την επίλυση των προβλημάτων αυτών. Επακόλουθα, εκπαιδεύουμε ένα Μοντέλο Εκμάθησης Κατάταξης με σκοπό να συνδυάσουμε τις διαφορετικές προσεγγίσεις μας, επιτυγχάνοντας ανταγωνιστικά αποτελέσματα.

Επιπρόσθετα, μεατρέπουμε το πρόβλημα Αποσαφήνισης Οπτικών Εννοιών σε ένα αμιγώς γλωσσικό πρόβλημα Ερώτησης-Απάντησης, θεωρώντας τις παραγόμενες λεζάντες των υποψήφιων εικόνων ως υποψήφιες απαντήσεις πολλαπλής επιλογής. Στρατηγικές χρήσης προτροπών με μηδενική βολή-χωρίς την ενσωμάτωση ενδεικτιών παραδειγμάτων- ή πολλαπλή βολή-με ενσωμάτωση μερικών ενδεικτικών παραδειγμάτων- προστίθενται στα πειράματά μας προκειμένου να εξερευνήσουμε πλήρως τη δυνατότητα ενός τέτοιου μετασχηματισμού για το πρόβλημα αυτό. Παράλληλα, η χρήση προτροπών Αλυσιδωτής-Σκέψης με μηδενική-βολή μας αποκαλύπτει τα εσωτερικά βήματα συλλογιστικής που ακολουθεί ένα Μεγάλο Γλωσσικό Μοντέλο για να επιλέξει την κατάλληλη υποψήφια εικόνα. Συνολικά, η προσέγγιση που παρουσιάζουμε είναι η πρώτη που αναλύει τα πλεονεκτήματα της αξιοποίησης γνώσης που βρίσκεται αποθηκευμένη στα Μεγάλα Γλωσσικά Μοντέλα με διαφορετικούς τρόπους για την επίλυση του προβλήματος της Αποσαφήνισης Οπτικών Εννοιών.

Λέξεις-κλειδιά — Αποσαφήνιση Οπτικών Εννοιών, Πολυτροπική Ανάκτηση Εικόνας-Κειμένου, Οπτικογλωσικοί Μετασχηματιστές, Μεγάλα Γλωσσικά Μοντέλα, Μεγάλα Γλωσσικά Μοντέλα ως Βάσεις Γνώσεων, Συλλογιστική Διαδικασία σε Μεγάλα Γλωσσικά Μοντέλα

Abstract

Visual Word Sense Disambiguation (VWSD) is a challenging task that lies at the intersection of linguistic sense disambiguation and fine-grained mulitmodal retrieval. In this task, the goal is to retrieve the appropriate image from a set of competitive candidates, given a word within a given context. In this thesis, we aim to make a substantial step towards unveiling this interesting task. As a starting point, we propose some recent state-of-the-art visiolinguistic (VL) transformers with promising baseline performance. We suggest the use of Large Language Models (LLMs) as Knowledge Bases, which could better the retrieval performance of VL transformers via knowledge-enhancement, in order to improve these baselines. Specifically, we utilise appropriate prompts to query the LLMs and retrieve the knowledge which is stored in their weights, thereby accomplishing performance improvements. We also study VWSD as a unimodal problem by converting to text-to-text and image-to-image retrieval, in order to thoroughly investigate the capabilities of relevant models. To combine our various modules, we train a learn-to-rank (LTR) model on a dataset derived by combining the features of the aforementioned techniques.

Moreover, we transform VWSD into a text-only question-and-answer (QA) problem. To achieve this, we designate each image with a generated caption and use the captions as potential multiple-choice textual answers. To reveal the potential of such a transformation, we employ zero-shot and few-shot strategies, as well as Chain-of-Thought (CoT) prompting in the zero-shot setting, in order to evoke the internal reasoning steps an LLM employs to select the most suitable candidate and to provide internal explanations for this selection. Overall, this thesis is the first one that attempts to analyse the merits of leveraging knowledge stored in LLMs in various ways to solve VWSD.

Keywords — Visual Word Sense Disambiguation, Multimodal Retrieval, VL Transformers, Large Language Models, Language Models as Knowledge Bases, Reasoning in Large Language Models

Ευχαριστίες

Θα ήθελα να τονίσω ότι το έργο αυτό δεν θα ήταν δυνατό χωρίς την υποστήριξη πολλών ανθρώπων. Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα μου, κ. Στάμου Γεώργιο, για την πολύτιμη καθοδήγηση του στην εκπόνηση αυτής της διπλωματικής, καθώς και την Μαρία Λυμπεραίου για τη στενή συνεργασία, την υποστήριξη και την καθοριστική συμβολή της στην εργασία αυτή.

Πολύ σημαντική για εμένα ήταν ακόμα η συναισθηματική συνεισφορά της οικογένειας και των φίλων μου που μου παρείχαν αγάπη, στήριξη και γέλιο. Ιδιαίτερα, θα ήθελα να ευχαριστήσω και τον αδερφό μου Διονύση με τον οποίο μάλιστα μοιραστήκαμε και αυτό το ακαδημαϊκό ταξίδι από την αρχή μέχρι το τέλος και βγήκαμε νικητές.

Κριθαρούλα Αναστασία, Οκτώβρης 2023

Contents

List of Figures

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

Contents

1.1 Θεωρητικό Υπόβαθρο

Η *Αποσαφήνιση Οπτικών Εννοιών* (Visual Word Sense Disambiguation - VWSD) είναι ένα καινούριο πρόβλημα με αρκετές προκλήσεις. Σκοπός του προβλήματος αυτού είναι δοθείσας μιας διφορούμενης λέξης-στόχου εντός ενός κειμένου, που έχει το ρόλο ενοιολογικού πλαισίου, να ανακτηθεί η κατάλληλη εικόνα από τις υποψήφιες, η οποία αντιπροσωπεύει καλύτερα την έννοια της λέξης-στόχου[67]. Για παράδειγμα, η φράση *Το δέντρο ανδρομέδα* περιέχει την αμφίσημη λέξη-στόχο *ανδρομέδα* συνοδευόμενη από το κείμενο-πλαίσιο *δέντρο* το οποίο και την αποσαφηνίζει. Μεταξύ των 10 υποψηφίων εικόνων που παρουσίζονται στην εικόνα 1.1.1, ένα σύστημα σχεδιασμένο για την επίλυση του προβλήματος Αποσαφήνισης Οπτικών Εννοιών επιχειρεί να ανακτήσει την εικόνα εκείνη που αναπαριστά το δέντρο γνωστό ως ανδρομέδα, η οποία επισημειώνεται με ροζ πλαίσιο.

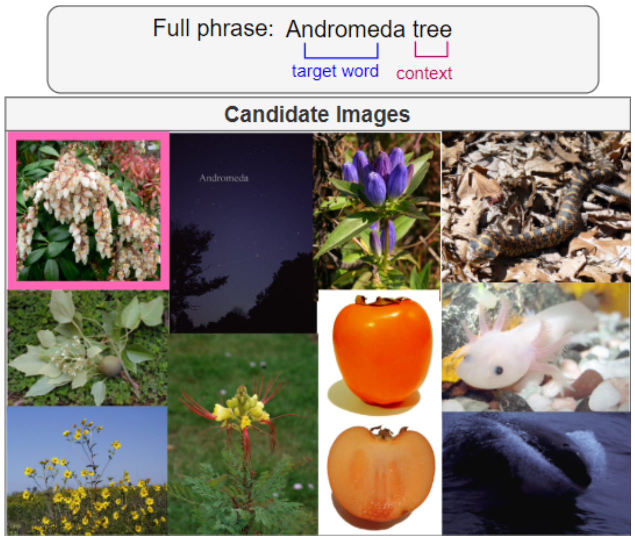

Figure 1.1.1: Παράδειγμα του προβλήματος Αποσαφήνισης Οπτικών Εννοιών

Παρότι το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών είναι ουσιαστικά ένα πολυτροπικό πρόβλημα ανάκτησης Εικόνας-Κειμένου, υπάρχουν κάποιες θεμελιώσεις διαφορές. Για παράδειγμα παρατηρώντας τις υποψήφιες εικόνες στην εικόνα 1.1.1, οι οποίες με κάποιο τρόπο είναι σχετικές με την αμφίσημη λέξη *ανδρομέδα* (μπορεί να είναι είτε αστερισμός, είτε είδος ψαριού, δέντρου, ερπετού κτλ), ένα επιτυχημένο σύστημα ανάκτησης πρέπει να είναι λεπτομερές και ιδιαίτερα ευαίσθητο στην τοποθέτηση της αμφίσημης λέξης-στόχου στο εννοιολογικό πλαίσιο. Σε αυτή την περίπτωση, το πλαίσιο *δέντρο* πρέπει να είναι αρκετά εμφανές -από την οπτική του συστήματος ανάκτησης- προκειμένου να μπορεί να αποσαφηνίσει τη έννοια. Παράλληλα όμως, αυτό το σύστημα ανάκτησης δεν πρέπει να βασίζεται αποκλειστικά στο εννοιολογικό πλαίσιο *δέντρο*: στην περίπτωση αυτή εικόνες που περιέχουν λουλούδια και πράσινο γρασίδι εισάγουν κάποια αναμενόμενη μεροληψία στη διαδικασία ανάκτησης, και με αυτόν τον τρόπο ίσως επιλεχθεί η εικόνα που έχει τη μεγαλύτερη πιθανότητα να περιέχει ένα *δέντρο*, αγνοώντας την αμφίσημη έννοια *ανδρομέδα*. Φυσικά, είναι πιθανό και το σενάριο το μοντέλο ανάκτησης να μην έχει εκπαιδευτεί ποτέ με την αμφίσημη έννοια: η σπανιότητα των εννοιών που χαρακτηρίζει το λεξιλόγιο των λέξεων-στόχων αυξάνει την πιθανότητα το σύστημα ανάκτησης να βασιστεί αποκλειστικά στο γνωστό σε αυτό εννοιολογικό πλαίσιο, με αποτέλεσμα την εισαγωγή σημαντικής τυχαιότητας κατά τη διαδικασία επιλογής. Για το λόγο αυτό, η αξιοπιστία της Αποσαφήνισης Οπτικών Εννοιών αποτελεί ένα κρίσιμο σημείο, αυξάνοντας την ανάγκη για επεξηγήσεις κατά τη διαδικασία επιλογής και επίλυσης.

Αυτή η διατριβή περιλαμβάνει ένα σύνολο προσεγγίσεων για το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών. Μεταξύ αυτών περιλαμβάνεται η χρήση σύγχρονων Μετασχηματιστών Εικόνας-Κειμένου (VL Transformers) ως συστημάτων ανάκτησης, η χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs) ως βάσεων γνώσης για την αποσαφήνιση της λέξης-στόχου καθώς και η μετατροπή του προβλήματος σε πρόβλημα Ερώτησης-Απάντησης φυσικής γλώσσας -εξάγοντας κειμενικές περιγραφές για τις υποψήφιες εικόνες- και η χρήση Αλυσιδωτού Συλλογισμού για την επεξήγηση των απαντήσεων.

1.2 Προτεινόμενες Πρεσεγγίσεις

1.2.1 Συνεισφορά

Οι συνεισφορές αυτής της διπλωματικής εργασίας είναι πολλαπλές μπορούν να συνοψιστούν ως εξής:

- Αξιοποιούμε τα Γλωωσικά Μοντέλα (LLMs) ως βάσεις γνώσης για να εμπλουτίσουμε τις δοθείσες φράσεις, ενσωματώνοντας περαιτέρω εννοιλογικό περιεχόμενο με σκοπό η λέξη-στόχος να αποσαφηνιστεί, αντιμετωπίζοντας με αυτόν τον τρόπο και περιπτώσεις που η αμφίσημη λέξη είναι άγνωστη στο σύστημα ανάκτησης.
- Μετατρέπουμε το πρόβλημα Αποσαφήνισης Οπτικών Εννοιών σε μονοτροπικό πρόβλημα, δηλαδή αποκλειστικά γλωσσικό ή αποκλειστικά οπτικό πρόβλημα. Πιο συγκεκριμένα, σε πρόβλημα Ανάκτησης Εικόνας από Εικόνα, Ανάκτησης Κειμένου από Κείμενο, καθώς και πρόβλημα Ερώτησης-Απάντησης με σκοπό να εξερευνήσουμε πλήρως τις δυαντότητες που τα σχετικά μοντέλα μπορούν να προσφέρουν.
- Χρησιμοποιούμε τα χαρακτηριστικά που εξάγουμε από τις προαναφερθείσες τεχνικές για να εκπαιδεύσουμε ένα μοντέλο Εκμάθησης Κατάταξης (LTR), επιτυγχάνοντας ανταγωνιστικά αποτελέσματα ανάκτησης.
- Ενσωματώνουμε τον Αλυσιδωτό Συλλογισμό στο πρόβλημα Ερώτησης-Απάντησης με σκοπό την κατευθυνόμενη παραγωγή απαντήσεων, ενώ παρουσιάζουμε τα ενδιάμεσα βήματα συλλογισμού που λειτουργούν ως επεξηγήσεις της διαδικασίας ανάκτησης.

1.2.2 Σύνολο Δεδομένων

Το σύνολο δεδομένων για το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών που αφορά την αγγλική γλώσσα αποτελείται από 12869 δείγματα εκπαίδευσης και 463 δείγματα δοκιμής. Κάθε δείγμα αποτελείται από 10 υποψήφιες εικόνες. Το μήκος της δοθείσας φράσης παρουσιάζει αμελητέες διαφορές με τη συντριπτική πλειοψηφία των φράσεων να αποτελούνται από 2 λέξεις. Το σύνολο δεδομένων και οι επίσημοι διαχωρισμοί μπορούν να βρεθούν στο https://raganato.github.io/vwsd/. Τα συνολικά στατιστικά του συνόλου δεδομένων παρουσιάζονται στον ακόλουθο πίνακα:

Διαχωρισμός	#Δείγματα	Μήκος φράσης			
		1 word	2 λέξεις	3 λέξεις	4 λέξεις
Εκπαίδευση	12869	0	12868	0	0
Δοκιμή	463	1	445	17	1

Table 1.1: Στατιστικά Συνόλου Δεδομένων

1.2.3 Μέθοδος

Ακολουθούμε 6 τεχνικές για να εξερευνήσουμε το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών από αρκετές διαφορετικές οπτικές γωνίες. Όλες οι τεχνικές μας έχουν δοκιμαστεί αποκλειστικά στην αγγλική γλώσσα.

Ανάκτηση Εικόνας-Κειμένου με VL Transformers ως Επίπεδο Αναφοράς

Υλοποιήσαμε ένα απλό σύστημα ανάκτησης εικόνας-κειμένου ως επίπεδο αναφοράς για να αξιολογήσουμε τις δυνατότητες των υπάρχοντων προεκπαιδευμένων VL transformers στο πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών. Οι VL transformers προβάλλουν τόσο τις εικόνες i όσο και τις γλωσσικές φράσεις t στον ίδιο μαθηματικό χώρο. Η ανάκτηση της καταλληλότερης εικόνας i με βάση τη δοθείσα γλωσσική φράση t γίνεται με βάση τον ακόλουθο βαθμό ομοιότητας:

$$score(t, i) = max(sim(t, i)) \tag{1.2.1}$$

για τον υπολογισμό του οποίου επιλέγουμε να χρησιμοποιήσουμε την ομοιότητα συνημιτόνου.

Οι VL Transformers που χρησιμοποιούμε απαριθμούνται παρακάτω:

- CLIP[65], με το ViT-base[15] ως κωδικοποιητή

- CLIP, με το ViT-large ως κωδικοποιητή, συμβολίζεται με CLIP-L
- CLIP_{LAION}, εκπαιδευμένο στο LAION-2B αγγλικό υποσύνολο του LAION-5B
- ALIGN[25]
- BLIP[41], με το ViT-base ως κωδικοποιητή, εκπαιδευμένο στο COCO[45], συμβολίζεται ως BLIP_C
- BLIP, με το ViT-large ως κωδικοποιητή, εκπαιδευμένο στο COCO, συμβολίζεται ως BLIP-L_C
- BLIP, με το ViT-base ως κωδικοποιητή, εκπαιδευμένο στο Flickr30k[93], συμβολίζεται ως BLIP_F
- BLIP, με το ViT-large ως κωδικοποιητή, εκπαιδευμένο στο Flickr30k, συμβολίζεται ως BLIP-L_F

Επιπλέον, πειραματιζόμαστε με την ενσωμάτωση ενός συντελεστή ποινής $p(i)$, όπως περιγράφεται στο [11] έτσι ώστε να προσαρμόσουμε την προτίμηση ανάκτησης των εικόνων που παρουσιάζουν υψηλό βαθμό ομοιότητας $sim(t, i)$ με πολλές φράσεις t. Η ποινή υπολογίζεται για κάθε εικόνα ως ο μέσος όρος ομοιότητας μεταξύ της εικόνας και όλων των γλωσσικών φράσεων στο σύνολο δεδομένων κανονικοποιημένος από τη συχνότητα εμφάνισης της εικόνας σύμφωνα με τον ακόλουθο τύπο:

$$p(i) = \left(\frac{1}{|T|} \sum_{t_k \in T} sim(t_k, i) \right) \cdot \frac{card(i)}{max_{i_m \in I} card(i_m)} \tag{1.2.2}$$

όπου T είναι το σύνολο όλων των φράσεων, I είναι το σύνολο όλων των εικόνων, και το $card(i)$ αντιστοιχεί στον αριθμό των δειγμάτων στα οποία εμφανίζεται η εικόνα i.

Σε αυτή την περίπτωση, ο βαθμός ομοιότητας παίρνει την ακόλουθη μορφή:

$$score(t, i) = sim(t, i) - p(i) \tag{1.2.3}$$

LLMs για τον εμπλουτισμό του εννοιολογικού πλαισίου

Επιλέγουμε να δοκιμάσουμε ένα σύνολο από LLMs ως βάσεις γνώσης για να εμπλουτίσουμε τις σύντομες φράσεις t με περισσότερες λετομέρειες σε μια ρύθμιση μηδενικής βολής-χωρίς την ενσωμάτωση ενδεικτικών παραδειγμάτων-, παράγοντας εμπλουτισμένες φράσεις t_e, και στη συνέχεια αξιοποιούμε το σύστημα ανάκτησης εικόνας-κειμένου που περιγράφτηκε στην προηγούμενη παράγραφο. Επιπλέον, επιχειρούμε και πάλι να ενσωματώσουμε τον συντελεστή ποινής $p(i)$ κατά το στάδιο της ανάκτησης, σύμφωνα με τον ακόλουθο βαθμό ομοιότητας, έχοντας ενσωματώσει τον εμπλουτισμό γνώσης:

$$score(t_e, i) = sim(t_e, i) - p(i) \tag{1.2.4}$$

Όλες οι προτροπές (prompts) με τις οποίες επερωτάμε τα LLMs είναι σχεδιασμένες με βάση χειροκίνητα κατασκευασμένα πρότυπα, βασισμένα στη διαίσθηση ότι το να ζητάς από το μοντέλο πληροφορίες με τη μορφή οδηγιών έχει αποδειχθεί ωφέλιμο [30].

Οι προτροπές που χρησιμοποιούμε παρουσιάζονται παρακάτω:

Prompt name	Prompt template
exact	"<phrase> "
what_is	"What is <phrase>?"
meaning_of	"What is the meaning of <phrase>?"
describe	"Describe <phrase>."
write_description	"Write a description of <phrase>."
to_describe	"To describe <phrase> I would say that "
could_describe	"I could describe <phrase> as "

Table 1.2: Προτροπές για εμπλουτισμό του εννοιολογικού πλαισίου μέσω LLMs

Οι δυνατότητες των LLMs είναι ανάλογες με το μέγεθος τους, υποστηρίζοντας ότι τα μοντέλα πέρα από μια συγκεκριμένη κλίμακα μπορούν να περιέχουν πιο προηγμένες γνώσεις. Εξετάζουμε εάν ο ισχυριμός αυτός είναι επίσης ευσταθής για για τον εμπλουτισμό γνώσης, πειραματιζόμενοι με μοντέλα μέχρι και 13 δισεκατομμυρίων παραμέτρων, όριο το οποίο αντιστοιχεί στο όριο των υπολογιστικών πόρων που διαθέτουμε, καθώς και LLMs τάξεων μεγέθους μεγαλύτερα (175 δισεκατομμύρια παραμέτρους) διαθέσιμων μέσω δημοσίων APIs, τα οποία και παρουσιάζουμε παρακάτω:

- GPT-3[9] (*text-davinci-003*) με 175 δισεκατομμύρια παραμέτρους & GPT-3.5-turbo
- GPT2-XL[64] με 1.5 δισεκατομμύριο παραμέτρους
- BLOOMZ-1.7B & BLOOMZ-3b [58]
- OPT-2.7B & OPT-6.7B [97]
- LLAMA-7B [83]
- Vicuna-7B & Vicuna-13B [99]
- Galactica-6.7B [80]

Εξαγωγή Κειμενικών Περιγραφών από τις Εικόνες για Ανάκτηση Κειμένου από Κείμενο

Ενσωματώνουμε τις μετρικές ενός μονοτροπικής ανάκτησης (αποκλειστικά γλωσσικής ή οπτικής), εκμεταλλευόμενοι σύγχρονους transformers για να μετατρέψουμε τις εικόνες i σε κειμενικές περιγραφές c_i. Πιο συγκεκριμένα τα μοντέλα (captioning transformers) που χρησιμοποίουμε για το σκοπό αυτό είναι:

- BLIP Captions [41] με το ViT-base[15] ως κωδικοποιητή
- BLIP Captions με το ViT-large ως κωδικοποιητή συμβολίζεται ως BLIP-L
- GiT [88] με το ViT-base ως κωδικοποιητή
- GiT με το ViT-large ως κωδικοποιητή, συμβολίζεται με GiT-L
- ViT-GPT2 [34], το οποίο χρησιμοποιεί το ViT-base ως κωδικοποιητή και το GPT-2[64] ως αποκωδικοποιητή

Για όλα τα παραπάνω μοντέλα, επιχειρούμε τόσο αναζήτηση δέσμης (beam search) πολυωνυμικής δειγματοληψίας με 5 δέσμες για να εξάγουμε k=10 περιγραφές ανά εικόνα i, καθώς και άπληστη αναζήτηση (greedy search). Συμβολίζουμε με c_i^k την k-οστή περιγραφή της εικόνας i, όπως προκύπτει από την αναζήτηση δέσμης (η άπληστη αναζήτηση επιστρέφει μια μόνο περιγραφή ανά εικόνα). Στην περίπτωση της αναζήτησης δέσμης, οι 10 περιγραφές υποβάλλονται επίσης σε μια επιπλέον επεξεργασία, καθώς ορισμένες από αυτές είναι πανομοιότυπες ή υποσυμβολοσειρές μεγαλύτερων.

Εξερευνούμε δύο επιλογές για την απόκτηση των διανυσματικών αναπαραστάσεων των περιγραφών c_i και των φράσεων t. Στην πρώτη περίπτωση, οι διανυσματικές αναπαραστάσεις εξάγονται με χρήση των ίδιων VL Transformers όπως στην πολυτροπική ανάκτηση. Στην δεύτερη περίπτωση, χρησιμοποιούμε ένα σύνολο από αμιγώς γλωσσικούς transformers προτάσεων, οι οποίοι έχουν εκπαιδευτεί περαιτέρω στην σημασιολογική ομοιότητα [68]. Αυτοί απαριθμούνται παρακάτω:

- distilroberta-base & xlm-r-distilroberta-paraphrase
- stsb-roberta-base, stsb-distilroberta-base & stsb-mpnet-base
- all-MiniLM-L6, all-MiniLM-L12 & all-mpnet-base
- multi-qa-distilbert-cos & multi-qa-MiniLM-L6-cos
- sentence-t5-base, sentence-t5-large, gtr-t5-base & gtr-t5-large

Έπειτα, και στις δύο περιπτώσεις, χρησιμοποιούμε ομοιότητα συνημιτόνου ή ευκλίδεια/manhattan απόσταση για τον υπολογισμό του $score(t, c_i^k)$, ανακτώντας έτσι το πιο όμοιο (με την πιο μεγάλη ομοιότητα ή αντίστοιχα μικρή απόσταση) διάνυσμα περιγραφής για κάθε διάνυσμα φράσης. Διεξάγουμε πειράματα τόσο χωρίς όσο και με τον εμπλουτισμό των φράσεων με χρήση των Γλωσσικών Μοντέλων.

Wikipedia & Wikidata για Ανάκτηση Εικόνας από Εικόνα

Η Ανάκτηση Εικόνας από Εικόνα είναι ένας ακόμα τρόπος προσέγγισης του προβλήματος Αποσαφήνισης Οπτικών Εννοιών μέσω πολυτροπικών αναπαραστάσεων. Για το σκοπό αυτό, ακολουθώντας την ιδέα του [11] χρησιμοποιούμε το Wikipedia API για να ανακτήσουμε αρχικά όλα τα σχετικά με τη δοθείσα φράση t άρθρα, και στη συνέχεια κρατάμε την πρωταρχική εικόνα i_w για καθένα από αυτά. Έπειτα, επεξεργαζόμαστε το ανακτημένο σύνολο εικόνων θεωρώντας ένα μέγιστο από $k = 10$ Wikipedia εικόνες για κάθε φράση t. Την ίδια διαδικασία επαναλαμβάουμε και για το Wikidata[85]. Για την εξαγωγή διανυσματικών αναπαραστάσεων για τις ανακτημένες εικόνες i_w, καθώς και για τις υποψήφιες εικόνες i, χρησιμοποιούμε τους ίδιους VL Transformers με την πολυτροπική ανάκτηση. Τέλος, αναζητούμε τις αναπαραστάσεις που βρίσκονται πιο κοντά μεταξύ τους στον διανυσματικό χώρο (χρησιμοποιώντας ομοιότητα συνημιτόνου ή ευκλείδια/manhattan απόσταση) σύμφωνα με το βαθμό ομοιότητας $score(i_w, i)$.

Μοντέλο Εκμάθησης Κατάταξης

Παρόμοια με το [11], υλοποιούμε ένα μοντέλο εκμάθησης κατάταξης που αξιοποιεί χαρακτηριστικά που εξάχθηκαν από τα δικά μας προαναφερθέντα πειράματα. Ως μοντέλο εκμάθησης κατάταξης χρησιμοποιούμε τον LGBM-Ranker με το lambdarank objective[1], υλοποιημένο πάνω στο gradient boosting framework[29].

Τα επιλεγμένα χαρακτηριστικά για το μοντέλο Εκμάθησης Κατάταξης αναπαριστούν σχέσεις μεταξύ κάθε δοθείσας φράσης και των υποψήφιων εικόνων, όπως αυτές εξάγονται κάθε φορά από κάθε μια από τις 4 προηγούμενες τεχνικές. Πιο συγκεκριμένα, τα ακόλουθα βήματα (α)-(ε) έχουν επιλεχθεί για τη δημιουργία χαρακτηριστικών που αντιστοιχούν στην περίπτωση αναφοράς:

(α) $score(t, i)$

(β) $max(score(t, i))$

(γ) $mean(score(t, i))$

(δ) *difference α-β*

(ε) *difference α-γ*

Με παρόμοιο τρόπο, τα βήματα (α)-(ε) επαναλαμβάνονται για τους βαθμούς: $score(t_e, i)$ (εμπλουτισμός φράσεων με χρήση LLMs), $score(t, c_i^k)$ (ανάκτηση Κειμένου από Κείμενο με εξαγωγή περιγραφών από τις εικόνες), $score(t_e, c_i^k)$ (ανάκτηση Κειμένου από Κείμενο με εξαγωγή περιγραφών από τις εικόνες και εμπλουτισμό φράσεων μέσω LLMs) και $score(i_w, i)$ (ανάκτηση Εικόνας από Εικόνα). Εκπαιδεύουμε το μοντέλο εκμάθησης κατάταξης σε αρκετούς συνδυασμούς αυτών των χαρακτηριστικών. Επιπλέον, μεταξύ αυτών των συνδυασμών επιχειρούμε να χρησιμοποιούμε ομοιότητα συνημιτόνου και ευκλείδια/manhattan απόσταση ως μετρικές ομοιότητας, ενώ αξιολογούμε και τη συνεισφορά του συντελεστή ποινής $p(i)$ (εξίσωση 1.2.3) τόσο στην βασική περίπτωση αναφοράς όσο και στην πολυτροπική ανάκτηση με εμπλουτισμό του εννοιλογικού πλαισίου μέσω LLMs. Προκειμένου, να ενισχύσουμε περαιτέρω την απόδοση του μοντέλου, επιχειρούμε επίσης να συνδυάσουμε χαρακτηριστικά των εμπλουτισμένων φράσεων προερχόμενων από διαφορετικά LLMs ή προτροπές.

Πρόβλημα Ερώτησης-Απάντησης και Αλυσιδωτός Συλλογισμός

Μετατρέπουμε το πρόβλημα Αποσαφήνισης Οπτικών Εννοιών σε ένα πρόβλημα Ερώτησης-Απάντησης, αντιστοιχίζοντας τις κειμενικές εκφράσεις t σε ερωτήσεις Q σύμφωνα με βάση χειροκίνητα κατασκευασμένα πρότυπα προτροπών (Πίνακας 1.3).

Τα πειράματά μας περιλαμβάνουν τόσο ρύθμιση μηδενικής βολής, όσο και ρύθμιση πολλαπλών βολών. Και στα δύο σενάρια, τα LLMs που επερωτώνται είναι το Vicuna-13B[99] και το GPT-3.5-turbo με 175 δισεκατομμύρια παραμέτρους. Καθώς τα LLMs προς το παρόν μπορούν να διαχειριστούν μόνο κειμενικές και όχι και οπτικές εισόδους, πρέπει να μεταβούμε σε αποκλειστικά κειμενικές αναπαραστάσεις τόσο για τις εικόνες i όσο και για τις φράσεις t. Για το λόγο αυτό ενσωματώνουμε και πάλι τεχνικές εξαγωγής περιγραφών από εικόνες για να

[1] LGBMRanker docs

Prompt name	Prompt template
think (greedy)	"Q: What is the most appropriate caption for the <context>? Answer choices: (A) <caption for image 1> (B) <caption for image 2> ... A: Let's think step by step. "
think (beam)	"Q: What is the most appropriate group of captions for the <context>? Answer choices: (A) <captions for image 1 (separated with comma)> (B) <captions for image 2> ... A: Let's think step by step. "
CoT	"<think_prompt> <response of llm with think prompt> Therefore, among A through J, the answer is "
no_CoT (greedy)	"Q: What is the most appropriate caption for the <context>? Answer choices: (A) <caption for image 1> (B) <caption for image 2> ... A: "
no_CoT (beam)	"Q: What is the most appropriate group of captions for the <context>? Answer choices: (A) <captions for image 1> (B) <captions for image 2> ... A: "
choose no_CoT (greedy)	You have ten images, (A) to (J), which are given to you in the form of captions.(A) <caption for image 1>. . . (J) <caption for image 10> You should choose the image, and therefore the caption that could better represent the <phrase>. What image do you choose?
choose no_CoT (beam)	You have ten images, (A) to (J), which are given to you in the form of captions.(A) <captions for image 1 (separated with comma)>. . . (J) <captions for image 10 (separated with comma)> You should choose the image, and therefore the set of captions that could better represent the <phrase>. What image do you choose?
choose CoT (greedy)	You have ten images, (A) to (J), which are given to you in the form of captions. (A) <caption for image 1> . . . (J) <caption for image 10> You should choose the image, and therefore the caption that could better represent the <phrase>. Use the following format: Question: What image do you choose? Thought: you should always think about what you choose. Result: the result of your thought Final Answer: the image that you choose. Begin! Question: What image do you choose?
choose CoT (beam)	You have ten images, (A) to (J), which are given to you in the form of a set of captions. (A) captions for image 1 (separated with comma) . . . (J) captions for image 10 (separated with comma) You should choose the image, and therefore the set of captions that could better represent the <phrase>. Use the following format: Question: What image do you choose? Thought: you should always think about what you choose Result: the result of your thought Final Answer: the image that you choose Begin! Question: What image do you choose?

Table 1.3: Προτροπές Ερώτησης-Απάντησης ή χωρίς Αλυσιδωτό Συλλογισμό.

επιτύχουμε το μετασχηματισμό αυτό, παρέχοντας περιγραφές c_i για κάθε υποψήφια εικόνα i. Τα μοντέλα που επιλέγουμε για την εξαγωγή των περιγραφών από τις εικόνες είναι τα GiT-L[88] BLIP-L[41]., τα οποία και τα δύο αποτελούνται από το ViT-large[15] ως κωδικοποιητή, και ViT-GPT2[34], το οποίο χρησιμοποιεί το ViT-base ως κωδικοποιητή και το GPT-2[64] ως αποκωδικοποιητή.

Οι συλλογιστικές ικανότητες των LLMs μπορούν να ξεκλειδωθούν μεσώ του λεγόμενου Αλυσιδωτού Συλλογισμού [30, 89], κατά τον οποίο τα LLMs ζητείται να παράξουν μια σειρά από ενδιάμεσα συλλογιστικά βήματα που οδηγεί λογικά στην τελική τους απάντηση. Παρότι ο Αλυσιδωτός Συλλογισμός έχει χρησιμοποιηθεί ως επι το πλείστον σε προβλήματα με πολλαπλά βήματα συλλογισμού, μπορεί επίσης να παρέχει κατανοητές για εμάς *επεξηγήσεις* σχετικά με την επιλογή της καταλληλότερης υποψήφιας εικόνας i για κάθε φράση t. Για το σκοπό αυτό τα 5 πρώτα πρότυπα του πίνακα 1.3 που έχουν υιοθετηθεί από το [30], αποτελούνται από μια προτροπή "συλλογισμού" (*"Let's think step by step/think* προτροπές από τον Πίνακα 1.3) ανακτά το συλλογιστικό μονοπάτι που βρίσκεται αποθηκευμένο στο LLM, και έπειτα μια προτροπή "απάντησης" (*"Therefore, among A through J, the answer is "* προτροπή του Πίνακα 1.3) που επιστρέφει την τελική απάντηση σε μια επιθυμητή μορφή. Οι υπόλοιπες προτροπές (προτροπές με το όνομα "choose") είναι εμπνευσμένες από τις προτροπές του LangChain[35].

Προτροπές μηδενικής βολής. Σε ρύθμιση μηδενικής βολής, εισάγουμε μια επιλεγμένη προτροπή από τον Πίνακα 1.3 στο LLM (είτε το Vicuna-13B είτε το GPT-3.5-turbo), το οποίο και παράγει την απάντηση **A** μαζί με μια επεξήγηση της επιλογής αυτής. Η παραγόμενη απάντηση **A** μπορεί να είναι είτε μια από τις περιγραφές των υποψηφίων εικόνων A-J είτε μια δήλωση ότι δε μπορεί να οριστεί απάντηση: σε κάθε περίπτωση, η παραγόμενη απάντηση **A** συγκρίνεται την περιγραφή της εικόνας που πραγματικά αντιστοιχεί στη δοθείσα φράση για να καταλήξουμε εάν η στρατηγική προτροπής μηδενικής βολής οδηγεί σε επιτυχία ή αποτυχία.

Προτροπές πολλαπλών βολών. Επιπρόσθετα, πειραματιζόμαστε με τη ρύθμιση προτροπής πολλαπλών βολών στη θέση της ρύθμισης μηδενικής βολής που περιγράψαμε προηγουμένως. Στην περίπτωση αυτή επιλέγουμε k *no_ CoT* προτροπές (Πίνακας 4.3), συνοδευόμενες μαζί με την αληθινή απάντησή τους **A**, σχηματίζοντας με αυτό τον τρόπο *QA in-context παραδείγματα*. Ο αριθμός των in-context παραδειγμάτων k αποφασίζεται από το χρήστη. Σχεδιάζουμε 3 διαφορετικούς τρόπους επιλογής των k in-context παραδειγμάτων. Στη βασική περίπτωση αναφοράς, τα k παραδείγματα επιλέγονται τυχαία από το σύνολο δεδομένων εκπαίδευσης. Ωστόσο, δεδομένου ότι η συνάφεια των επιλεγμένων παραδειγμάτων σε σχέση με το δοθέν δείγμα είναι σημαντική[46] ,καθώς και η σειρά των παραδειγμάτων[53], σχεδιάζουμε δύο τεχνικές επιλογής παραδειγμάτων βασισμένες στην ομοιότητα, τις οποίες καλούμε *top* και *inverse-top*. Και οι δύο τεχνικές επιλογής χρησιμοποιούν τις διανυσματικές αναπαραστάσεις των φράσεων t που συμπεριλαμβάνονται στις QA προτροπές, και προήλθαν με χρήση του ALIGN[25] VL Transformer. Τα k κοντινότερα διανύσματα με το διάνυσμα της δοθείσας φράσης t ανακτόνται αποτελεσματικά με τη βοήθεια της ομοιότητας συνημιτόνου και της βιβλιοθήκης FAISS[27]. Στη συνέχεια, η τεχνική ονόματι *top* τοποθετεί το πιο σχετικό παράδειγμα πρώτο, ακολουθούμενο από το 2ο πιο σχετικό, μέχρι και το k-οστό πιο σχετικό παράδειγμα στην k-οστή θέση. Από την άλλη πλευρά, η *inv-top* τεχνική αντιστρέφει τη σειρά αυτή τοποθετώντας το πιο σχετικό παράδειγμα στην k-οστή θέση. Η επιλογή του k ποικίλλει μεταξύ των διαφόρων πειραμάτων, ανάλογα με το μήκος της προτροπής και τους περιορισμούς λόγω των διαθέσιμων υπολογιστικών πόρων, και ορίζεται στον Πίνακα 1.4.

Captioner Model	**Captioner Strategy**	k
GiT-L / BLIP-L / ViT-GPT2	greedy	5
GiT-L	beam	2
BLIP-L	beam	1
ViT-GPT2	beam	2

Table 1.4: Επιλογή της παραμέτρου k

1.3 Πειραματικό Μέρος

Στην ενότητα αυτή παρουσιάζουμε τα αποτελέσματα εκτεταμένων πειραμάτων που διεξαγάγαμε στα πλαίσια των 6 τεχνικών που δοκιμάσαμε προκειμένου να διερευνήσουμε πλήρως το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών. Για την αξιολόγηση των αποτελεσμάτων βασιζόμαστε στις ακόλουθες μετρικές: *Accuracy* και *Mean Reciprocal Rank (MRR)*.

1.3.1 LLMs για τον εμπλουτισμό του εννοιολογικού πλαισίου

Στους Πίνακες 1.5 και 1.6 παρουσιάζουμε τα αποτελέσματα που προκύπτουν από την πολυτροπική ανάκτηση με χρήση VL Transformers, με ή χωρίς την ενσωμάτωση του συντελεστή ποινής $p(i)$ αντίστοιχα. Τα *Baselines* αναφέρονται στις περιπτώσεις αναφοράς, δηλαδή πολυτροπικής ανάκτησης μέσω VL Transformers χωρίς τον εμπλουτισμό των φράσεων από τα LLMs.

Table 1.5: Αποτελέσματα πολυτροπικής ανάκτησης με την ενσωμάτωση του συντελεστή ποινής $p(i)$. Τα σημειωμένα με ροζ αντιστοιχούν στα καλύτερα αποτελέσματα ανά μετρική, ενώ τα τα σημειωμένα με **έντονη** γραμματοσειρά στα καλύτερα αποτελέσματα για κάθε LLM.

		CLIP		**CLIP-L**		**CLIP$_{LAION}$**		**ALIGN**		**BLIP$_C$**		**BLIP-L$_C$**		**BLIP$_F$**		**BLIP-L$_F$**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
Baseline		63.28	76.27	62.85	76.24	71.06	81.50	68.90	80.00	60.91	74.33	64.58	77.51	60.48	73.87	69.76	80.42
GPT2-XL	exact	53.88	69.51	56.32	71.12	57.45	72.53	53.35	69.57	47.52	63.67	47.95	64.49	41.90	59.79	50.54	66.96
	what_is	61.22	74.89	61.44	75.83	64.58	77.61	63.93	76.33	57.02	70.42	57.67	71.78	51.62	67.78	61.34	74.42
	meaning_of	60.82	75.00	**65.58**	77.55	65.23	**78.04**	64.15	76.65	58.75	72.64	59.18	73.27	52.92	68.27	63.07	75.96
	describe	57.58	72.38	60.82	74.82	65.01	77.17	58.10	72.67	54.43	68.78	53.78	69.38	47.73	65.08	57.24	71.80
	write_description	58.90	72.87	62.42	75.63	63.28	77.11	60.26	74.37	50.76	67.19	54.21	69.57	48.60	65.12	60.04	73.43
	to_describe	59.74	73.86	62.34	75.55	63.50	76.40	59.40	73.46	52.92	67.80	53.35	68.86	48.16	64.47	55.94	70.67
	could_describe	57.08	71.87	59.26	73.23	65.01	76.87	58.10	72.49	52.05	68.22	55.08	70.18	46.87	64.44	57.02	72.03
BLOOMZ-1b7	exact	61.44	74.50	64.92	77.42	66.52	78.50	65.87	77.60	64.58	76.28	65.66	77.13	59.18	72.70	67.39	78.67
	what_is	63.71	76.41	66.74	78.92	69.98	80.67	65.44	77.65	63.28	75.95	65.23	77.68	58.32	72.06	66.52	78.30
	meaning_of	62.63	76.38	65.01	78.17	69.33	80.69	66.74	78.27	63.50	76.44	65.44	78.29	58.53	72.50	68.25	79.74
	describe	64.72	77.21	64.07	77.47	**69.98**	**81.40**	68.90	80.28	61.34	75.61	64.36	77.45	58.53	72.41	66.74	78.81
	write_description	62.61	76.00	65.93	78.07	68.90	80.13	68.68	79.70	60.04	73.52	64.58	77.13	57.02	71.19	67.17	78.66
	to_describe	63.20	76.36	66.45	78.20	69.11	80.13	70.41	80.70	64.15	76.77	66.95	78.66	57.88	72.20	68.25	79.81
	could_describe	64.86	77.13	63.99	77.10	66.95	79.11	69.33	79.96	62.42	75.65	65.87	78.17	58.10	72.47	68.47	79.89
OPT-2.7B	exact	62.85	76.00	62.85	75.93	**71.06**	**81.46**	68.68	79.89	61.12	74.46	64.58	77.41	60.26	73.73	69.76	80.36
	what_is	60.98	74.85	66.30	78.10	66.95	78.99	63.28	75.95	60.91	74.43	66.31	77.86	57.24	71.15	67.70	78.58
	meaning_of	62.15	75.60	65.25	77.45	69.11	80.56	65.66	77.54	61.99	75.35	63.93	76.88	58.32	71.65	65.44	77.69
	describe	61.05	74.75	66.08	78.14	68.03	79.89	64.79	77.62	61.77	74.73	66.31	77.57	57.67	71.48	68.03	79.03
	write_description	56.44	71.82	63.37	76.52	68.03	80.34	65.44	77.68	61.99	74.65	62.42	75.29	54.64	69.49	64.79	76.70
	to_describe	62.53	76.07	65.17	78.02	68.03	79.77	64.15	76.92	63.07	75.22	61.99	74.91	57.02	70.74	65.44	77.33
	could_describe	59.83	73.70	56.99	72.31	67.39	79.06	68.68	80.41	65.23	77.44	66.09	77.44	57.45	71.88	68.25	79.57
BLOOMZ-3b	exact	61.26	74.59	62.99	76.18	67.82	79.31	66.52	78.36	60.48	73.13	63.28	76.00	57.02	71.23	65.66	77.49
	what_is	64.36	76.82	68.25	79.82	**71.92**	**81.78**	67.39	78.72	61.34	74.94	66.95	78.47	59.61	73.35	68.47	79.58
	meaning_of	65.58	77.96	67.32	78.76	70.63	81.47	68.47	79.14	63.71	76.52	66.31	78.55	59.40	73.60	68.03	79.26
	describe	62.01	75.38	65.28	78.07	70.84	81.11	66.09	78.60	62.85	75.65	67.39	78.71	57.24	71.72	67.82	79.20
	write_description	65.73	77.16	66.81	79.16	68.90	79.86	68.68	80.07	62.85	76.23	64.36	77.61	58.53	72.21	67.39	79.17
	to_describe	64.79	77.49	68.03	79.21	68.68	79.94	70.19	80.66	65.23	77.45	66.31	78.16	60.69	73.92	68.68	79.81
	could_describe	65.23	77.40	65.66	78.09	69.33	80.80	69.11	79.87	65.23	77.38	67.17	78.75	61.34	73.87	69.33	80.18
OPT-6.7B	exact	62.63	75.84	62.20	75.54	70.63	81.15	67.82	79.24	60.91	74.23	64.79	77.58	59.83	73.40	69.11	79.94
	what_is	61.79	75.70	64.63	77.68	68.03	79.41	64.79	77.23	61.77	75.01	63.07	76.16	57.88	71.79	65.87	77.77
	meaning_of	62.17	75.84	63.61	77.19	69.55	80.79	66.74	78.47	63.28	75.93	65.44	77.43	59.83	72.96	68.03	78.75
	describe	64.43	76.91	65.73	78.24	**70.84**	**81.19**	65.23	77.89	61.12	74.67	63.93	77.07	56.16	71.30	66.09	78.38
	write_description	60.38	74.46	60.61	75.19	69.76	80.90	64.58	77.04	57.02	71.13	61.34	74.24	53.13	68.91	64.58	76.70
	to_describe	63.98	76.77	61.14	75.49	66.31	78.74	68.03	78.95	61.77	74.73	60.26	74.38	55.51	70.23	65.87	77.70

	could_describe	58.59	73.21	61.45	74.60	66.31	78.27	65.44	77.76	63.07	75.34	64.36	76.09	58.32	71.86	65.23	77.28
Galactica-6.7B	exact	49.66	66.59	56.24	71.44	55.72	71.25	52.48	68.45	43.21	60.82	50.32	66.95	41.90	59.68	54.00	70.23
	what_is	60.13	74.40	62.78	76.11	65.66	77.73	62.63	75.50	57.88	72.62	61.12	74.59	53.56	68.73	64.79	76.81
	meaning_of	60.09	74.69	62.20	74.80	**68.68**	**80.24**	60.32	75.04	61.56	74.57	61.34	74.99	52.27	68.02	61.99	75.49
	describe	59.61	74.04	60.48	75.13	63.93	77.78	62.20	75.80	55.51	70.74	56.59	71.25	52.92	68.13	58.96	73.49
	write_description	58.54	73.10	60.75	74.93	65.01	77.73	57.88	72.64	55.72	70.27	56.16	70.83	49.24	64.97	60.91	74.25
	to_describe	57.78	73.32	59.75	74.52	63.28	76.21	56.16	71.45	55.29	70.42	55.72	70.92	51.19	66.89	60.26	74.05
	could_describe	62.45	75.98	63.97	76.62	67.60	79.44	58.10	72.89	55.51	70.00	56.59	71.38	49.24	66.15	60.48	74.29
LLAMA-7b	exact	55.08	70.70	57.45	72.23	62.20	76.04	60.48	74.10	52.92	68.76	54.00	69.51	48.81	65.54	58.10	72.33
	what_is	64.92	77.26	67.10	78.93	68.90	79.07	66.55	80.48	68.03	68.03	66.95	78.48	58.75	73.29	70.84	80.84
	meaning_of	61.76	76.12	62.90	76.53	67.82	80.11	65.66	78.49	60.26	74.40	62.42	75.71	54.00	69.84	65.87	78.19
	describe	64.21	76.80	66.81	79.04	**72.14**	**82.02**	69.76	80.34	64.79	76.58	66.09	77.99	58.53	72.44	69.11	80.15
	write_description	62.28	76.00	63.84	76.77	71.27	81.62	69.76	80.59	67.17	77.86	65.87	77.80	59.40	73.34	68.47	79.55
	to_describe	65.54	77.80	66.32	79.24	68.68	79.93	60.91	74.71	61.34	74.32	61.77	74.77	54.00	69.65	63.07	75.97
	could_describe	60.34	74.16	65.02	76.30	64.15	76.40	61.56	74.36	54.00	68.67	57.45	71.65	52.27	67.00	61.34	74.50
VICUNA-7b	exact	58.10	72.46	59.18	73.62	61.99	74.95	64.58	77.20	55.94	70.98	59.83	73.52	53.56	69.18	62.20	75.68
	what_is	66.02	77.82	69.05	80.16	68.25	79.76	69.76	80.49	66.74	78.00	69.33	80.00	61.34	74.61	70.84	81.00
	meaning_of	65.14	77.85	67.76	79.49	69.76	80.25	70.41	81.13	65.87	77.84	66.31	78.43	62.20	74.73	68.03	79.40
	describe	65.44	78.01	68.47	79.97	70.41	81.18	72.79	82.26	68.25	79.09	70.41	80.85	62.20	75.09	**73.43**	**82.45**
	write_description	63.83	76.61	67.32	79.40	68.47	80.24	72.14	81.17	65.23	77.36	67.39	79.05	58.53	73.03	70.63	80.73
	to_describe	64.47	77.04	68.20	79.61	67.82	79.40	69.98	81.07	63.93	76.32	64.15	76.84	57.88	72.27	68.03	79.26
	could_describe	65.44	77.46	65.23	77.83	68.25	79.69	72.14	82.16	63.93	76.32	65.01	76.84	57.67	72.27	68.90	79.26
VICUNA-13b	exact	60.61	74.49	62.12	75.60	64.15	77.10	67.60	79.48	60.69	73.98	65.01	77.46	54.86	69.91	66.74	78.74
	what_is	65.44	77.63	69.98	80.58	70.84	81.05	72.14	81.69	67.82	79.01	69.11	79.87	59.18	73.48	**73.43**	**82.91**
	meaning_of	65.43	78.25	68.26	79.83	70.63	81.03	70.63	81.33	67.82	78.70	69.76	80.66	61.99	75.08	71.71	81.80
	describe	64.36	77.01	63.28	76.85	66.74	78.64	69.98	80.70	61.12	74.73	66.95	78.57	56.16	71.10	67.17	79.11
	write_description	65.35	77.71	66.45	78.70	70.41	81.60	71.71	82.01	66.31	77.95	67.39	78.74	61.34	74.70	69.98	80.87
	to_describe	65.86	78.12	69.58	80.49	70.19	80.88	71.27	81.32	63.71	76.04	65.01	77.12	58.32	72.17	68.68	79.42
	could_describe	63.08	76.10	63.30	76.87	68.03	79.26	66.74	78.98	63.28	75.26	63.93	76.57	55.94	69.90	66.74	78.28
GPT-3.5	exact	58.86	72.09	60.18	72.73	64.36	75.38	62.42	74.43	57.02	70.78	59.18	72.32	52.92	67.40	63.07	74.65
	what_is	66.52	78.81	69.35	80.51	70.63	81.46	70.41	81.42	67.60	78.56	68.47	79.67	60.91	74.30	71.71	82.02
	meaning_of	67.76	79.76	69.06	80.55	**73.65**	**82.71**	70.41	81.38	66.52	78.59	66.52	79.16	58.53	73.31	69.98	81.46
	describe	67.32	78.95	69.28	80.31	73.22	82.50	73.22	82.73	69.33	79.90	70.41	80.80	59.83	73.65	70.63	81.29
GPT-3	exact	61.98	74.90	64.07	76.58	68.03	78.41	66.52	78.37	60.48	73.99	64.15	76.58	59.61	72.91	65.23	77.06
	what_is	67.92	79.27	70.73	81.57	72.35	82.19	71.71	82.27	68.25	78.93	68.90	79.91	60.48	74.24	69.11	80.25
	meaning_of	68.07	80.08	69.84	81.56	73.65	83.52	**74.95**	**84.09**	66.74	78.37	71.71	81.55	62.63	75.55	72.35	82.28
	describe	68.25	79.40	68.72	80.26	70.63	81.05	72.57	82.52	64.58	76.75	68.25	79.35	61.34	74.03	69.33	80.47

Table 1.6: Αποτελέσματα πολυτροπικής ανάκτησης χωρίς την ενσωμάτωση του συντελεστή ποινής $p(i)$. Τα σημειωμένα με **ροζ** αντιστοιχούν στα καλύτερα αποτελέσματα ανά μετρική, ενώ τα τα σημειωμένα με **έντονη** γραμματοσειρά στα καλύτερα αποτελέσματα για κάθε LLM.

		CLIP		**CLIP-L**		**CLIP$_{LAION}$**		**ALIGN**		**BLIP$_C$**		**BLIP-L$_C$**		**BLIP$_F$**		**BLIP-L$_F$**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
Baseline		59.18	72.94	60.69	74.42	67.82	79.50	65.66	77.48	57.24	60.91	61.34	64.58	57.67	60.48	65.01	69.76
GPT2-XL	exact	49.45	66.24	53.66	69.09	54.64	70.54	51.19	67.22	44.28	61.19	45.57	62.60	35.85	55.50	47.52	64.43
	what_is	58.61	72.59	58.61	73.62	63.28	76.56	60.91	74.25	54.21	68.22	53.56	69.03	46.00	64.08	55.94	70.71
	meaning_of	58.44	72.97	62.55	75.60	**64.58**	**77.08**	61.56	72.54	55.08	70.24	54.64	70.76	50.32	66.74	57.02	72.54
	describe	54.76	70.24	56.49	72.14	62.20	75.37	55.94	70.55	50.97	66.20	50.11	66.62	44.71	62.21	55.51	70.07
	write_description	54.51	69.99	59.34	73.54	62.20	75.99	57.45	71.77	47.52	64.76	48.16	65.10	45.36	62.79	56.16	70.50
	to_describe	54.11	70.00	59.52	73.81	61.77	74.84	57.02	71.67	50.32	65.60	49.89	66.08	42.33	60.39	53.56	68.46
	could_describe	51.85	68.28	55.77	70.84	62.42	74.97	54.64	69.73	48.81	65.20	51.19	66.97	42.76	61.44	53.13	68.74

BLOOMZ-1b7	exact	58.82	72.23	61.66	75.05	63.71	76.53	63.28	75.56	59.18	72.96	62.85	74.99	56.37	70.60	63.50	76.20
	what_is	62.42	75.30	65.01	77.33	67.60	79.04	63.07	75.58	59.40	73.50	62.85	76.00	56.16	70.54	65.23	77.26
	meaning_of	58.75	73.78	64.15	77.03	66.52	78.70	64.36	76.44	60.48	74.34	61.99	76.13	56.16	70.61	65.01	77.89
	describe	60.82	74.68	62.99	76.05	67.60	79.54	66.52	78.51	59.83	74.59	62.63	76.47	53.56	69.84	63.71	77.06
	write_description	59.29	73.49	64.82	77.08	67.82	78.85	66.31	77.86	57.67	71.61	61.34	74.76	52.48	68.47	64.36	76.72
	to_describe	59.96	73.99	63.64	76.37	**67.82**	**78.96**	68.03	78.94	61.56	74.87	64.79	77.09	55.94	70.84	66.31	78.34
	could_describe	60.74	74.63	60.52	74.64	66.95	78.72	67.17	78.20	60.69	74.14	63.50	76.46	55.72	70.94	66.31	78.54
OPT-2.7B	exact	58.96	72.77	60.26	74.15	**67.82**	**79.47**	65.66	77.48	57.45	72.19	61.12	75.77	57.24	71.68	65.01	77.90
	what_is	58.31	72.91	62.75	75.47	65.44	77.60	61.12	73.94	59.83	73.13	61.12	74.54	53.35	68.71	63.50	76.22
	meaning_of	58.19	72.97	62.99	75.79	67.60	79.28	64.58	76.48	59.18	73.38	60.26	74.70	54.86	69.43	62.42	75.86
	describe	59.08	72.95	63.89	76.31	65.87	78.09	62.20	75.80	59.83	73.28	62.20	75.17	54.43	69.86	63.28	76.28
	write_description	54.95	70.44	59.16	73.87	66.31	78.85	63.50	76.13	56.80	71.49	59.83	73.35	50.76	66.97	63.07	75.67
	to_describe	61.21	74/61	62.27	75.92	65.01	77.78	60.69	74.45	58.75	72.16	58.75	72.69	53.56	68.09	61.56	75.02
	could_describe	56.77	71.43	54.37	69.93	66.74	78.42	66.74	78.48	61.56	75.25	62.42	75.56	54.43	69.98	65.01	77.61
BLOOMZ-3b	exact	56.93	71.53	59.52	73.78	64.15	76.98	63.93	76.15	58.10	71.77	59.61	74.06	54.86	69.66	61.12	74.99
	what_is	62.20	75.39	65.66	77.88	**69.11**	**80.03**	62.85	75.51	61.34	74.35	65.01	77.32	57.24	71.85	68.03	79.12
	meaning_of	61.69	75.51	64.94	77.17	68.25	79.73	66.31	77.62	61.77	74.92	62.42	76.27	57.02	71.79	65.23	77.21
	describe	60.04	73.83	62.88	76.11	68.68	79.47	63.50	76.35	60.48	73.87	62.85	76.06	54.86	70.48	65.66	77.64
	write_description	62.26	74.85	63.56	77.02	67.39	78.64	65.87	77.75	60.91	74.43	62.42	75.68	55.51	70.35	65.01	77.40
	to_describe	61.34	74.87	64.15	76.57	65.66	77.92	66.95	78.53	62.20	75.50	63.28	76.23	56.80	71.73	65.23	77.70
	could_describe	61.56	74.96	61.77	75.50	68.03	79.62	66.95	78.10	62.85	75.85	65.23	77.25	57.24	71.49	67.82	79.06
OPT-6.7B	exact	58.75	72.63	59.61	73.86	67.60	79.33	64.15	76.57	57.24	71.96	61.12	75.83	56.80	71.40	64.79	77.66
	what_is	60.48	74.10	62.45	75.89	66.95	78.61	61.77	75.18	57.88	72.27	61.77	74.89	52.92	68.83	61.99	75.23
	meaning_of	59.28	73.77	62.17	76.04	68.03	79.61	63.71	76.31	52.92	74.37	61.99	75.47	55.94	70.67	65.01	77.27
	describe	60.74	74.28	63.12	76.19	**69.55**	**79.99**	63.28	76.26	59.40	73.03	58.96	73.86	52.92	69.18	62.63	76.13
	write_description	56.60	71.61	57.31	72.84	67.39	79.14	62.85	75.52	55.08	69.45	58.32	71.93	49.68	66.35	60.48	74.51
	to_describe	60.90	74.51	58.29	73.30	65.87	78.12	65.87	77.33	60.26	73.65	57.24	72.50	54.21	69.13	60.69	74.78
	could_describe	55.29	70.73	59.25	72.95	65.44	77.56	62.20	75.18	60.04	73.27	59.83	73.50	52.70	68.52	60.26	74.46
Galactica-6.7B	exact	45.35	63.57	53.97	69.58	54.21	69.84	49.89	66.30	38.66	57.14	47.08	64.40	38.01	56.83	50.76	67.94
	what_is	56.61	71.74	59.91	73.82	64.79	76.92	60.91	74.12	55.72	70.32	58.75	72.82	52.27	67.77	61.77	74.84
	meaning_of	56.69	72.21	56.24	72.63	**65.44**	**78.46**	58.96	72.64	55.72	70.79	56.37	71.65	50.11	66.57	58.32	72.90
	describe	56.80	71.45	58.53	73.31	61.56	75.76	59.61	73.85	53.56	69.03	51.40	67.91	50.97	66.30	54.43	70.60
	write_description	55.65	71.09	57.21	72.40	62.85	76.12	54.64	70.47	50.11	66.71	51.62	67.61	43.84	61.69	57.88	72.33
	to_describe	54.32	70.73	56.30	72.15	60.69	74.35	52.05	68.27	51.40	67.87	50.97	67.77	46.87	63.77	54.21	70.28
	could_describe	60.26	74.07	62.23	75.20	65.23	77.64	54.86	69.99	54.21	68.60	54.21	69.24	45.57	63.44	58.32	72.09
LLAMA-7b	exact	53.13	68.98	56.16	70.76	60.91	74.66	56.80	71.06	52.05	67.33	50.54	66.81	47.86	64.17	54.86	69.66
	what_is	63.18	75.98	64.71	77.31	67.60	79.19	66.31	78.23	64.36	76.30	64.36	77.00	56.37	71.90	68.47	79.43
	meaning_of	57.24	72.84	60.41	74.79	62.85	76.63	66.09	78.60	58.75	72.70	58.53	73.12	52.70	68.03	62.42	76.07
	describe	61.39	74.97	63.77	76.99	**70.41**	**81.20**	66.74	78.84	64.15	76.25	63.50	76.40	58.10	72.58	68.25	79.36
	write_description	58.71	73.22	61.61	74.97	70.19	80.72	66.09	78.63	64.15	76.09	62.20	75.74	56.16	71.16	67.17	78.73
	to_describe	60.62	74.88	63.99	77.52	66.52	78.42	58.96	73.13	59.18	72.83	58.32	72.59	51.62	67.48	59.83	74.11
	could_describe	57.39	72.11	61.58	74.15	63.28	75.49	59.61	72.67	52.02	66.83	52.92	68.56	49.03	64.34	58.32	72.55
VICUNA-7b	exact	54.86	70.07	57.45	71.92	61.99	75.30	58.96	72.68	52.70	68.00	56.16	71.28	50.32	66.75	60.04	74.04
	what_is	64.94	76.70	65.58	77.93	68.47	79.28	64.58	77.37	64.79	76.50	66.52	78.11	57.88	72.77	67.17	78.98
	meaning_of	61.66	75.77	65.80	78.11	69.11	80.13	68.25	79.18	63.28	75.79	63.07	75.96	59.18	72.88	64.36	77.09
	describe	63.07	76.16	65.87	77.91	71.06	81.17	68.25	79.74	66.09	77.78	68.90	79.63	60.69	73.97	70.19	80.69
	write_description	61.22	74.95	66.01	78.02	68.90	80.11	66.74	78.81	63.28	76.12	63.71	76.43	55.29	70.78	69.11	79.45
	to_describe	62.06	75.24	65.79	77.99	67.60	79.55	65.44	77.85	62.85	75.44	62.63	75.72	54.86	70.73	63.28	76.48
	could_describe	62.20	75.40	63.07	76.10	**71.06**	**81.24**	65.66	77.86	60.69	73.94	61.56	74.99	57.24	71.41	65.87	77.58
VICUNA-13b	exact	56.93	71.67	58.44	73.10	65.23	77.73	61.56	74.91	58.60	71.48	60.69	74.43	51.19	66.95	63.28	76.68
	what_is	62.85	75.51	66.74	78.55	**70.63**	**80.71**	68.68	79.42	65.44	77.22	68.47	79.14	58.96	73.22	68.90	80.27
	meaning_of	63.70	76.64	65.87	78.38	68.90	80.18	67.60	78.99	65.66	77.62	66.31	78.31	58.96	73.19	68.03	79.64
	describe	60.48	75.75	60.69	74.88	67.82	79.14	64.58	77.12	57.67	72.47	62.20	75.75	52.27	68.72	63.50	76.66
	write_description	62.72	75.84	64.47	77.04	69.55	80.57	68.03	79.55	64.15	76.37	65.23	77.18	57.02	72.32	66.31	78.85

	to_describe	63.89	76.59	66.74	78.50	68.25	79.41	68.68	79.50	61.99	74.95	62.63	75.40	56.59	71.33	65.23	77.50
	could_describe	60.66	74.35	61.32	75.06	64.58	77.33	65.01	76.99	59.40	72.74	61.77	74.81	53.78	69.16	65.44	76.81
GPT-3.5	exact	56.89	69.85	57.11	70.36	62.20	73.38	60.48	72.15	54.43	68.33	56.80	70.42	51.50	65.68	58.32	71.11
	what_is	65.00	77.11	65.87	78.11	69.55	80.51	67.82	79.52	64.15	75.91	65.87	77.78	58.10	72.32	68.03	79.36
	meaning_of	65.14	77.61	67.10	79.07	**72.57**	**82.05**	68.47	79.87	63.93	77.05	65.66	78.33	63.93	72.23	68.25	80.17
	describe	65.80	77.26	66.67	78.42	72.57	81.78	70.84	81.16	65.44	77.57	69.11	80.20	58.96	72.66	67.60	79.47
GPT-3	exact	59.88	73.38	61.68	74.91	66.74	77.22	64.79	76.27	58.96	71.92	60.48	74.02	55.72	70.34	62.42	75.04
	what_is	66.51	77.62	68.15	79.38	71.06	81.12	69.55	80.22	63.28	75.56	65.01	77.40	56.59	71.54	67.82	79.03
	meaning_of	66.52	78.32	68.96	80.26	**73.00**	**82.89**	72.57	82.29	65.87	77.56	69.55	80.26	60.26	74.26	70.41	81.09
	describe	67.30	78.50	68.25	79.81	69.55	80.15	71.27	81.21	63.93	75.81	66.31	77.74	58.96	72.62	67.17	78.93

Παρατηρούμε ότι ο εμπλουτισμός του εννοιολογικού πλαισίου με χρήση των LLMs βοηθά στην υπέρβαση του accuracy και του MRR αναφοράς όταν χρησιμοποιούνται κατάλληλες προτροπές, ανεξάρτητα από την ενσωμάτωση του συντελεστή ποινής $p(i)$. Όσον αφορά την επιλογή των προτροπών, παρατηρούμε μια ενδιαφέρουσα διακύμανση σχετικά με το ποιές προτροπές οδηγούν σε καλύτερα αποτελέσματα εμπλουτισμού, με διαφορετικές προτροπές να έχουν καλύτερη ή χειρότερη απόδοση μεταξύ των διαφορετικών LLMs. Αυτό επιβεβαιώνει τον ισχυρισμό ότι οι προτροπές δεν είναι μεταβιβάσιμες, δηλαδή μια προτροπή που έχει καλή απόδοση σε συνδυασμό με ένα συγκεκριμένο μοντέλο δεν είναι απαραίτητο ότι θα αποδίδει το ίδιο καλά όταν χρησιμοποιηθεί σε κάποιο άλλο μοντέλο. Η προτροπή "exact" φαίνεται να είναι μάλλον η πιο αδύναμη ως προς την εκμαίευση της απαραίτητης γνώσης που μπορεί να ενισχύσει περαιτέρω την πολυτροπική ανάκτηση, καθώς σε αρκετές περιπτώσεις οι μετρήσεις που αντιστοιχούν σε φράσεις εμπλουτισμένες με χρήση της προτροπής αυτής παρουσιάζουν ακόμα και τιμές κάτω από τις μετρήσεις αναφοράς. Από την άλλη πλευρά, η προτροπή "meaning_of" φαίνεται να αποτελεί την πιο ισχυρή προτροπή που επιχειρήθηκε, οδηγώντας σε αξιοσημείωτες βελτιώσεις απόδοσης συγκριτικά με την απόδοση αναφοράς στις περισσότερες περιπτώσεις. Συνολικά, ο συνδυασμός του εμπλουτισμού φράσεων μέσω του GPT-3 μαζί με το CLIP_{LAION} (με ενσωματωμένο το συντελεστή ποινής $p(i)$) ως πολυτροπικό μοντέλο ανάκτησης εξασφαλίζει τα βέλτιστα αποτελέσματα.

Επιπρόσθετα, μια ενδιαφέρουσα παρατήρηση είναι ότι τα μοντέλα Vicuna-7/13B αποδίδουν συγκρίσιμα με τα μοντέλα GPT-3/3.5 παρά το γεγονός ότι είναι τάξεις μεγέθους μικρότερα. Αυτό είναι ενθαρρυντικό αποτέλεσμα που υποδηλώνει ότι ο εμπλουτισμός φράσεων μέσω LLMs *ίσως* να μπορεί να εκτελεστεί με επιτυχία και με πιο ελαφριά LLMs. Ωστόσο, στις περισσότερες περιπτώσεις που τα LLMs με πολύ μικρότερη κλίμακα χρησιμοποιούνται για εμπλουτισμό με νέα γνώση, τα αποτελέσματα ανάκτησης δυσκολεύονται αρκετά να ανταγωνιστούν τα αποτελέματα αναφοράς, χωρίς προσθήκη γνώσης, αποκαλύπτοντας μια μη αμελητέα σχέση μεταξύ της κλίμακας και των δυνατοτήτων βελτίωσης γνώσης. Για παράδειγμα ο εμπλουτισμός μέσω του OPT-6.7B οδηγεί σε αποτελέσματα ελαφρώς χαμηλότερα από τα αποτελέσματα αναφοράς χωρίς επιπρόσθετη γνώση στις περισσότερες περιπτώσεις, ανεξάρτητα από την ενσωμάτωση του συντελεστή ποινής. Τα μικρότερα μοντέλα GPT2-XL (1.5B) και BLOOMZ-1.7B παρουσιάζουν ορισμένες βελτιώσεις σε σύγκριση με τα αντίστοιχα αποτελέσματα αναφοράς όταν δεν χρησιμοποιείται ο συντελεστής ποινής, με τα σχετικά αποτελέσματα ωστόσο να παραμένουν χαμηλά σε σύγκριση με αυτά των μεγαλύτερων γλωσσικών μοντέλων.

1.3.2 Εξαγωγή Κειμενικών Περιγραφών από τις Εικόνες για Ανάκτηση Κειμένου από Κείμενο

Παρακάτω παρουσιάζουμε τα αποτελέματα που προκύπτουν μετατρέποντας το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών σε πρόβλημα Ανάκτησης Κειμένου από Κείμενο μεταξύ των εξαγμένων περιγραφών από τις υποψήφιες εικόνες c_i και των δοθέντων φράσεων t. Στους πίνακες 1.7 και 1.8 παρουσιάζουμε τα αποτελέσματα με τις διανυσματικές κειμενικές αναπαραστάσεις που προκύπτουν με χρήση των VL Transformers και των αμιγώς γλωσσικών transformers εκπαιδευμένων στην σημασιολογική ομοιότητα. Το No-LLM αναφέρεται στην περίπτωση που δε χρησιμοποιούμε εμπλουτισμό των φράσεων, ενώ οι υπόλοιπες περιπτώσεις αντιστοιχούν σε εμπλουτισμένες φράσεις με χρήση προτροπών που παρουσιάζονται στον Πίνακα 1.2. Σε όλα τα αποτελέσματα που παρουσιάζονται το LLM που επερωτάται είναι το GPT-3, καθώς παρουσίασε την ανώτερη απόδοση κατά των εμπλουτισμό φράσεων. Αν και ως μετρικές ομοιότητας έχουμε χρησιμοποιήσει τόσο την ομοιότητα συνημιτόνου όσο και την ευκλείδια και manhattan απόσταση, παρακάτω παρουσιάζουμε αποτελέσματα με χρήση αποκλειστικά της απόστασης manhattan, καθώς φάνηκε να εξασφαλίζει την καλύτερη απόδοση στην πλειοψηφία των περιπτώσεων. (Αναλυτικά αποτελέσματα και με τις υπόλοιπες μετρικές ομοιότητας παρουσιάζονται στην Ενότητα

5.2.2).

Table 1.7: Αποτελέσματα από την ανάκτηση Κειμένου από Κείμενο (με ή χωρίς τον εμπλουτισμό φράσεων μέσω του GPT-3) με χρήση της απόστασης manhattan ως μετρική ομοιότητας, για τα διαφορετικά VL μοντέλα.

		BLIP		**BLIP-L**		**GiT**		**GiT-L**		**ViT-GPT2**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
BLIP	**Greedy Search**										
	No-LLM	13.61	32.89	14.90	34.00	15.55	34.67	18.57	36.70	14.47	33.37
	exact	10.80	30.16	12.31	31.88	9.94	29.31	11.88	30.58	11.02	29.46
	what_is	10.58	29.79	13.17	32.38	11.02	30.07	11.23	30.70	10.15	29.05
	describe	9.94	29.26	12.31	31.22	10.58	29.67	11.66	30.51	10.58	29.89
	meaning_of	10.58	30.02	12.96	32.07	9.29	28.91	9.94	30.14	10.37	29.60
	Beam Search										
	No-LLM	14.04	33.24	15.77	35.19	16.63	35.70	17.71	36.52	14.47	33.91
	exact	10.80	30.57	10.37	29.76	10.37	30.04	10.15	30.46	12.74	31.75
	what_is	13.39	32.60	10.15	30.31	12.74	32.00	10.37	30.47	11.66	31.02
	describe	11.45	31.15	11.02	30.22	11.23	30.41	10.58	30.61	12.31	31.40
	meaning_of	10.37	30.96	10.37	30.10	9.72	30.16	10.80	30.74	13.17	33.08
CLIP	**Greedy Search**										
	No-LLM	36.50	56.58	41.25	58.18	36.93	55.57	39.09	56.98	26.78	46.51
	exact	45.21	63.28	47.60	63.26	43.41	60.58	45.51	62.14	25.75	46.00
	what_is	44.96	63.88	46.60	63.48	48.01	63.91	48.24	64.68	34.43	52.81
	describe	47.87	65.46	48.82	64.84	48.34	64.90	53.08	67.90	28.44	49.29
	meaning_of	46.78	64.70	46.56	63.47	44.35	62.07	48.34	64.99	33.48	52.48
	Beam Search										
	No-LLM	41.04	58.45	44.71	61.70	41.68	59.16	43.84	60.14	29.59	49.30
	exact	47.90	64.16	50.60	67.08	48.20	63.93	49.10	64.70	32.63	52.13
	what_is	49.18	65.23	55.04	70.18	51.05	66.74	49.18	65.81	36.77	55.08
	describe	46.92	64.50	57.35	71.63	54.03	68.52	54.98	69.84	35.55	55.07
	meaning_of	46.56	64.01	52.55	68.57	49.00	65.45	50.55	66.27	37.47	55.65
CLIP-L	**Greedy Search**										
	No-LLM	38.01	57.95	41.90	60.49	39.96	57.78	45.36	62.24	28.94	48.00
	exact	44.61	62.07	45.21	62.40	43.11	60.71	46.11	63.33	28.74	47.68
	what_is	43.09	61.97	47.31	64.83	44.73	62.32	50.59	66.57	31.62	50.46
	describe	45.97	63.01	48.34	65.36	46.45	63.40	49.29	65.05	27.49	48.35
	meaning_of	45.01	63.42	47.45	64.70	45.01	63.42	50.33	67.02	31.49	50.19
	Beam Search										
	No-LLM	42.12	60.12	47.52	65.43	43.11	61.85	45.36	62.95	27.65	48.12
	exact	48.80	64.89	51.20	67.12	43.11	61.85	49.70	65.13	32.04	50.97
	what_is	49.41	65.88	54.10	70.17	49.18	65.85	51.05	67.35	35.83	54.31
	describe	46.45	63.57	54.50	70.82	48.82	65.37	52.61	68.64	36.02	54.45
	meaning_of	47.89	65.37	52.55	69.29	45.90	64.44	51.22	67.35	32.82	52.16
CLIP$_{LAION}$	**Greedy Search**										
	No-LLM	40.82	59.47	46.22	63.72	38.88	57.00	43.84	61.74	32.18	51.15
	exact	40.82	58.17	43.20	60.76	39.52	56.45	41.47	59.42	33.26	51.29
	what_is	42.12	60.14	45.79	63.26	42.33	59.32	45.57	63.02	35.85	53.45
	describe	42.55	60.55	46.00	63.22	39.09	58.00	45.36	62.68	32.04	52.06
	meaning_of	46.65	63.01	47.30	64.27	39.31	57.91	46.22	63.41	35.21	52.77
	Beam Search										
	No-LLM	46.00	62.43	47.30	65.14	43.20	60.20	46.65	63.66	32.83	52.24
	exact	44.06	60.97	47.08	63.31	41.90	58.63	46.65	62.78	29.37	50.28
	what_is	47.30	63.86	51.62	67.12	44.28	60.67	49.24	65.24	33.05	53.42
	describe	46.00	62.80	51.84	67.54	45.14	61.11	49.24	65.52	31.32	52.51

	meaning_of	47.08	63.66	51.62	67.47	45.57	61.61	48.38	65.09	33.05	53.44
ALIGN	**Greedy Search**										
	No-LLM	40.60	59.82	48.38	64.71	44.92	62.19	48.60	65.30	28.29	48.73
	exact	44.06	61.16	50.32	64.51	45.14	61.40	50.76	65.63	28.29	48.05
	what_is	47.73	64.65	50.32	66.29	47.95	64.32	54.64	69.27	34.77	53.33
	describe	47.08	64.31	51.40	66.87	46.87	63.98	54.43	69.11	31.97	51.27
	meaning_of	50.54	66.93	53.78	68.79	50.54	66.38	57.02	70.92	33.48	51.96
	Beam Search										
	No-LLM	46.65	63.48	54.64	69.92	45.36	62.87	54.00	68.47	33.05	52.52
	exact	46.87	63.94	53.13	68.22	46.22	62.88	53.35	67.79	33.91	52.33
	what_is	49.68	66.77	61.12	74.14	51.40	66.93	57.67	71.49	37.15	55.86
	describe	52.48	67.89	59.83	73.42	52.27	68.00	58.75	72.02	33.26	53.84
	meaning_of	49.89	67.42	62.42	75.67	55.51	69.99	59.61	73.21	36.72	55.98

Table 1.8: Αποτελέσματα από την ανάκτηση Κειμένου από Κείμενο (με τον εμπλουτισμό φράσεων μέσω του GPT-3) με χρήση της απόστασης manhattan ως μετρική ομοιότητας, για τα διαφορετικά γλωσσικά μοντέλα, εκπαιδευμένα στην σημασιολογική ομοιότητα.

		Greedy								**Beam**							
		BLIP		**BLIP-L**		**GiT**		**GiT-L**		**BLIP**		**BLIP-L**		**GiT**		**GiT-L**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
distil-roberta base	exact	38.66	56.78	38.88	57.05	36.93	54.85	42.33	59.19	42.55	59.35	42.98	60.78	41.04	58.83	48.16	63.82
	what_is	41.68	58.96	39.09	57.36	41.68	58.12	43.20	60.53	44.92	62.06	45.36	63.27	42.12	59.99	50.11	65.65
	describe	39.96	58.52	42.98	59.69	41.04	58.48	45.57	62.19	43.41	60.34	46.44	63.53	44.06	61.57	48.60	64.99
	meaning_of	41.47	59.13	41.04	59.03	39.96	57.64	44.28	62.26	44.49	61.76	47.73	64.63	45.57	62.30	53.13	67.83
stsb roberta base	exact	38.88	56.84	40.39	58.95	42.33	59.11	40.39	58.95	40.17	57.47	45.36	61.89	43.20	59.87	45.36	61.95
	what_is	40.60	58.51	44.06	61.87	39.74	58.31	44.06	61.87	42.98	59.92	44.71	63.28	44.06	61.24	47.73	64.56
	describe	42.55	60.03	40.17	57.58	42.12	59.63	45.14	62.17	42.98	60.04	47.95	65.14	45.14	62.48	47.73	64.66
	meaning_of	45.14	61.37	40.17	58.30	41.04	59.02	46.65	63.33	43.63	61.02	49.46	66.34	43.20	61.61	49.89	66.24
stsb-dis tilroberta base	exact	36.50	54.82	38.88	56.45	39.52	57.27	36.93	56.61	38.23	56.15	46.00	62.39	41.47	58.30	41.68	58.72
	what_is	37.58	56.76	38.88	57.49	40.39	57.71	42.33	59.96	40.17	58.15	46.65	64.31	41.68	59.28	44.49	61.41
	describe	36.93	56.10	42.55	59.51	39.96	57.50	43.41	60.55	41.47	58.89	47.73	64.76	41.90	60.25	47.73	63.33
	meaning_of	37.80	56.97	41.68	59.40	40.39	57.80	43.41	60.67	39.74	58.31	49.89	66.33	43.84	60.83	46.00	63.03
stsb mpnet base	exact	36.50	56.07	41.04	58.82	41.25	59.10	43.63	60.66	40.60	58.75	46.87	64.34	43.41	61.26	46.00	62.31
	what_is	40.39	58.92	42.33	60.80	42.55	60.90	45.14	62.40	43.41	61.05	49.68	66.47	45.14	62.24	49.46	65.19
	describe	42.12	60.14	43.41	60.90	44.06	61.67	49.89	65.66	43.20	61.66	47.08	65.61	47.52	64.19	50.11	66.02
	meaning_of	41.25	59.56	43.84	62.06	44.49	62.28	50.76	66.00	44.06	61.86	50.11	67.09	47.52	64.08	50.76	66.71
all MiniLM L6	exact	42.55	59.89	45.36	62.33	41.04	59.53	45.14	61.81	42.98	60.78	49.24	65.45	43.41	61.58	49.24	64.92
	what_is	44.49	61.49	44.71	62.44	45.79	62.51	46.22	63.57	42.55	61.10	48.60	66.17	47.52	64.16	50.76	66.74
	describe	43.41	61.08	44.49	62.29	41.04	59.91	49.03	65.22	43.63	61.43	50.11	66.96	42.55	61.56	49.24	66.29
	meaning_of	42.12	60.09	45.79	63.28	44.92	62.26	45.57	63.12	43.84	61.46	49.03	66.81	45.79	63.37	51.84	67.58
all MiniLM L12	exact	39.52	57.99	46.22	62.19	40.60	59.30	42.98	60.83	41.04	60.16	48.81	65.19	42.55	60.67	48.60	64.91
	what_is	40.39	59.02	43.63	61.88	41.68	60.14	46.00	63.36	42.76	61.36	48.81	66.42	44.71	61.97	49.46	66.10
	describe	40.60	59.33	44.28	61.94	41.90	60.48	47.73	64.36	42.12	61.47	48.38	66.11	44.49	62.40	51.40	67.73
	meaning_of	40.60	59.37	43.41	62.00	43.20	61.37	47.52	64.20	42.76	61.46	50.54	67.42	46.44	63.33	49.46	66.37
all mpnet base	exact	42.55	60.63	44.71	62.80	42.98	61.06	46.22	63.00	42.98	61.72	50.76	66.84	47.08	63.90	50.32	66.39
	what_is	42.55	61.56	46.22	64.69	45.79	63.20	49.24	66.13	43.41	62.80	54.43	70.45	46.65	64.33	50.97	67.57
	describe	42.76	61.40	46.00	64.07	43.63	61.87	48.60	65.58	44.06	62.54	52.70	68.71	45.36	63.52	52.27	68.36
	meaning_of	43.84	62.12	49.03	66.06	45.57	62.82	48.81	65.87	44.92	63.55	54.43	70.37	48.38	65.30	50.54	67.68
multi-qa distilbert cos	exact	40.60	59.86	43.41	60.81	42.33	60.13	42.55	60.73	42.98	61.53	49.03	65.71	42.98	61.51	49.24	65.35
	what_is	41.47	60.90	44.06	62.50	47.52	63.33	46.22	64.01	44.71	63.42	50.11	67.47	47.73	64.83	49.46	66.49
	describe	41.90	60.71	42.76	61.00	47.73	63.65	46.65	63.76	42.55	61.81	49.24	66.61	46.87	64.17	50.11	66.96
	meaning_of	42.55	61.11	44.71	62.70	45.79	62.61	46.65	64.59	45.57	63.70	50.97	68.08	45.57	63.66	50.76	67.46

multi-qa MinLM L6-cos	exact	39.09	57.37	40.60	58.07	40.39	58.64	41.25	59.06	42.55	60.68	47.52	64.10	39.96	58.96	44.06	61.66
	what_is	40.82	59.40	42.76	60.14	40.82	59.49	43.84	61.65	44.28	62.08	49.24	66.57	41.90	61.13	46.87	64.42
	describe	38.23	57.35	42.33	59.58	39.74	58.84	45.14	62.05	41.68	60.51	48.38	65.32	40.82	59.65	47.30	64.50
	meaning_of	40.39	59.04	42.33	60.46	41.68	60.05	45.57	62.92	42.33	61.35	48.60	65.93	42.33	61.26	49.89	66.55
sentence t5-base	exact	39.74	59.01	41.68	60.16	39.52	58.88	42.76	60.50	42.98	60.79	47.30	64.34	41.90	59.86		
	what_is	41.25	59.57	45.57	62.66	41.47	59.73	45.36	63.04	44.49	62.61	52.92	68.35	44.71	62.50		
	describe	42.12	60.87	47.52	63.63	42.98	61.17	49.03	65.37	44.06	62.14	52.92	68.66	46.00	63.14		
	meaning_of	44.28	62.02	45.36	63.02	42.76	60.73	47.52	64.92	44.49	62.61	53.35	69.15	45.79	63.18		
sentence t5-large	exact	40.60	58.94	42.76	60.61	40.17	58.41	46.65	63.08	43.84	61.44	48.81	65.22	44.06	61.32		
	what_is	41.47	60.24	47.95	63.77	40.60	59.06	47.95	64.69	46.65	63.26	53.56	69.46	48.60	64.63		
	describe	42.33	60.99	47.30	64.14	40.82	59.82	52.27	67.49	46.22	63.25	51.84	68.13	47.08	64.22		
	meaning_of	44.71	62.40	48.30	64.94	43.20	60.32	51.84	67.44	49.46	65.12	52.92	69.43	46.87	64.17		
gtr-t5 base	exact	44.71	61.47	44.28	62.11	42.33	60.17	47.08	63.21	45.57	62.99	49.03	66.35	46.22	63.08		
	what_is	43.41	61.73	45.14	62.86	44.49	62.20	47.30	63.85	47.52	64.23	49.46	67.20	47.30	64.22		
	describe	44.28	61.99	46.65	63.36	45.14	62.44	49.24	65.32	47.52	64.42	52.70	68.95	46.65	63.60		
	meaning_of	44.06	62.08	45.57	62.88	41.90	60.55	47.52	64.25	46.65	64.27	51.84	68.64	47.08	64.17		
gtr-t5 large	exact	41.68	60.12	44.49	61.76	41.47	59.94	47.73	64.05	44.92	62.17	52.70	68.39	48.16	64.31		
	what_is	44.28	62.55	46.00	64.00	45.79	62.64	48.38	64.98	46.65	64.35	53.35	69.95	49.24	65.79		
	describe	46.00	63.91	45.79	63.78	46.65	63.57	51.40	67.31	48.38	65.20	54.00	70.30	51.19	66.46		
	meaning_of	44.28	62.59	45.36	63.76	44.49	61.82	49.03	65.61	49.03	65.69	55.29	70.96	48.38	65.58		

Παρατρούμε ότι ο εμπλουτισμός των φράσεων με χρήση LLMs ενισχύει την απόδοση της ανάκτησης Κειμένου από Κείμενο, συγκρινόμενη με την αντίστοιχη περίπτωση αναφοράς-χωρίς τη χρήση LLMs- στην πλεοψηφία των περιπτώσεων. Παρόλα αυτά παραμένει πίσω από την καλύτερη επίδοση που έχουμε επιτύχει μέχρι στιγμής με τον συνδυασμό πολυτροπικής ανάκτησης και εμπλουτισμού των φράσεων μέσω του GPT-3. Υποθέτουμε ότι αυτό οφείλεται στην απώλεια πληροφοριών που προκαλείται με τη μετάβαση από οπτική σε κειμενική πληροφορία κατά την εξαγωγή περιγραφών για τις εικόνες.

Γενικά, η πλειονότητα των VL Transformers (εκτός του BLIP το οποίο φαίνεται να δυσκολεύεται αρκετά να εντοπίσει σχέσεις μεταξύ των φράσεων -εμπλουτισμένων ή μη- και των περιγραφών των εικόνων) αποδίδουν καλύτερα στην παραγωγή διανυσματικών αναπαραστάσεων για τα κέιμενα συγκριτικά με τους αμιγώς γλωσσικούς transformers, παρότι οι τελευταίοι έχουν εκπαιδευτεί ρητά εκπαιδευτεί στη σημασιολογική ομοιότητα μεταξύ κειμένων. Ειδικότερα, το ALIGN επιτυγχάνει accuracy 62.42% ξεπερώντας το βέλτιστο accuracy που επιτυγχάνουν οι αμιγώς γλωσσικοί transformers, και συγκεκριμένα το gtr-t5-large κατά 7% (55.29%).

1.3.3 Wikipedia & Wikidata για Ανάκτηση Εικόνας από Εικόνα

Στον Πίνακα 1.9 παρουσιάζουμε τα αποτελέσματα που προκύπτουν από ανάκτηση Εικόνας από Εικόνα, δηλαδή της ανάκτησης των υποψηφίων εικόνων i με βάση τις σχετικές εικόνες i_w που έχουμε εντοπίσει από το διαδίκτυο. Από το 463 δείγματα του συνόλου δοκιμής, το Wikipedia API και το Wikidata API επιστρέφουν αποτελέσματα για 460 και 324 φράσεις αντίστοιχα.

		CLIP		**ALIGN**	
Similarity	**Image source**	**acc.**	**MRR**	**acc.**	**MRR**
Cosine	Wikidata Images	34.26	50.13	31.11	47.84
	Wikipedia Images	53.26	68.14	53.26	**68.44**
Euclidean	Wikidata Images	33.64	49.24	30.83	47.52
	Wikipedia Images	52.17	66.95	**53.48**	68.40
Manhattan	Wikidata Images	33.02	48.75	31.11	47.66
	Wikipedia Images	52.82	67.25	53.26	68.27

Table 1.9: Αποτελέσματα από την ανάκτηση Εικόνας από Εικόνα

Παρατηρούμε ότι ακόμα και τα καλύτερα αποτελέσματα της ανάκτησης Εικόνας από Εικόνα δεν είναι ικανά να ανταγωνιστούν αυτά των προηγούμενων τεχνικών. Υποθέτουμε ότι αυτό οφείλεται στο γεγονός ότι οι αποκλειστικά οπτικές αναπαραστάσεις δεν είναι αρκετά εκφραστικές για να διακρίνουν λεπτομέρειες μεταξύ σημασιολογικά σχετικών υποψηφίων.

1.3.4 Μοντέλο Εκμάθησης Κατάταξης

Στον πίνακα 1.10 παρουσιάζουμε αποτελέσματα από το μοντέλο εκμάθησης κατάταξης, χρησιμοποιώντας το ALIGN ως μοντέλο ανάκτησης. Περισσότερα αποτελέσματα του μοντέλου εκμάθησης κατάταξης με χρήση του ALIGN, αλλά και του CLIP ως μοντέλου ανάκτησης παρουσιάζονται αναλυτικά στην ενότητα 5.2.4. Οι συνδυασμοί χαρακτηριστικών που παρουσιάζονται περιλαμβάνουν τα ακόλουθα:

Baseline	**LLM-enhance**		**Text retrieval features**				**Image retrieval feat.**		**Metrics**	
$p(i)$	**Prompt**	$p(i)$	**Captioner**	**Embedding**	**Similarity**	**Phrase**	**Embedding**	**Similarity**	**Acc.**	**MRR**
-	-	-	-	-	-	-	-	-	63.93	76.33
✓	-	-	-	-	-	-	-	-	68.90	80.04
✓	meaning_of	-	-	-	-	-	-	-	73.22	82.79
✓	meaning_of	✓	-	-	-	-	-	-	75.16	84.13
✓	exact	✓	-	-	-	-	-	-	70.41	81.10
✓	what_is	✓	-	-	-	-	-	-	71.71	81.52
✓	describe	✓	-	-	-	-	-	-	73.00	82.84
✓	all prompts	✓	-	-	-	-	-	-	73.87	83.96
✓	all-except exact	✓	-	-	-	-	-	-	74.30	83.80
✓	meaning_of + describe	✓	-	-	-	-	-	-	74.30	83.86
✓	all-except exact	✓	-	-	-	-	ALIGN	manhattan	76.09	85.36
✓	all-except exact	✓	-	-	-	-	ALIGN	cosine	76.52	85.29
✓	all prompts	✓	-	-	-	-	ALIGN	cosine	76.52	85.70
✓	all prompts	✓	BLIP-L-beam	ALIGN	cosine	t	ALIGN	cosine	77.61	**85.90**
✓	all prompts	✓	BLIP-L-beam	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	77.17	**86.08**
✓	all prompts	✓	BLIP-L-beam	ALIGN	cosine	$t_{meaning\_of}$	ALIGN	cosine	76.52	85.63
✓	all prompts	✓	BLIP-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	**78.48**	**86.65**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	t	ALIGN	cosine	77.83	**86.30**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	$t_{meaning\_of}$	ALIGN	cosine	77.39	**85.92**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	**79.35**	**87.23**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	euclidean	76.96	85.85
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	manhattan	76.96	**86.00**
✓	all prompts	✓	GiT-L-beam	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	76.96	**85.92**
LTR of [11] (best results)									77.97	85.88
SemEval organizers' baseline									60.48	73.87

Table 1.10: Αποτελέσματα του Μοντέλου Εκμάθησης Κατάταξης, με χρήση του ALIGN ως μοντέλου Εκμάθησης Κατάταξης. Τα επισημειωμένα με **ροζ** αντιστοιχούν στα καλύτερα αποτελέσματα, ενώ τα επισημειωμένα με **έντονη** γραμματοσειρά αντιστοιχούν σε αποτελέσματα που ξεπερνούν τα καλύτερα αποτελέσματα του [11].

1. *Βασικά Χαρακτηριστικά*: επιλογή για ενσωμάτωση (ή όχι) του συντελεστή ποινής $p(i)$ στο $score(t, i)$ για την πολυτροπική ανάκτηση
2. *Χαρακτηριστικά από τον εμπλουτισμό με χρήση LLMs*: προτροπή για την παραγωγή εμπλουτισμένων φράσεων t_e (ή ένα σύνολο n προτροπών που οδηγεί σε πολλαπλά t_e) και επιλογή για ενσωμάτωση (ή όχι) του συντελεστή ποινής $p(i)$ στο $score(t_e, i)$
3. *Χαρακτηριστικά από την ανάκτηση Κειμένου από Κείμενο*: μοντέλο για την εξαγωγή των περιγραφών c_i από τις εικόνες, σε συνδυασμό με το μοντέλο που χρησιμοποιείται για την εξαγωγή των διανυσματικών αναπαραστάσεων των κειμένων και τη μετρική ομοιότητας (ομοιότητα συνημιτόνου, ευκλείδια/manhattan

αποστάσεις), καθώς και η φράση που χρησιμοποιείται (πρωτότυπη t, ή εμπλουτισμένη t_e, ή ένα σύνολο εμπλουτισμένων φράσεων t_e εξαγμένων με διαφορετικά LLMs ή προτροπές)

4. *Χαρακτηριστικά από την ανάκτηση Εικόνας από Εικόνα*: μοντέλο για την εξαγωγή των διανυσματικών αναπαραστάσεων των εικόνων και μετρική ομοιότητας (ομοιότητα συνημιτόνου, ευκλείδια/manhattan αποστάσεις)

Για όλα τα πειράματα του Πίνακα 1.10 χρησιμοποιήσαμε τις ακόλουθες υπερπαραμέτρους: n_estimators: 500, early_stopping: 100, learning_rate: 0.03, feature_fraction: 0.25, max_bin: 100, min_child_samples: 50 and reg_alpha: 0.05. Επιπλέον, χρησιμοποιήθηκε ένας διαχωρισμός 80-20 στο σύνολο εκπαίδευσης/επικύρωσης, τοποθετώντας 2514 δείγματα στο σύνολο επικύρωσης.

Γενικά, η ενσωμάτωση του εμπλουτισμού φράσεων με χρήση LLMs φαίνεται να είναι εξαιρετικά ωφέλιμη, εξασφαλίζοντας βέλτιστες επιδόσεις σε σύγκριση με τους υπόλοιπους συνδυασμούς χαρακτηριστικών ή τις υπόλοιπες τεχνικές που έχουμε παρουσιάσει. Συνολικά, τα καλύτερα αποτελέσματα εξασφαλίζει η χρήση του ALIGN ως μοντέλου πολυτροπικής ανάκτησης έχοντας ενσωματώσει όλους τους συνδυασμούς χαρακτηριστικών. Αυτή είναι μια ενδιαφέρουσα παρατήρηση ιδιαίτερα εάν αναλογιστούμε ότι η αυτόνομη ανάκτηση Κειμένου από Κείμενο και Εικόνας από Εικόνα δεν εξασφαλίζουν ιδιαίτερα ανταγωνιστικά αποτελέσματα. Επιπρόσθετα, ο συνδυασμός πολλαπλών χαρακτηριστικών φαίνεται να είναι αρκετά ωφέλιμος. Αυτό φαίνεται τόσο στο συνδυασμό διαφορετικών προτροπών (π.χ. η επιλογή *all prompts* συνδυάζει τα χαρακτηριστικά από τη χρήση όλων των προτροπών $t_{exact}, t_{what-is}, t_{describe} t_{meaning-of}$), όσο και στο συνδυασμό των χαρακτηριστικών από διαφορετικές εκδοχές της ανάκτησης Κειμένου από Κείμενο (π.χ. η επιλογή *all* $t_e + t$ αναφέρεται στο συνδυασμό χαρακτηριστικών από τη χρήση των εμπλουτισμένων φράσεων και με τις 4 προαναφερθείσες προτροπές, συν την πρωτότυπη δοθείσα φράση t).

1.3.5 Πρόβλημα Ερώτησης-Απάντησης και Αλυσιδωτός Συλλογισμός

Στον Πίνακα 1.11 παρουσιάζουμε το accuracy που επιτυγχάνουμε με τη μετατροπή του προβλήματος Αποσαφήνισης Οπτικών Εννοιών σε ένα πρόβλημα Ερώτησης-Απάντησης (με ή χωρίς τη χρήση Αλλυσιδωτού Συλλογισμού (CoT)), είτε σε ρύθμιση μηδενικής-βολής (zero-shot) ή πολλαπών-βολών (few-shot).

Captioner	**Zero-shot**				**Few-shot (random)**	**Few-shot (top)**	**Few-shot (inv.top)**
	no_CoT	CoT	choose no_CoT	choose CoT	no_CoT	no_CoT	no_CoT
			GPT-3.5-turbo				
GiT-L (greedy)	44.49	47.30	51.84	52.27	51.19	51.40	53.56
GiT-L (beam)	40.82	36.50	50.54	49.68	46.12	47.83	45.61
BLIP-L (greedy)	47.95	43.84	49.46	44.06	48.16	48.81	50.32
BLIP-L (beam)	38.01	34.13	50.97	50.97	40.91	40.49	40.49
ViT-GPT2 (greedy)	28.94	25.05	32.40	29.81	31.32	31.45	28.91
ViT-GPT2 (beam)	30.24	25.92	32.83	33.05	32.03	28.73	23.64
			Vicuna-13B				
GiT-L (greedy)	34.34	27.65	20.52	20.52	31.89	33.63	36.30
GiT-L (beam)	11.02	7.91	19.44	11.23	< 2	< 2	< 2
BLIP-L (greedy)	30.02	23.76	20.95	21.81	35.56	36.08	36.48
BLIP-L (beam)	9.41	6.27	12.74	8.64	< 2	< 2	< 2
ViT-GPT2 (greedy)	21.60	21.17	17.49	15.33	24.83	24.94	26.11
ViT-GPT2 (beam)	11.45	6.91	16.85	12.74	2.81	3.89	4.75

Table 1.11: Accuracy από τη μετατροπή του προβλήματος Αποσαφήνισης Οπτικών Εννοιών σε πρόβλημα Ερώτησης-Απάντησης.

Γενικά, υπάρχει μια εμφανής διαφορά στην απόδοση του GPT-3.5-turbo με το Vicua-13B, υπογραμμίζοντας το ρόλο που πάιζει η κλίμακα του μοντέλου στην περίπτωση αυτή, σε σύγκριση με τις προηγούμενες. Εδώ τα μικρότερα γλωσσικά μοντέλα φαίνεται να μη διαθέτουν την απαραίτητη γνώση ή συλλογιστικές ικανότητες για την εξαγωγή της σωστής απάντησης, ανεξάρτητα από την επιλογή της προτροπής ή τη ρύθμιση μηδενικής ή πολλαπλών-βολών. Αυτό συμφωνεί με την παρατήρηση του [30] ότι οι συλλογιστικές ικανότητες των LLMs αναδύονται σύμφωνα με την κλίμακά τους. Επιπρόσθετα, δεν υπάρχει κάποιο ξεκάθαρο μοτίβο για το εάν η άπληστη αναζήτηση ή η αναζήτηση δέσμης κατά την εξαγωγή των περιγραφών των εικόνων είναι πιο αποτελεσματική για την εκμαίευση της κατάλληλης γνώσης από το LLM: στην περίπτωση των GiT-L και BLIP-L υπάρχει μια καθαρή υπεροχή της άπληστης αναζήτησης όταν αυτά συνδυάζονται με τις CoT και no_CoT προτροπές. Ειδικά για το Vicuna-13B αυτή η υπεροχή είναι πολύ χαρακτηριστική επιδυκνείοντας σημαντική πτώση απόδοσης όταν η αναζήτηση δέσμης παίρνει τη θέση της άπληστης αναζήτησης. Ωστόσο, για το ViT-GPT2 ισχύει το αντίθετο. Πάντως η απόδοση με χρήση αναζήτησης δέσμης φαίνεται να παρουσιάζει αξιοσημείωτη πτώση όταν συνδυάζεται με ρύθμιση πολλαπλών-βολών. Ταυτόχρονα, ούτε οι "choose" προτροπές φαίνεται να παρουσιάζουν κάποιο ξεχωριστό μοτίβο όταν συνδυαστούν με τις δύο μεθόδους αναζήτησης. Συνολικά, το GiT-L φαίνεται να είναι το πιο υποσχόμενο αναφορικά με τις δυνατότητές του στην εξαγωγή περιγραφών από τις εικόνες στην πλειοψηφία των αποτελεσμάτων του Πίνακα 1.11, ενώ το BLIP-L (με άπληστη αναζήτηση) φαίνεται να είναι πιο ικανό σε ρύθμιση πολλαπλών-βολών σε συνδυασμό με το Vicuna-13B.

1.4 Συμπεράσματα

Παρουσιάσαμε μια θεμελιώδη ανάλυση με εκτενή πειράματα για το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών. Πιο συγκεκριμένα χρησιμοποιήσαμε αρκετά από τα πιο επιτυχημένα μοντέλα της εποχής μας για πολυτροπική ανάκτηση, θέτοντας ισχυρά αποτελέσματα αναφοράς, παρουσιάσαμε τα πλεονεκτήματα του εμπλουτισμού των αμφίσημων φράσεων με εξωτερική γνώση, αποθηκευμένη στα βάρη των LLMs και επιτύχαμε ανταγωνιστικά αποτελέσματα εκπαιδεύοντας ένα ελαφρύ μοντέλο ανάκτησης χρησιμοποιώντας χαρακτηριστικά που εξάγαμε από τα ανεξάρτητα πειράματα που διεξάγαμε. Πιο συγκεκριμένα, η τρέχουσα εργασία είναι η πρώτη που αξιοποιεί την πλούσια γνώση που είναι αποθηκευμένη στα LLMs μέσω της χρήσης διαφορετικών προτροπών. Ακολουθώντας το παράδειγμα του LLM-as-KB καταφέραμε να ενισχύσουμε την απόδοση αναφοράς των μοντέλων αναφοράς. Επιπρόσθετα, εξετάσαμε την απόδοση της αποκλειστικά κειμενικής ή οπτικής ανάκτησης, καθώς και την απόδοση της μετατροπης του προβλήματος της Αποσαφήνισης Οπτικών Εννοιών σε ένα αμιγώς κειμενικού προβλήματος Ερώτησης-Απάντησης, στα πλαίσια του οποίου αντικαταστίσαμε τις υποψήφιες εικόνες με κειμενικές περιγραφές. Τέλος, μέσω προτροπών Αλυσιδωτού Συλλογισμού ξεδιπλώσαμε τη συλλογιστική διαδικασία των LLMs πίσω από το πρόβλημα της Αποσαφήνισης Οπτικών Εννοιών και τονίσαμε τις ερμηνεύσιμες από τον άνθρωπο πτυχές επεξήγησης σχετικά με τη διαδιακσία εξαγωγής γνώσης από τα LLMs. Συνολικά, η ανάλυσή μας αποκαλύπτει τη σημασία της κλίμακας ενός γλωσσικού μοντέλου για την εκτέλεση εργασιών σχετικών με την εξαγωγή γνώσης και συλλογισμό. Τα αποτελέσματά μας ξεπερνούν αρκετές σύγχρονες υλοποιήσεις και τα αρχικά αποτελέσματα αναφοράς, ενώ δίνουν το έναυσμα για καινοτόμες υλοποιήσεις.

Κλείνοντας αυτή τη διατριβή θα θέλαμε να προτείνουμε μερικές κατευθύνσεις για περαιτέρω βελτίωση αυτής της εργασίας που ίσως εμπνεύσουν και για ενδιαφέρουσες διαφορετικές προσεγγίσεις. Αρχικά, θα μπορούσε να διρευνηθεί η χρήση soft prompting για την περαιτέρω βελτίωση και εμπλουτισμό της διαθέσιμης γνώσης. Παράλληλα, θα μπορούσαν να δοκιμαστούν και υβριδικές προσεγγίσεις συνδυάζοντας LLMs με μεγάλης κλίμακας γράφους γνώσεων που θα μπορούσαν να βοηθήσουν στην περαιτέρω αποσαφήνιση των αμφίσημων φράσεων. Επιπλέον, πτυχές σχετικές με την επεξήγηση του προβλήματος Αποσαφήνισης Οπτικών Εννοιών θα μπορούσαν να αποτελέσουν νέα πεδία έρευνας. Αναμφίβολα, μια σε βάθος ανάλυση των αποτυχιών ανάκτησης που παρουσιάζουν οι πειραματισμοί θα μπορούσαν να δώσουν ενδιαφέρουσες πληροφορίες και να οδηγήσουν στο δρόμο για την βελτίωση και επίλυση αυτών. Τέλος, βασικό περιορισμό, που μελλοντικά θα μπορούσε να ξεπεραστεί, αποτελεί και ο πειραματισμός και με τις υπόλοιπες γλώσσες του συνόλου δεδομένων, των ιταλικών και των περσικών, πέρα από τα αγγλικά, στα οποία επικεντρωθήκαμε στη συγκεκριμένη διατριβή.

Chapter 2

Introduction

Visual Word Sense Disambiguation (VWSD) is a recently proposed challenging task. In this task, the goal is to identify the correct image from a set of competing candidates, given an ambiguous target word within a given context [67]. For instance, the phrase *andromeda tree* contains the ambiguous target word *andromeda* accompanied by the context *tree* which clarifies this ambiguity. Among the 10 candidates presented in Fig. 2.0.1, a VWSD framework attempts to retrieve the ground truth image, denoted with colored border.

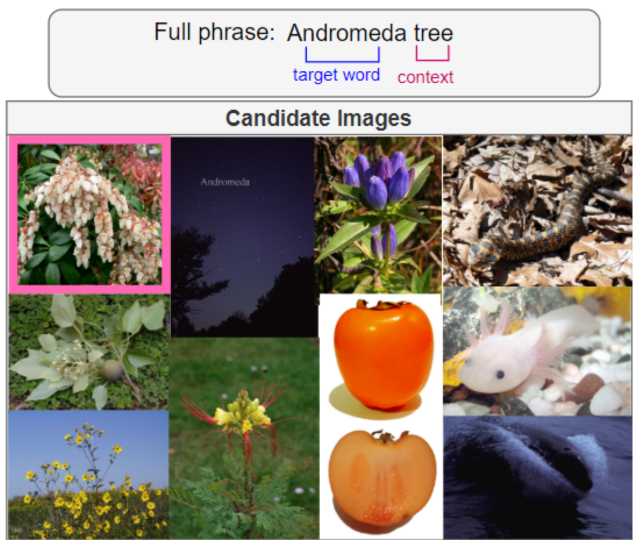

Figure 2.0.1: An example of the VWSD task.

Even though VWSD can be classified as a text-image retrieval task, there are certain inherent distinctions. For instance, upon examing the candidates presented in Fig. 2.0.1, which are somehow associated with the ambiguous word *andromeda* (which can refer to either a constellation, fish species, tree, reptile etc), an effective retrieval system should possess a high level of granularity and be exceptionally sensitive to the contextualization of the ambiguous target word. In this example, it is important for the context word *tree* to be sufficiently visible, as viewed from the perspective of the retrieval module, in order to effectively resolve any misunderstanding. Simultaneously, it is important for the retrieval module to avoid over-reliance on the *tree* context. This is because images that feature flowers and green grass can introduce a visual bias that influences the retrieval process. Consequently, the image with the highest likelihood of containing a tree may be chosen, disregarding the potentially ambiguous *andromeda* attribute. It is worth considering that there may be instances where a retrieval model has not been trained on a particular ambiguous word. In such cases, the infrequency of the concepts associated with the target word vocabulary raises the likelihood of solely depending on the context word, leading to a notable level of unpredictability during the selection process. To this end, the trustworthiness of VWSD also emerges as a pivotal aspect, hence necessitating the development of solutions that can be effectively explained.

In this thesis, a comprehensive range of implementations for Visual Word Sense Disambiguation (VWSD) is presented. Several experiments were undertaken for each of the proposed methods, resulting in one of the initial comprehensive contributions to this intriguing task:

- We exploit Large Language Models (LLMs) as knowledge bases to enrich given full phrases, so that the target word is disambiguated by incorporating more context, addressing even cases that the ambiguous word is unknown to the retrieval module.
- We convert VWSD to a unimodal problem: retrieval (text-to-text and image-to-image) and question-answering (QA) to fully explore the capabilities related models have to offer.
- Features extracted from the aforementioned techniques are used to train a learning to rank model, achieving competitive retrieval results.
- Chain-of-Thought (CoT) prompting is leveraged to guide answer generation, while revealing intermediate reasoning steps that act as explanations for retrieval.

Parts of the approaches and experiments presented in this dissertation are also published in [31] and [32].

Chapter 3

Related Work

Contents

3.1 Large Language Models (LLMs)

3.1.1 Background

Language Models (LMs) are computational models that have the capability to understand and generate human language. More specifically, a language model is a probability distribution over word sequences, having the ability to predict the likelihood of these sequences or generate new text based on a given input.

The fundamental methodology for probabilistic language modeling since 1980's has been **n-grams models**. These models are grounded on the Markov chain rule, assuming that the probability of the coming word in a word sequence depends only on a fixed size window of previous words. Hence a bigram model considers one previous word, a trigram two, and in general a n-gram n-1 previous words.

For example a bigram language model models the probability of the sequence $w_1, w_2, \ldots w_n$ as:

$$P(w_1, w_2, \ldots w_n) = P(w_2, w_1) \cdot P(w_3|w_2) \cdot \cdots \cdot P(w_n|w_{n-1}) \tag{3.1.1}$$

where the conditional probability $P(w_k|w_{k-1})$ can be estimated the proportion of occurrences of the word w_{k-1} followed by the word w_k in the corpora. The estimation of these probabilities constitutes the training of an n-gram model on text corpora in one or more languages. However, given that a language can be used to express an infinite variety of valid sentences, an n-gram model struggles to assign non-zero probabilities to word sequences that may never be encountered in the training corpora. To address this problem, a variety of smoothing techniques has been proposed over the years.

Subsequently, the advancements in the development of neural networks suggest the neural language models, which was first introduced with simple **feedforward neural language models** by [7]. A feedforward neural language model is a simple feedforward network that takes a sequence of previous words as input and gives a probability distribution over possible next words as output. Therefore, similarly to a n-gram LM, the feedforward neural LM is trained to predict the probability of a word considering the n-1 previous words.

A major difference between n-grams and feedforward neural language models is found in the way the two models represent the sequence of input words. An n-gram model assigns the word identity i to each word w_i included in the prior word sequence which should be represented. On the contrary, a feedforward neural language model mainly uses an embedding vector for each word. Thus, the latter generalizes better to unseen word sequences of the test set.

While feedforward language model introduced many of the important concepts of neural language modeling, modern neural language models use more powerful architectures like the recurrent networks or transformers networks.

Recurrent neural network (RNN) language models [56] analyse input sequences in a sequential manner, analyzing one word at a time. The aim to predict the subsequent word by considering the current word and the previous hidden state. Consequently, RNNs do not face the limited context problem observed in n-gram models, nor do they suffer from the fixed context constraint found in feedforward language models, since the hidden state can in principle represent information about all of the previous words all the way back to the begging of the sequence.

All of the aforementioned models are examples of conventional supervised learning systems for NLP, in which a word sequence x is used as an input and a text output y is predicted using the the conditional probability $P(y|x)$. In order to estimate this conditional probability we train a model using a dataset with pairs of inputs and outputs. However, in this way it becomes necessary for any task to have supervised data, which for many of them is extremely difficult to be found in large amounts. To address this, the standard evolved to *pre-train and fine-tune*, in which a fixed architecture is pre-trained as a language model, predicting the probability of observed textual data. Because the raw textual data required to train LMs is abundant, these LMs can be trained on enormous datasets, acquiring robust general-purpose properties of the language they modeling. Then, the pre-trained LMs will be modified to various downstream tasks by introducing new parameters and fine-tuning them with task-specific objective functions. Within this paradigm, the emphasis shifted to objective engineering, which involved defining the training objectives employed throughout both the pre-training and fine-tuning stages. In recent years, the *pre-train and fine-tune* paradigm has been supplanted by the *pre-train, prompt and predict* paradigm. Rather that adapting pre-trained LMs to downstream tasks through objective engineering, downstream tasks are reformulated to look more line those completed during the original LM training using a textual prompt. For instance, if we select the prompt "English: This is a sentence in english. Greek: _", an LM has the potential to offer a Greek translation to complete the missing text. By picking the relevant prompts, we may adjust the model behaviour to the point where the pre-trained LM can anticipate the correct output without any extra task-specific training. The benefit of this approach is that a single LM trained in a totally unsupervised manner can be utilised to solve a wide range of tasks when provided with a set of proper prompts. However, as with most conceptually appealing prospects, there is a catch: this method necessitates prompt engineering (section 3.1.4), which involves determining the most appropriate prompt to allow an LM to tackle the task at hand.

3.1.2 Transformer

Large Language Models (LLMs) are advanced language models with vast parameter sizes and remarkable learning capabilities that are typically pretrained on large unstructured text corpora. All modern LLMs are now built on Transformer architecture [84], which eschews recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The *encoder* and *decoder* are the two major components of the Transformer architecture. These components are employed in sequence-to-sequence operations like machine translation, where the encoder processes the input sequence and the decoder generates the output sequence.

The Transformer architecture is summarised below:

1. **Input Representation**: The input sequence is first embedded into continuous vector representations. Positional embeddings are added to these embeddings to provide information about the position of each token in the sequence.

2. **Encoder**: The encoder consists of a stack of identical layers. Each layer has two sub-layers:

   - *Multi-Head Self-Attention*: This sub-layer computes a weighted sum of embeddings, allowing each word to focus on different parts of the input sequence. Multiple attention heads run in parallel, capturing different relationships between words.

   - *Position-wise Feed-Forward Neural Network*: After the attention mechanism, each token's representation is passed through a position-wise feed-forward neural network. This introduces non-linearity and further refines the token representations.

   Residual connections[22], followed by layer-normalization[5] are employed around each of the sub-layer.

3. **Decoder**: The decoder also consists of a stack of identical layers, each containing three sub-layers:

   - *Masked Multi-Head Self-Attention*: This sub-layer acts similar to the corresponding encoder's sub-layer, but with a mask applied to prevent attending to future positions during training.

   - *Multi-Head Encoder-Decoder Attention*: This sub-layer focuses on the encoded input sequence, allowing the decoder to consider the relevant parts of the input during sequence generation.

   - *Position-wise Feed-Forward Neural Network*: Similar to the encoder, this sub-layer follows the attention mechanisms.

   As with the encoder, residual connections are used around each sub-layer, followed by layer-normalization.

4. **Output Generation**: The output of the final decoder layer is transformed into probability distributions over the output vocabulary using a linear transformation followed by a softmax activation. Throughout the training process, the model is fed with a word sequence as input to predict the subsequent word.

Several architectural variations of the Transformer have been proposed since it was first introduced by [84]. The masking pattern used on the inputs, which acts as contextual information for the model to generate a prediction, is a key distinction between these systems.

Encoder-Decoder. As previously indicated and originally proposed, the Transformer consisted of two stacks (Fig. 3.1.1): the encoder and the decoder. The encoder processes the input sequence and generates context-rich representations, which are used by the decoder to generate the output sequence step by step. Notable pretrained language models using an encoder-decoder architecture include BART[38] and T5 [66].

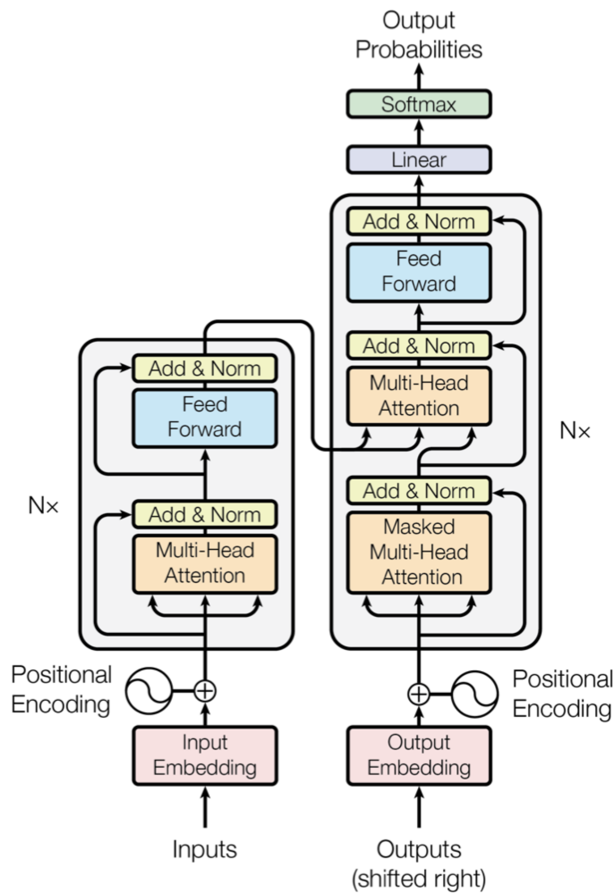

Figure 3.1.1: **The Transformer - model architecture**. The original Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of figure respectively[84]

.

Decoder-Only. While the encoder-decoder design serves as the foundational variation of the Transformer model, contemporary LLMs predominately employ a decoder-only architecture. These models have the capability to train as a conventional language model, wherein they learn to predict the next token in a given sequence. Decoder-only models lack the ability to process or represent the input sequence and output sequence separately. All tokens are treated equally during processing, and conditioning is only dependent on prior tokens due to the casual masking pattern, implying that the representation of any conditioning text is intrinsically weaker. However, this produces a simpler architecture that is well-suited to a standard autoregressive next-step-prediction pretraining objective. Notably, this architecture is the foundation of the GPT series of models [64, 9] as well as numerous other recent LLMs [92, 10, 81].

Encoder-Only. As an aside, there is an additional prevalent architectural variant that employs only a Transformer encoder layer stack. This model architecture serves as the foundation for the ubiquitous BERT [12] and its derivatives.

Overall, Transformers have revolutionised the field of NLP due to their capacity to efficiently manage sequential data, enabling parallelization and capturing long-range dependencies in texts. Using the attention mechanism to establish dependencies between input and output data, demonstrate that there is no requirement for convolutions or recurrent units to achieve state-of-the-art performance in linguistic tasks.

3.1.3 Pretraining Objectives

Pretraining is a crucial phase in the construction of LLMs, wherein the model undergoes training on an extensive, unlabeled dataset through the process of self-supervision. The selection of a pretraining objective can have a substantial influence on the subsequent applicability of the LLM. In this section, we provide an overview of the fundamental concepts behind the prevalent token-level pre-training objectives that have been extensively studied and documented in academic literature.

Maked Language Modeling (MLM) was proposed by [12]. Encoder-only models are commonly pre-trained with a masked language modeling objective. In the input text, either individual tokens or sequences of tokens are substituted with a designated mask token. The model is then trained to predict the omitted tokens.

Casual Language Modeling (CLM) is used to train autoregressive models, like encoder-decoder or decoder-only models, by predicting the next token given a prior sequence. This process enforces a causal relationship, where the model only attends to tokens that come before the predicted token in the sequence.

Next Sentense Prediction (NSP) attempts to predict whether a given pair of sentences is consecutive or not. This objective mainly serves as a supplementary task in the pre-training phase of encoder-only models and facilitates the model's acquisition of sentence associations.

3.1.4 Prompt Engineering

As previous stated, Language Models have traditionally been pre-trained on extensive text corpora and then fine-tuned on smaller labelled datasets, in order to address longstanding problems in NLP. The process of increasing the scale of language models to encompass billions of parameters, known as Large Language Models (LLMs), facilitates the emergence of novel model capabilities that may be effectively conveyed through *prompting* [47]. Prompt engineering is a crucial technique involving the meticulous design and formulation of prompts to guide the behaviour of models and achieve the desired outcomes. Effective prompt engineering is crucial to improving the performance and adaptability of language models for a variety of applications. It is imperative to initially consider the *shape* of the prompt and then determine the appropriate method, either *manual* or *automated*, for generating prompts of the desired shape.

Prompt Shape

There are two primary types of prompts: *cloze prompts*, which require the model to fill in the blanks of a string, and *prefix prompts*, which extend a given string. The selection of the appropriate option is contingent upon both the nature of the task, and the specific model employed to address the problem. For tasks involving generation or those being solved using a conventional auto-regressive LM, prefix prompts are typically preferable because they align well with the left-to-right orientation of the model. Cloze prompts are well-suited for tasks that employ masked language modeling (MLM) objective due to their near resemblance to the form of this objective. Full text reconstruction models are more flexible and can be utilised with either cloze or prefix prompts.

Manual Template Engineering

Manually creating intuitive templates based on human intuition is perhaps the most natural way for generating prompts. For instance, in their study, [9] developed a set of manually constructed prefix prompts that can effectively address a variety of tasks, such as question answering, translation and common sense reasoning.

Automated Template Engineering

Manually template creation is intuitive, enabling a certain level of accuracy in addressing diverse tasks. However, this approach is not without its limitations, as it requires a significant investment of time and knowledge in order to maximise the prompt's efficacy. To resolve these issues, a number of methods to automate the template design process have been proposed. Particularly, the automatically induced prompts can be further divided into *discrete prompts*, where the prompt is an actual text string, and *continuous prompts*, where the prompt is described directly in the embedding space of the underlying LM.

Discrete Prompts. Research efforts focused on the automated discovery of discrete prompts, also known as *hard prompts* typically involve the search for templates within a discrete space. This discrete space is often representative of natural language phrases. Several proposed techniques are enumerated below:

- *Prompt Mining*, wherein extensive text corpora are scraped to identify intermediate words or dependency paths between inputs and outputs, which are then incorporated into the template[26].
- *Prompt Paraphrasing*, which takes an initial seed prompt and generates a collection of alternative prompts, from which the prompt that yields the maximum training accuracy on the desired task is selected. This can be accomplished in a variety of methods, such as translating the prompt into another language and back [26].
- *Gradient-based Search*, which uses a gradient-based search over actual tokens to uncover short sequences that can activate the pre-trained LM to predict the desired target [86].
- *Prompt Generation*, which considers the prompt generation as a text generation task that is accomplished using standard language generation models [17].
- *Prompt Scoring*, wherein a set of prompt templates is created as candidates, then a uniderictional LM is used to rate these prompts, and finally the one with the highest probability is selected[16].

Continuous Prompts. As the purpose of prompt generation is to motivate an LM to perform a task, no human-interpretable natural language is required. Motivated by this, a variety of approaches to continuous prompts, also known as *soft prompts*, have been investigated. In particular, continuous prompts eliminate both the requirement that the embeddings of template words be native language and also the restriction that template's parameters must match those of pre-trained LM. In lieu of this, soft prompts have their own learnable parameters that can be adjusted based on training data from the downstream task. Several representative methods are highlighted below:

- *Prompt Tuning.* The central concept of prompt tuning is that prompt tokens have their own, independently-updated parameters. Thus, you can maintain the parameters of the pretrained model unchanged and only update the gradients of the prompt token embeddings. The results are comparable to the conventional method of training the entire model, and the efficacy of prompt tuning scales with model size [37].

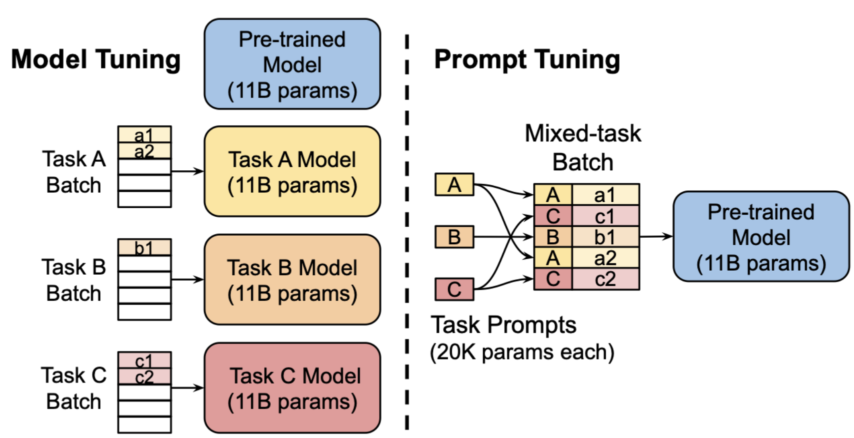

Figure 3.1.2: Prompt Tuning [37]

- *Prefix-Tuning* The technique of prefix tuning was specifically developed for the purpose of enhancing the performance of Natural Language Generation (NLG) tasks on GPT models. Prefix tuning is a technique that has resemblance to *prompt tuning*. It involves adding a sequence of task-specific vectors to the input, which can be trained and modified independently while keeping the remaining parameters of the pretrained model unchanged. One notable distinction is in the placement of the prefix parameters, which are incorporated into every layer of the model. In contrast, prompt tuning solely introduces the prompt parameters to the model's input embeddings. The optimisation of the prefix parameters is achieved through a distinct feed-forward network (FFN). The FFN is no longer used once the soft prompts have been updated [43].

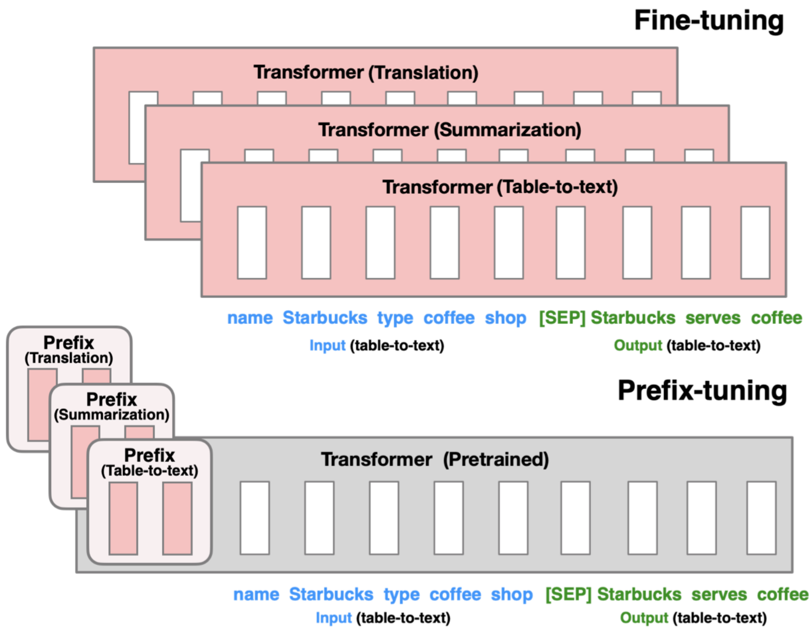

Figure 3.1.3: Prefix Tuning [43]

- *P-tuning*. P-tuning is an automated process that explores prompts throughout a continuous space in order to address the disparity between Generative Language Models and Natural Language Understanding (NLU) applications. The process of P-tuning involves utilising a limited number of continuous free parameters as prompts, which are then provided as input to pre-trained language models. Subsequently, the continuous prompts are optimised through gradient descent as a viable alternative to the process of discrete prompt searching[48].

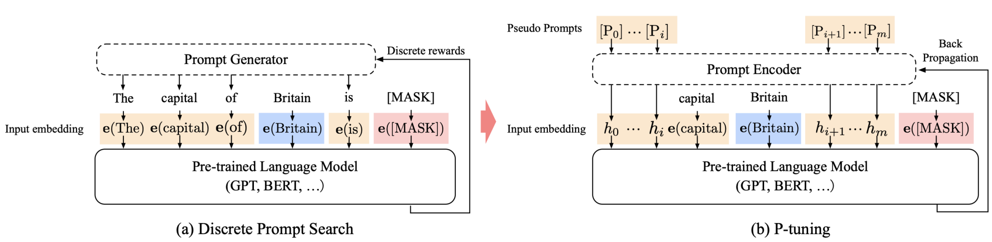

Figure 3.1.4: P-tuning [48]

In many cases, prompting methods can be used without explicit training of the LM for the downstream task; rather, an LM trained to predict the probability of text P(x) can be applied as-is to populate cloze or prefix prompts defined to specify the task. This is commonly referred to as the *zero-shot setting*, as there is no training data for the desired task. However there are methods, in which few-shot examplars are provided to the LM to guide answer-generation [9]. This is referrd to as *few-shot setting*. Specifically, *in-context learning (ICL)* is a novel few-shot paradigm which involves utilising a small number of sample demonstrations from the dataset to retrieve the related knowledge stored in the LLM without updating any parameters [14].

This technique is inspired from the way humans learn from analogy [91] and has been effectively applied ever since to serve several NLP tasks [72, 75, 73, 63, 101]. Another technique that draws inspiration from human thinking is known as *Chain-of-Thought (CoT)*, where the LM is encouraged by the prompt phrasing to generate rationales together with the predicted answer [30, 89]. Overall, prompting might be considered analogous to querying knowledge graphs in the context of traditional knowledge retrieval.

3.1.5 Language models as Knowledge Bases

Knowledge Bases (KBs) are structured repositories of information that store facts, relationships, and concepts in a way that computers can understand and query. These repositories serve as organized reservoirs of knowledge, often in the form of entities and their attributes, connected by semantic relationships. Knowledge Bases play a crucial role in various fields, including NLP, providing a foundation for systems to access and reason about information, aiding in tasks such as question answering, information extraction, and knowledge-driven decision-making. By consolidating data into a structured format, KBs facilitate the extraction of meaningful insights and the development of intelligent systems that can better understand and interact with the world. *Knowledge Graphs (KGs)* can be seen as a specialized form of KBs that employ a graph structure to represent and capture relationships and context between entities and information. With a pronounced emphasis on entity relationships, Knowledge Graphs have found extensive application in applications that require complex and contextual knowledge representation, such as semantic search, recommendation systems, question answering, and semantic reasoning.

Language models as Knowledge Bases is a novel paradigm that harnesses the implicit knowledge stored in neural weights of Language Models (LMs) similar to how explicit knowledge of Knowledge Graphs (KGs) has served related applications [61, 2]. LLMs pretrained on extensive web-based corpora have demonstrated the capacity to inherently encompass various forms of knowledge within their parameters, without requiring human supervision. These forms includes commonsense [44], factual [62], temporal [13] and beyond. Such knowledge access is pivotal for LMs to achieve state-of-the-art performance on a range of downstream tasks. Nevertheless, as is typical with many neural systems, the knowledge encoded within LMs is diffused, rendering interpretation challenging and updates complex. Consequently, this complexity presents challenges when utilizing them in real-world scenarios in contrast with KBs, which can access and update relational knowledge easier than LMs.

The question about how we can control the knowledge stored implicitly in LM's weights was initially raised by [61]. This work was among the first to analyze the accuracy and limitations of language models as a source of structured knowledge in comparison to traditional knowledge bases. Subsequently, numerous studies have delved into the LM-as-KB scenario. For instance, [87] examines the idea of viewing LMs as open Knowledge Graph and proposed a method to extract structured triples from the lm's responses to queries, effectively transforming it into a knowledge graph, while [77] explores the potential of LMs to serve as biomedical knowledge bases.

The scale of models has unveiled unprecedented capabilities of language models (LMs) in various aspects, including different types of reasoning. [3] attempts to address the challenge of solving math word problems using arithmetic reasoning, while [8] presents an extensive survey on commonsense reasoning and generation. [40] delves into the exploration of causal reasoning in popular pre-trained language models by leveraging counterfactual conditionals, which force the models to predict unusual consequences based on hypothetical propositions. Furthermore, [90] explores how generating a *chain of thought* —a series of intermediate reasoning steps— improves the ability of large language models to perform complex symbolic, commonsense and symbolic reasoning. However, certain related research questions still remain open, such as whether these models simply overfit extensive data or genuinely possess human-like reasoning capabilities [23]. More recently, the LM-as-KB paradigm has been favored by the VL community to enhance popular VL tasks [55, 82, 19, 20, 52, 98] (section 3.2.4).

3.2 Visual-Linguistic (VL) Learning

Both the Natural Language Processing (NLP) and Computer Vision (CV) communities have shown interest in vision-language tasks. For example, Image Captioning [94, 45, 71], seeks to generate a sentence or textual phrase that describes the context of an input image. Visual Question Answering (VQA) [1] seeks to provide an accurate natural language answer given an image and a natural language question about the image. Visual Commonsense Reasoning (VCR) [95] necessitates the model be able not only to answer the commonsense question but also to choose a supporting rationale for the answer." Text-Image Retrieval [28] attempts to retrieve the most relevant image based on a given text, or vice versa. As can be seen, the essence of VL learning consists of the presentation and interaction of visual and linguistic representations.

3.2.1 Linguistic Representation

Throughout the years, numerous works for text representations strategies have been proposed. *Static Word Embeddings*, *Recuurent Architectures*, and *Transformers* are three of the major milestones in the history of linguistic representations.

Static Word Embeddings. Natural Language Processing (NLP) has been significantly influenced by the evolution of word embeddings, which are frequently employed as initialization for other methods. The evolution of word embeddings has significantly influenced the development of Natural Language Processing (NLP). Rooted in the hypothesis that words appearing in similar contexts share semantic meaning, ealry attempts such as latent semantic analysis laid the foundation for the representation fo words as vecotrs in high-dimensional spaces. Word2Vec [57] marked a significant advancement by employing neural network architectures to learn word embeddings by predicting neighboring words. GloVe [60] followed this methodology and incorporated global word co-occurrence statistics into the embedding process. Doc2Vec [36], as an extension of Word2Vec, is capable of generating vector representations for a collection of words.

Recurrent Neural Networks. Recurrent Neural Networks (RNNs), designed to effectively handle sequential data (i.e. one word at a time), enabled the creation of dynamic word embeddings by considering the order of words in a text. This architecture allowed for the capture of contextual information and inter-word dependencies, representing a significant improvement over conventional word embeddings. However, early RNNs encountered difficulties in modelling long-range dependencies due to gradient vanishment problems. To address them, later extensions such as LSTMs, GRUs etc have been proposed.

Language Transformers. Transformers revolutionised Natural Language Processing (NLP) by addressing context and dependency more effectively. This innovative architecture [84] replaced sequential processing with a self-attention mechanism, enabling words to interact directly regardless of their proximity in the text. This innovation facilitated the ability of transformers to more efficiently capture extensive dependencies compared to conventional approaches and RNNs. Multiple Language Transformers have been utilised as text encoders for a variety of VL tasks. Among them, BERT [12] has become a golden standard, while other implementations utilize variants such as RobERTa[49], GPT2[64], T5[66] and BART[38].

3.2.2 Visual Representation

The representation of the visual modality has a far lower level of variation in comparison to text encoding. The majority of works in the field predominantly utilise Convolutional Architectures with few changes. It is only in recent years that some works have made efforts to utilise visual encoders based on Transformers.

Convolutional Neural Networks. Convolutional Neural Networks have become the cornerstone of modern Computer Vision for visual representations. These architectures marked a significant departure from handcrafted fetaures by leveraging the power of deep learning to automatically learn abstract visual features from raw pixel data. Visual representations can involve image-level or object-level features. CNNs, such as AlexNet [33] and VGG [74], ushered in a new era of image analysis and representation by obtaining remarkable performance in benchmark image classification tasks. The subsequent development of deeper architectures such as ResNet [21] and Inception [78] addressed gradient vanishing and computational efficiency challenges, allowing for the extraction of even more fine-grained visual details. Many works rely on CNN based classifiers such as them, while others prefer more fine-grained local representations supported by object detectors, such as Fast-RCNN[18] and Faster-RCNN[69].

Image Transformers. The introduction of transformers, which were originally designed for natural language processing but have since been adapted for computer vision, has resulted in a revolutionary transformation of visual representations. This revolutionary change marked a departure from conventional Convolutional Neural Networks (CNNs) by employing self-attention mechanisms to capture long-range dependencies and relationships in visual data. Vision Transformer (ViT) [15] suggests dividing images into fixed-size patches and treating them as token sequences, allowing transformers to process visual information. This strategy revolutionised tasks such as image classification and demonstrated transformers' capacity to comprehend and represent complex visual content. Swin Transformer [50] is a more efficient implementation due to the use of self-attention only in non-overlapping local image patches, which results to linear computation complexity as opposed to quadratic computation complexity to image size, resulting from the computation of self-attention globally.

3.2.3 Multimodal Representation

In the field of vision and language intersection, conventional approaches involved the utilisation of core distinct neural network models that independently processed visual and textual data. These models were subsequently integrated to resolve the downstream vl task. In particular, the task-specific model was fine-tuned for the downstream task by directly utilising the two separate backbones for visual and text representations. Thus, this fine-tuning process did not involve any form of generic visual-linguistic pretraining. Consequently, the final model may be susceptible to overfitting, while the pretraining of the model fails to effectively capture complex interactions and dependencies between visual and textual features and produce comprehensive visual-linguistic representations that could prove valuable for relevant downstream tasks.

Despite their limitations, these traditional approaches underscored the need for more comprehensive and integrated methods that can better exploit the inherent multimodality of vl tasks, and the requirement for aligning visual and textual features in a shared embedding space. Inspired by the success of pre-trained models in NLP, such as BERT [12] and GPT [64, 9], which significantly elevated the performance of various NLP tasks, researches began recognizing the importance of cross-modal pre-training and joint representations, and numerous cross-modal pre-training models have been developed. Similar to NLP, research, focuses mainly on two aspects, excluding pre-training data: *Model Architecture* and *Pretraining Objectives*.

- **Model Architecture**. The majority of related works are based on different variants of Transformers[84]. ViLBERT[51] and LXMERT[79] introduced the *two-stream architecture*, where two Transformers are applied to images and text independently, and then these ones are fused by a third Transformer producing the joint representations. On the contrary, VisualBERT[42], Unicoder-VL[39], VL-BERT[76] proposed the *single-stream architecture*, where a single Transformer is applied to both images and text.
- **Pretraining Objectives**. Inspired by the pretraining objectives in text models, several crucial pre-training objectives have been proposed, among them the following:
  - *Masked Language Modeling* and consequently *Masked Region Modeling*[51], which force the model to predict the masked out tokens or image regions respectively
  - *Image-Text Matching* [51, 76], which aims to align the visual and textual embeddings in a shared space. It encourages the model to understand the correspondence between images and their associated textual descriptions, facilitating cross-modal understanding
  - *Word-Region Alignment*, which finds correlations between image region and words
  - *Masked Region Classification*, which predicts the object class for each masked image region

The methods described above constitute *Supervised Learning*, where labels are either inferred from the input (known as self-supervised) or obtained from a labeled dataset, usually consisting of image-text pairs. A significant milestone is the adoption of *Contrastive Learning* for text-image representations, which is followed by certain models. Contrastive Learning is employed to autonomously learn visual-semantic embeddings. The fundamental concept involves creating an embedding space where similar pairs are positioned close to each other, while dissimilar pairs are distanced from one another. Noteworthy models that have been trained using this strategy include CLIP[65] and ALIGN[25].

The CLIP model comprises two distinct sub-models: a text encoder and an image encoder, both of which map text and images, respectively, into a shared embedding space. Guided by the principles of Contrastive Learning, these encoders are trained to assign high similarity scores to well-matched image-text pairs and low similarity scores to mismatched pairs (as illustrated in Figure 3.2.1). ALIGN employs a dual-encoder approach, focusing on aligning visual and language representations within image-text pairs. The encoder is trained with a contrastive loss function, formalized as a normalized softmax.

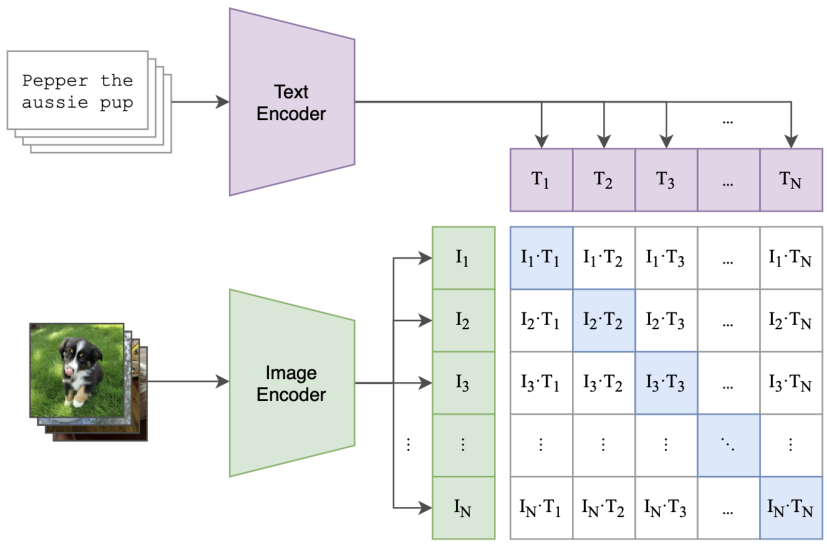

Figure 3.2.1: CLIP. Contrastive pre-training.[65]

In various works, these learning strategies are frequently amalgamated, such as BLIP[41], a multimodal fusion of encoder-decoder architecture that undergoes training encompassing both contrastive and supervised modeling objectives. Specifically, BLIP represents a unified vision-language model capable of operating within three distinct functionalities (as depicted in Figure 3.2.2):

1. A unimodal encoder, trained with an Image-Text contrastive objective, facilitating the alignment of vision and language representations.

2. An image-grounded text encoder, incorporating additional cross-attention layers to model interactions between vision and language. This encoder is trained with a Image-Text Matching objective to distinguish between positive and negative image-text pairs.

3. An image-grounded text decoder, which replaces the bidirectional self-attention layers with causal self-attention layers. This decoder shares cross-attention layers and feedforward networks with the encoder. Its training involves a Causal Language Modeling objective, generating captions based on provided images.

3.2.4 VL Learning with knowledge

Recent advancements in visiolinguistic (VL) learning have facilitated the development of multiple models and techniques, offering impressive implementations that presently address a variety of tasks necessitating the fusion of vision and language. These advancements have paved the way for transformative implementations, bringing together the realms of vision and language to tackle challenges that range from image captioning to visual question answering. Despite these remarkable achievements, the datasets commonly employed for VL pretraining are not without their limitations. These datasets, while valuable, possess a finite reservoir of visual and linguistic knowledge, creating a bottleneck that restricts the potential generalization capabilities of many VL models.

To bridge this gap and enable VL models to transcend their existing limitations, researchers have turned to

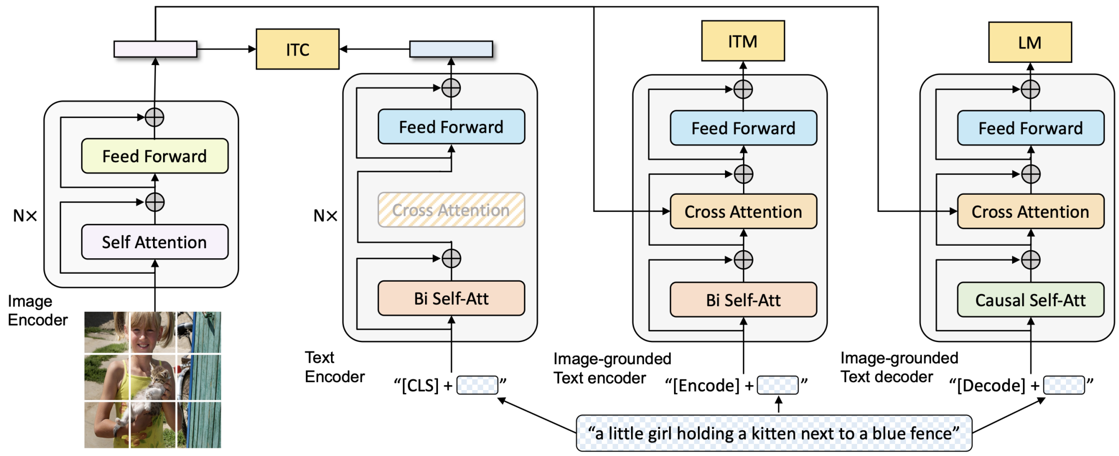

Figure 3.2.2: Pre-training model architecture and objectives of BLIP[41]

external sources of knowledge. Knowledge graphs (KGs), which organize information in a structured manner by defining entities, relationships, and semantic descriptions, have emerged as a potent tool for supplementing VL models with a richer understanding of the world. Similarly, the rise of Large Language Models (LLMs) has highlighted their prowess in capturing relational knowledge from linguistic data during pretraining. This paradigm, known as the LM-as-KB scenario, presents an intriguing approach where LLMs serve as dynamic repositories of information that can be accessed and queried to provide missing context and connections. The integration of external knowledge into VL models has led to the emergence of hybrid architectures, where traditional VL expertise collaborates seamlessly with the insights extracted from KGs and LLMs.

Notably, despite the extensive VL knowledge amassed during pretraining and fine-tuning, contemporary transformer-based VL models struggle to generalize across various concepts and scenarios requiring commonsense knowledge, abstract entities understanding, factual comprehension, and real-world event recognition. This outcome is somewhat expected, given that neither pretraining nor fine-tuning VL datasets necessitate the understanding of concepts beyond visual descriptions. Consequently, the inclusion of external knowledge, introduced at earlier or later stages of the pretraining/fine-tuning phase, becomes imperative to amplify the capabilities of VL models, enabling them to adeptly respond to more intricate real-world tasks. Such external knowledge is typically organized in structured Knowledge Graphs (KGs), employing entities, relationships, and semantic descriptions [24]. Language Models (LMs) have demonstrated the capacity to retain relational knowledge learned from linguistic data during their pretraining phase, leading to the concept of LM-as-KB [61]. This knowledge can be retrieved by constructing queries in the form of fill-in-the-blank statements, which the LM is tasked with completing. Subsequent works further establish the prowess of LMs for storing and retrieving world knowledge while showcasing their scalability corresponding to the augmentation in the number of parameters [87]. Successful deployment of LMs as knowledge bases requires meeting certain prerequisites, including accessing data similarly to KG querying, updating outdated facts without succumbing to catastrophic forgetting, unlocking their latent reasoning capabilities, and gauging their level of interpretability and explainability[2]. Despite these challenges, the remarkable achievements of Large Language Models (LLMs) in diverse linguistic tasks offer inspiration for their potential role as comprehensive and expansive knowledge bases (KBs), augmenting VL learning.

Presently, few surveys in VL learning comprehensively explore the synergy between knowledge and deep learning in VL models. An extensive exploration of knowledge-enhanced VL (KVL) was first introduced in [54]. Subsequent research [55] shifts the focus towards state-of-the-art initiatives involving transformer models for VL representation, resulting in hybrid methodologies through the integration with external knowledge.

Chapter 4

Approach

In this section, we present a wide range of implementations for VWSD. Multiple experiments are conducted for each of the implemented methods, resulting in one of the first substantial contributions to the VWSD task.

Primarily, we highlight the main contributions of this thesis, and then we provide an in-depth explanation of the implemented approaches, following the [31] and [32].

Contents

4.1 Contributions

The contributions of this dissertation are multiple and can be summarized as follows:

- We exploit Large Language Models (LLMs) as knowledge bases to enrich given full phrases, so that the target word is disambiguated by incorporating more context, addressing even cases that the ambiguous word is unknown to the retrieval module.
- We convert VWSD to a unimodal problem: retrieval (text-to-text and image-to-image) and question-answering (QA) to fully explore the capabilities related models have to offer.
- Features extracted from the aforementioned techniques are used to train a learning to rank model, achieving competitive retrieval results.
- Chain-of-Thought (CoT) prompting is leveraged to guide answer generation, while revealing intermediate reasoning steps that act as explanations for retrieval.

4.2 Dataset

VWSD Dataset

The VWSD dataset for English language consists of 12869 training samples and 463 test samples, with 10 candidate images per sample. There are negligible differences in phrase length, with the vast majority of phrases consisting of 2 words. The data samples and the official splits can be found in https://raganato.github.io/vwsd/. Additionally, the statistics of the VWSD dataset are presented bellow:

Split	#Samples	Phrase length			
		1 word	2 words	3 words	4 words
Train	12869	0	12868	0	0
Test	463	1	445	17	1

Table 4.1: Dataset statistics

4.3 Method

We followed 6 approaches to investigate the VWSD task from several different perspectives. All our approaches were tested exclusively on English.

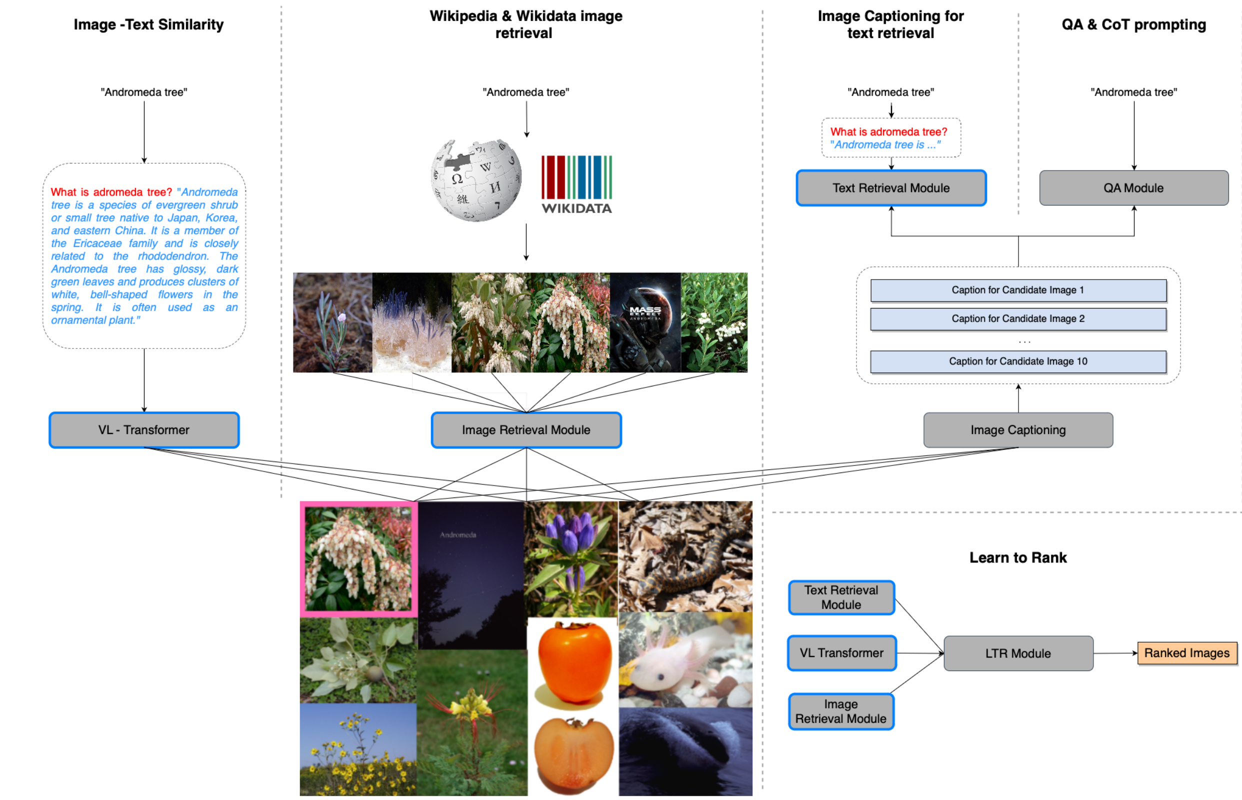

Figure 4.3.1: Method Outline

4.3.1 Image-Text similarity Baseline

As a start, we implemented a simple multimodal (VL) retrieval baseline to evaluate the capabilities of existing pre-trained VL transformers on the VWSD task. VL transformers place both images i and textual phrases t in a joint embedding space. The retrieval of the most appropriate image i with respect to a given text phrase t relies on a similarity score:

$$score(t, i) = max(sim(t, i)) \tag{4.3.1}$$

which can be computed using various similarity measures, such as cosine similarity or euclidean/manhattan distance. In this approach we exclusively experimented with cosine similarity.

The VL transformers used are reported bellow:

- CLIP with ViT[15] base encoder [1]
- CLIP with ViT large encoder, denoted as CLIP-L [2]
- CLIP$_{LAION}$ [70] (LAION/CLIP ViT-H/14), trained on LAION-2B English subset of LAION-5B [3]
- ALIGN[25] [4]

[1] https://huggingface.co/openai/clip-vit-base-patch32
[2] https://huggingface.co/openai/clip-vit-large-patch14
[3] https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K
[4] https://huggingface.co/kakaobrain/align-base

- BLIP[41] with ViT base encoder, trained on COCO[45] and denoted as BLIP_C [5]
- BLIP with ViT large encoder, trained on COCO and denoted as BLIP-L_C [6]
- BLIP with ViT base encoder, trained on Flickr30k[93] and denoted as BLIP_F [7]
- BLIP with ViT large encoder, trained on Flickr30k and denoted as BLIP-L_F [8]

We also conducted experiments with the penalty factor $p(i)$ described in [11] to modulate the retrieval preference of images that represent high similarity scores $sim(t, i)$ with multiple phrases t. The penalty is computed for each image as the average similarity between that image and all the phrases in the dataset, normalized by the frequency of image occurrence, using the following formula:

$$p(i) = \left(\frac{1}{|T|} \sum_{t_k \in T} sim(t_k, i) \right) \cdot \frac{card(i)}{max_{i_m \in I} card(i_m)} \tag{4.3.2}$$

in which T is the set of all phrases, I is the set of all images, and $card(i)$ denotes the number of samples in which image i appears.

In this case, the similarity score obeys to the following:

$$score(t, i) = sim(t, i) - p(i) \tag{4.3.3}$$

4.3.2 LLMs for phrase enhancement

We utilize a diverse range of LLMs as knowledge bases to augment the short phrases t by providing more details in a zero-shot manner, resulting in enhanced phrases t_e. Then, VL retrieval, described in the previous section 4.3.1, is facilitated. In addition, as in the baseline case, we attempt to include the penalty $p(i)$ during the final VL retrieval step, according to the following knowledge-enhanced similarity score:

$$score(t_e, i) = sim(t_e, i) - p(i) \tag{4.3.4}$$

All prompts provided to LLMs are designed upon manually crafted templates, based on the intuition that instructively requesting specific information from the model has been proven to be beneficial [30].

Prompt templates used for zero-shot LLM knowledge enhancement of textual phrases t are reported bellow:

Prompt name	**Prompt template**
exact	"<phrase> "
what_is	"What is <phrase>?"
meaning_of	"What is the meaning of <phrase>?"
describe	"Describe <phrase>."
write_description	"Write a description of <phrase>."
to_describe	"To describe <phrase> I would say that "
could_describe	"I could describe <phrase> as "

Table 4.2: Prompts for phrase enhancement via LLMs

Emerging capabilities of LLMs, such as multiple types of reasoning, are analogous to model size [30, 90], indicating that models beyond a particular scale may contain more advanced knowledge. Experimenting with models up to 13B parameters, which corresponds to the upper limit of our hardware, as well as orders of magnitude larger language models (175 parameters) accessible via public APIs, we investigate whether this assumption also holds true for knowledge enrichment. The LLMs we utilized are reported bellow:

[5] https://huggingface.co/Salesforce/blip-itm-base-coco
[6] https://huggingface.co/Salesforce/blip-itm-large-coco
[7] https://huggingface.co/Salesforce/blip-itm-base-flickr
[8] https://huggingface.co/Salesforce/blip-itm-large-flickr

- GPT-3[9] with 175B parameters, specifically *text-davinci-003* & GPT-3.5-turbo [9]
- GPT2-XL[64] with 1.5B parameters [10]
- BLOOMZ-1.7B [11] & BLOOMZ-3b [58] [12]
- OPT-2.7B [13] & OPT-6.7B [97] [14]
- LLAMA-7B [83] [15]
- Vicuna-7B [16] & Vicuna-13B [99] [17]
- Galactica-6.7B [80] [18]

4.3.3 Image Captioning for text retrieval

We leverage the metrics of unimodal retrieval by exploiting state-of-the-art image captioning transformers to convert images i to textual captions c_i. In particular, the following captioning models are used:

- BLIP Captions [41] with ViT-base[15] encoder [19]
- BLIP Captions with ViT-large encoder, denoted as BLIP-L Captions [20]
- GiT [88] with ViT-base encoder [21]
- GiT with ViT-large encoder, denoted as GiT-L [22]
- ViT-GPT2 [34] which uses ViT-base as encoder and GPT-2[64] as decoder [23]

For all image-captioning models we exploit both beam-search multinomial sampling with 10 beams to obtain k=10 captions per image i, as well as greedy search. We represent as c_i^k the k-th caption for image i, as obtained from beam search (greedy search returns only one caption). In the case of beam search, as some of the 10 captions are identical or substrings of longer ones, they are post-processed, in order to end up with only distinct ones.

We investigate two possibilities for obtaining embedding representations for the captions c_i and the phrases t. In the first scenario, embedding representations are generated using the same VL transformers as in multimodal retrieval. In the second scenario, we exploit an assortment of purely textual sentence transformers that are fine-tuned for semantic similarity [68]. These are illustrated bellow:

- distilroberta-base[24] & xlm-r-distilroberta-paraphrase [25]
- stsb-roberta-base [26], stsb-distilroberta-base[27] & stsb-mpnet-base[28]
- all-MiniLM-L6[29], all-MiniLM-L12[30] & all-mpnet-base[31]

[9] https://platform.openai.com/docs/models/gpt-3
[10] https://huggingface.co/gpt2-xl
[11] https://huggingface.co/bigscience/bloom-1.7B
[12] https://huggingface.co/bigscience/bloomz-3b
[13] https://huggingface.co/facebook/opt-2.7b
[14] https://huggingface.co/facebook/opt-6.7b
[15] https://huggingface.co/decapoda-research/llama-7b-hf
[16] https://huggingface.co/TheBloke/Wizard-Vicuna-7B-Uncensored-HF
[17] https://huggingface.co/TheBloke/Wizard-Vicuna-13B-Uncensored-HF
[18] https://huggingface.co/facebook/galactica-6.7b
[19] https://huggingface.co/Salesforce/blip-image-captioning-base
[20] https://huggingface.co/Salesforce/blip-image-captioning-large
[21] https://huggingface.co/microsoft/git-base
[22] https://huggingface.co/microsoft/git-large
[23] https://huggingface.co/nlpconnect/vit-gpt2-image-captioning
[24] https://huggingface.co/distilroberta-base
[25] https://huggingface.co/sentence-transformers/xlm-r-distilroberta-base-paraphrase-v1
[26] https://huggingface.co/sentence-transformers/stsb-roberta-base-v2
[27] https://huggingface.co/sentence-transformers/stsb-distilroberta-base-v2
[28] https://huggingface.co/sentence-transformers/stsb-mpnet-base-v2
[29] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[30] https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2
[31] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

- multi-qa-distilbert-cos[32] & multi-qa-MiniLM-L6-cos[33]
- sentence-t5-base[34], sentence-t5-large[35], gtr-t5-base[36] & gtr-t5-large[37]

Then, in both cases, we use cosine similarity or euclidean/manhattan distance to calculate $score(t, c_i^k)$, retrieving the caption embedding that is most similar to each phrase embedding. Experiments were conducted both with and without LLM-based phrase enhancement.

4.3.4 Wikipedia & Wikidata image retrieval

Image-to-image retrieval is another unimodal representations-based approach that we engaged with for the VWSD task. Consequently, in accordance with the idea of [11], we utilize Wikipedia API in order to retrieve all the relevant articles with respect to the provided phrase t, and then we retain the primary image i_w from each article. Then, we post-processed the retrieved image set by considering a maximum of $k = 10$ Wikipedia images per phrase t. The same process is repeated for Wikidata[85]. Using the same VL transformers as in multimodal retrieval, we obtained embedding representations for both the retrieved images i_w and the candidate image i. Ultimately, we search for the embeddings that are closer to one another within the embedding space. To this end, we employed either cosine similarity or euclidean/manhattan distance, according to $score(i_w, i)$.

4.3.5 Learn to Rank

Inspired by [11], we implement a Learning to Rank (LTR) model that incorporates features derviced from our aformentioned experiments. LGBMRanker with lambdarank objective[38], implemented upon the gradient boosting framework [29], is chosen as the LTR module.

The input features chosen for the LTR model captures the associations between each given phrase and the candidated images, and derived from each one of the 4 previous methods separately. More specifically, the features crafted from the baseline case, are extracted with the following steps (a)-(e):

(a) $score(t, i)$

(b) $max(score(t, i))$

(c) $mean(score(t, i))$

(d) *difference a-b*

(e) *difference a-c*

In a similar way, the steps (a)-(e) are repeated for each one of the other methods, with only replacing the $score(t, i)$ with the proper score equation. Particularly, instead of $score(t, i)$, for LLM-enhancement we use the $score(t_e, i)$, for caption-phrase retrieval the $score(t, c_i^k)$, for enhanced caption-phrase retrieval the $score(t_e, c_i^k)$, and finally for the image retrieval the $score(i_w, i)$. As a result, we end up with a 5-feature-addition into our final feature vector for each method we leverage.

We train the LTR module on several combinations of the designed features. In order to further advance LTR performance, we also attempt to combine features from enriched phrases t_e derived using different prompts. Furthermore, different similarity (cosine) and distance (euclidean/manhattan) scores are exploited

[32] https://huggingface.co/sentence-transformers/multi-qa-distilbert-cos-v1
[33] https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1
[34] https://huggingface.co/sentence-transformers/sentence-t5-base
[35] https://huggingface.co/sentence-transformers/sentence-t5-large
[36] https://huggingface.co/sentence-transformers/gtr-t5-base
[37] https://huggingface.co/sentence-transformers/gtr-t5-large
[38] LGBMRanker docs

in conjunction with these combinations, while the contribution of considering penalty factor $p(i)$ is evaluated both in baseline VL retrieval (eq. 4.3.3), as well as in the LLM-enhanced VL retrieval module (eq. 4.3.4).

4.3.6 Question Answering for VWSD and CoT prompting

We transform VWSD to a question-answering (QA) task by converting the textual phrases t to questions Q that adhere to manually crafted prompt templates. The prompts, we experimented with, are illustrated in Table. 4.3.

Prompt name	**Prompt template**
think (greedy)	“Q: What is the most appropriate caption for the <context>? Answer choices: (A) <caption for image 1> (B) <caption for image 2> ... A: Let’s think step by step. ”
think (beam)	“Q: What is the most appropriate group of captions for the <context>? Answer choices: (A) <captions for image 1 (separated with comma)> (B) <captions for image 2> ... A: Let’s think step by step. ”
CoT	“<think_prompt> <response of llm with think prompt> Therefore, among A through J, the answer is ”
no_CoT (greedy)	“Q: What is the most appropriate caption for the <context>? Answer choices: (A) <caption for image 1> (B) <caption for image 2> ... A: ”
no_CoT (beam)	“Q: What is the most appropriate group of captions for the <context>? Answer choices: (A) <captions for image 1> (B) <captions for image 2> ... A: ”
choose no_CoT (greedy)	You have ten images, (A) to (J), which are given to you in the form of captions.(A) <caption for image 1>. . . (J) <caption for image 10> You should choose the image, and therefore the caption that could better represent the <phrase>. What image do you choose?
choose no CoT (beam)	You have ten images, (A) to (J), which are given to you in the form of captions.(A) <captions for image 1 (separated with comma)>. . . (J) <captions for image 10 (separated with comma)> You should choose the image, and therefore the set of captions that could better represent the <phrase>. What image do you choose?
choose CoT (greedy)	You have ten images, (A) to (J), which are given to you in the form of captions. (A) <caption for image 1> . . . (J) <caption for image 10> You should choose the image, and therefore the caption that could better represent the <phrase>. Use the following format: Question: What image do you choose? Thought: you should always think about what you choose. Result: the result of your thought Final Answer: the image that you choose. Begin! Question: What image do you choose?
choose_CoT (beam)	You have ten images, (A) to (J), which are given to you in the form of a set of captions. (A) captions for image 1 (separated with comma) . . . (J) captions for image 10 (separated with comma) You should choose the image, and therefore the set of captions that could better represent the <phrase>. Use the following format: Question: What image do you choose? Thought: you should always think about what you choose Result: the result of your thought Final Answer: the image that you choose Begin! Question: What image do you choose?

Table 4.3: QA prompts with and without CoT.

Our experimentation includes both *zero-shot* and *few-shot* prompting. In both scenarios, the Vicuna-13B[99] and the 175B GPT3.5-turbo are suggested as LLMs to be prompted. Since, the LLMs, and so the aforementioned ones, currently handle textual but not VL inputs, we need transition to text-only representations both for images i and phrases t. Therefore, image captioning techniques are utilised to accomplish this transformation, with captions c_i being provided for each candidate image i. The models chosen for captioning are GiT-L[88] and BLIP-L[41], which are both based on ViT-large [15] encoder, and ViT-GPT2 [34], which uses ViT-base as the encoder and GPT-2 [64] as the decoder.

Reasoning capabilities of LLMs can be unlocked using a technique known as Chain-of-Thought (CoT) prompting [30, 89], in which the LLM is prompted to produce a series of intermediate reasoning steps that logically leads to its answer. Even though CoT prompting has been predominantly utilised for multi-step reasoning tasks, it is also capable of providing human-understandable *explanations* regarding the selection of the most suitable prospective image i for each phrase t. To this end, the first 5 templates and prompting pipelines of Tab. 4.3 are adopted from [30], where a "reasoning" prompt (*"Let’s think step by step/think* prompts of Tab.

4.3) retrieves the reasoning path stored in the LLM, followed by an "answer" prompt (*"Therefore, among A through J, the answer is "* CoT prompt of Tab. 4.3) that returns the final answer in an appropriate format. The rest of the templates ("choose" prompt names) are inspired from LangChain prompts [35], with "choose CoT" prompt also attempting to retrieve a reasoning path, which supports the answer of the LLM.

Zero-shot prompting

In the *zero-shot* scenario, a specific prompt from Tab. 4.3 is provided as input to either the Vicuna-13B or GPT-3.5-turbo language model. Subsequently, the language model generates the answer, designated as **A**, along with an accompanying explanation for this particular section. The answer **A** can correspond to either one of the caption options A-J or a statement that indicates that no answer can be determined. In any case, the generated answer **A** is compared to the ground truth caption to assess the effectiveness of zero-shot prompting approach under investigation.

Few-shot prompting

In addition, we conduct experiments with *few-shot prompting* instead of the previously described *zero-shot prompting*. In this case, we select k no_CoT prompts (Tab. 4.3) along with their corresponding ground truth answer **A**, thereby creating *QA in-context samples*. The number of in-context samples, denoted as k, is determined by the user. We devise three different methods for selecting these k in-context examples. In the *baseline* case, the k samples are randomly chosen from the training dataset. Nonetheless, because the relevance of selected samples relative to a chosen sample [46] and sample ordering [53] is significant, we develop two similarity-based sample selection algorithms, namely *top* and *inverse-top*. Both selection methods utilize embedding representations of full phrases included in QA prompts, which are acquired using ALIGN [25]. The retrieval of the k nearest embeddings to a given phrase embedding is easily accomplished through the utilisation of cosine similarity using the FAISS library [27]. The proposed *top* ordering technique involves arranging samples in descending order of similarity, with the most similar sample being placed first, followed by the second most similar sample, and so on until the k-th most similar sample is positioned in the k-th position. Conversely, the *inverse-top* technique involves reversing the order by positioning the most similar example in the k-th position. The choice of k varies among the different experiments, contingent upon factors like as the length of the prompt and restrictions in computer resources. This information may be found in Table 4.4.

Captioner Model	**Captioner Strategy**	k
GiT-L / BLIP-L / ViT-GPT2	greedy	5
GiT-L	beam	2
BLIP-L	beam	1
ViT-GPT2	beam	2

Table 4.4: k Value Selection

Chapter 5

Experiments

In this section, we will present the results of various experiments we conducted, in order to investigate the VWSD task from several different perspectives, following the [31] and [32]. Firstly, some preliminary information will be presented about the computational resources, hyperparameters and metrics to be utilized and then we present the evaluation results for our 6 different approaches. In addition to the quantitative results, we will present insights for a more intuitive understanding of our approaches.

Contents

5.1 Preliminaries

5.1.1 Computational Resources

In Tab. 5.1 we analyze the resources used throughout our experiments, as well as the time needed for inference on the entire test set of 463 samples. Regarding captioners, we demonstrate the time needed for one batch of 1000 images. As for LTR training time refers to the train split exclusively (12869 samples).

Model	Hardware	Time (hours)
VL Transformers for retrieval		
CLIP	GPU - NVIDIA Tesla K40 12GB	00:10 h
CLIP-L	GPU - NVIDIA Tesla K40 12GB	00:15 h
CLIP$_{LAION}$	GPU - NVIDIA Tesla K40 12GB	00:40 h
ALIGN	GPU - NVIDIA Tesla K40 12GB	00:08 h
BLIP	GPU - NVIDIA Tesla K40 12GB	00:20 h
BLIP-L	GPU - NVIDIA Tesla K40 12GB	00:45 h
LLMs for phrase enhancement		
GPT2 XL 1.5B	GPU - NVIDIA Tesla K40 12GB	00:30 h
OPT 2.7B	GPU - NVIDIA Tesla K40 12GB	01:45 h
OPT 6.7B	2 x GPU T4 14.8GB	02:00 h
BLOOMZ 1.7B	GPU - NVIDIA Tesla K40 12GB	00:15 h
BLOOMZ 3B	GPU 12.8GB	00:20 h
Galactica 6.7B	2 x GPU T4 14.8GB	02:15 h
LLAMA 7B	2 x GPU T4 14.8GB	01:00 h
Vicuna 7B	2 x GPU T4 14.8GB	01:15 h
Vicuna 13B	2 x GPU T4 14.8GB	02:00 h
Image Captioners		
BLIP (batches with 1000 images each)	GPU - NVIDIA Tesla K40 12GB	∼02:00 h / 1000 images
BLIP-L (batches with 1000 images each)	GPU - NVIDIA Tesla K40 12GB	∼03:00 h / 1000 images
GiT (batches with 1000 images each)	GPU - NVIDIA Tesla K40 12GB	∼03:00 h / 1000 images
GiT-L (batches with 1000 images each)	GPU - NVIDIA Tesla K40 12GB	∼04:00 h / 1000 images
ViT-GPT2 (batches with 1000 images each)	GPU - NVIDIA Tesla K40 12GB	∼02:00 h / 1000 images
Sentence Transformers		
xlm-r-distilroberta	NVIDIA TITAN Xp 12GB	< 00:08 h
stsb-roberta-base	NVIDIA TITAN Xp 12GB	< 00:08 h
stsb-distilroberta-base	NVIDIA TITAN Xp 12GB	< 00:04 h
stsb-mpnet-base	NVIDIA TITAN Xp 12GB	< 00:07 h
all-MiniLM-L6	NVIDIA TITAN Xp 12GB	< 00:03 h
all-MiniLM-L12	NVIDIA TITAN Xp 12GB	< 00:04 h
all-mpnet-base	NVIDIA TITAN Xp 12GB	< 00:06 h
multi-QA-distilbert	NVIDIA TITAN Xp 12GB	< 00:06 h
multi-QA-MiniLM-L6	NVIDIA TITAN Xp 12GB	< 00:04 h
LTR		
LTR training	CPU - 16GB RAM	∼00:20 h
LTR prediction	CPU - 16GB RAM	< 00:01 h

Table 5.1: Resources used for our experiments and time needed

5.1.2 Hyperparameters

In this section we report the hyperparameters used and insights for each model among the different approaches, we followed.

LLMs for phrase enhancement

For each LLM model we tested as enhancer for the given phrases, we use a max of 70 tokens as text output length, in order to fit with the text input length constraint for CLIP VL transformer and its versions. Except from *max_tokens* parameter, for GPT-3 ns GPT-3.5 turbo models we further set the parameters shown in Tab. 5.2 and Tab. 5.3 respectively.

Parameter	Value
temperature	0.0
top_p	1.0
frequency_penalty	0.0
presence_penalty	0.6

Table 5.2: GPT-3 Hyperparameters

Parameter	Value
system_message	"You are an intelligent assistant."

Table 5.3: GPT-3.5 turbo Hyperparameters

Image Captioning for text retrieval

For each captioner we used to extract captions for the candidate images, we use a max of 50 tokens as text output (caption) length.

Learn to Rank

As learning to rank model we used LGBMRanker from lightgbm library with the hypeparameters shown in Tab. 5.4.

Hyperparam	Value
n_estimators	500
early_stopping_patience	100
learning_rate	0.03
feature_fraction	0.25
max_bin	100
min_child_samples	50
reg_alpha	0.05

Table 5.4: LTR model Hyperparameters

5.1.3 Metrics

The methods to be tested ultimately output ranked list of the the most similar candidate images i for a given phrase t. Therefore, the following metrics were be used:

Accuracy

Accuracy is one metric for evaluating classification models. It is defined by the following formula:

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} \tag{5.1.1}$$

Informally, we could say that accuracy is the fraction of predictions our model got right.

In our case, we define a prediction as correct, when the ground truth image among the 10 candidate images of a sample, predicted indeed as the most appropriate image for the corresponding phrase.

Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) is a measure to evaluate systems that return a ranked list of answers to queries.

For a single query, the *reciprocal rank* is $\frac{1}{rank}$, where *rank* is the position of the highest-ranked answer $(1, 2, 3, ..., N)$ for N answers returned in a query. If no correct answer was returned in the query, then the reciprocal rank is 0.

For multiple queries Q, the Mean Reciprocal Rank is the mean of the Q reciprocal ranks.

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{5.1.2}$$

In our case, each sample of the dataset corresponds to one single query Q, with rank be equal to the position in which the ground truth image was predicted in the ranked list of the most similar candidate images.

5.2 Results

5.2.1 LLMs for phrase enhancement

Quantitative results. In Tab. 5.5 and Tab. 5.6 we present results regarding LLM-based phrase enhancement involving all VL retrieval models with and without penalty $p(i)$ respectively. Baselines refers to VL retrieval with non-enhanced phrases t.

Table 5.5: Results for zero-shot LLM-based enhancement with penalty $p(i)$. **Colored** instances denote overall best results per metric, while **bold** numbers indicate best results for each LLM.

		CLIP		**CLIP-L**		**CLIP$_{LAION}$**		**ALIGN**		**BLIP$_C$**		**BLIP-L$_C$**		**BLIP$_F$**		**BLIP-L$_F$**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
Baseline		63.28	76.27	62.85	76.24	71.06	81.50	68.90	80.00	60.91	74.33	64.58	77.51	60.48	73.87	69.76	80.42
GPT2-XL	exact	53.88	69.51	56.32	71.12	57.45	72.53	53.35	69.57	47.52	63.67	47.95	64.49	41.90	59.79	50.54	66.96
	what_is	61.22	74.89	61.44	75.83	64.58	77.61	63.93	76.33	57.02	70.42	57.67	71.78	51.62	67.78	61.34	74.42
	meaning_of	60.82	75.00	**65.58**	77.55	65.23	**78.04**	64.15	76.65	58.75	72.64	59.18	73.27	52.92	68.27	63.07	75.96
	describe	57.58	72.38	60.82	74.82	65.01	77.17	58.10	72.67	54.43	68.78	53.78	69.38	47.73	65.08	57.24	71.80
	write_description	58.90	72.87	62.42	75.63	63.28	77.11	60.26	74.37	50.76	67.19	54.21	69.57	48.60	65.12	60.04	73.43
	to_describe	59.74	73.86	62.34	75.55	63.50	76.40	59.40	73.46	52.92	67.80	53.35	68.86	48.16	64.47	55.94	70.67
	could_describe	57.08	71.87	59.26	73.23	65.01	76.87	58.10	72.49	52.05	68.22	55.08	70.18	46.87	64.44	57.02	72.03
BLOOMZ-1b7	exact	61.44	74.50	64.92	77.42	66.52	78.50	65.87	77.60	64.58	76.28	65.66	77.13	59.18	72.70	67.39	78.67
	what_is	63.71	76.41	66.74	78.92	69.98	80.67	65.44	77.65	63.28	75.95	65.23	77.68	58.32	72.06	66.52	78.30
	meaning_of	62.63	76.38	65.01	78.17	69.33	80.69	66.74	78.27	63.50	76.44	65.44	78.29	58.53	72.50	68.25	79.74
	describe	64.72	77.21	64.07	77.47	**69.98**	**81.40**	68.90	80.28	61.34	75.61	64.36	77.45	58.53	72.41	66.74	78.81
	write_description	62.61	76.00	65.93	78.07	68.90	80.13	68.68	79.70	60.04	73.52	64.58	77.13	57.02	71.19	67.17	78.66
	to_describe	63.20	76.36	66.45	78.20	69.11	80.13	70.41	80.70	64.15	76.77	66.95	78.66	57.88	72.20	68.25	79.81
	could_describe	64.86	77.13	63.99	77.10	66.95	79.11	69.33	79.96	62.42	75.65	65.87	78.17	58.10	72.47	68.47	79.89
OPT-2.7B	exact	62.85	76.00	62.85	75.93	**71.06**	**81.46**	68.68	79.89	61.12	74.46	64.58	77.41	60.26	73.73	69.76	80.36
	what_is	60.98	74.85	66.30	78.10	66.95	78.99	63.28	75.95	60.91	74.43	66.31	77.86	57.24	71.15	67.70	78.58
	meaning_of	62.15	75.60	65.25	77.45	69.11	80.56	65.66	77.54	61.99	75.35	63.93	76.88	58.32	71.65	65.44	77.69
	describe	61.05	74.75	66.08	78.14	68.03	79.89	64.79	77.62	61.77	74.73	66.31	77.57	57.67	71.48	68.03	79.03
	write_description	56.44	71.82	63.37	76.52	68.03	80.34	65.44	77.68	61.99	74.65	62.42	75.29	54.64	69.49	64.79	76.70
	to_describe	62.53	76.07	65.17	78.02	68.03	79.77	64.15	76.92	63.07	75.22	61.99	74.91	57.02	70.74	65.44	77.33
	could_describe	59.83	73.70	56.99	72.31	67.39	79.06	68.68	80.41	65.23	77.44	66.09	77.44	57.45	71.88	68.25	79.57
BLOOMZ-3b	exact	61.26	74.59	62.99	76.18	67.82	79.31	66.52	78.36	60.48	73.13	63.28	76.00	57.02	71.23	65.66	77.49
	what_is	64.36	76.82	68.25	79.82	**71.92**	**81.78**	67.39	78.72	61.34	74.94	66.95	78.47	59.61	73.35	68.47	79.58
	meaning_of	65.58	77.96	67.32	78.76	70.63	81.47	68.47	79.14	63.71	76.52	66.31	78.55	59.40	73.60	68.03	79.26
	describe	62.01	75.38	65.28	78.07	70.84	81.11	66.09	78.60	62.85	75.65	67.39	78.71	57.24	71.72	67.82	79.20
	write_description	65.73	77.16	66.81	79.16	68.90	79.86	68.68	80.07	62.85	76.23	64.36	77.61	58.53	72.21	67.39	79.17
	to_describe	64.79	77.49	68.03	79.21	68.68	79.94	70.19	80.66	65.23	77.45	66.31	78.16	60.69	73.92	68.68	79.81
	could_describe	65.23	77.40	65.66	78.09	69.33	80.80	69.11	79.87	65.23	77.38	67.17	78.75	61.34	73.87	69.33	80.18
OPT-6.7B	exact	62.63	75.84	62.20	75.54	70.63	81.15	67.82	79.24	60.91	74.23	64.79	77.58	59.83	73.40	69.11	79.94
	what_is	61.79	75.70	64.63	77.68	68.03	79.41	64.79	77.23	61.77	75.01	63.07	76.16	57.88	71.79	65.87	77.77
	meaning_of	62.17	75.84	63.61	77.19	69.55	80.79	66.74	78.47	63.28	75.93	65.44	77.43	59.83	72.96	68.03	78.75
	describe	64.43	76.91	65.73	78.24	**70.84**	**81.19**	65.23	77.89	61.12	74.67	63.93	77.07	56.16	71.30	66.09	78.38
	write_description	60.38	74.46	60.61	75.19	69.76	80.90	64.58	77.04	57.02	71.13	61.34	74.24	53.13	68.91	64.58	76.70
	to_describe	63.98	76.77	61.14	75.49	66.31	78.74	68.03	78.95	61.77	74.73	60.26	74.38	55.51	70.23	65.87	77.70
	could_describe	58.59	73.21	61.45	74.60	66.31	78.27	65.44	77.76	63.07	75.34	64.36	76.09	58.32	71.86	65.23	77.28
Galactica-6.7B	exact	49.66	66.59	56.24	71.44	55.72	71.25	52.48	68.45	43.21	60.82	50.32	66.95	41.90	59.68	54.00	70.23
	what_is	60.13	74.40	62.78	76.11	65.66	77.73	62.63	75.50	57.88	72.62	61.12	74.59	53.56	68.73	64.79	76.81
	meaning_of	60.09	74.69	62.20	74.80	**68.68**	**80.24**	60.32	75.04	61.56	74.57	61.34	74.99	52.27	68.02	61.99	75.49
	describe	59.61	74.04	60.48	75.13	63.93	77.78	62.20	75.80	55.51	70.74	56.59	71.25	52.92	68.13	58.96	73.49
	write_description	58.54	73.10	60.75	74.93	65.01	77.73	57.88	72.64	55.72	70.27	56.16	70.83	49.24	64.97	60.91	74.25
	to_describe	57.78	73.32	59.75	74.52	63.28	76.21	56.16	71.45	55.29	70.42	55.72	70.92	51.19	66.89	60.26	74.05

	could_describe	62.45	75.98	63.97	76.62	67.60	79.44	58.10	72.89	55.51	70.00	56.59	71.38	49.24	66.15	60.48	74.29
LLAMA-7b	exact	55.08	70.70	57.45	72.23	62.20	76.04	60.48	74.10	52.92	68.76	54.00	69.51	48.81	65.54	58.10	72.33
	what_is	64.92	77.26	67.10	78.93	68.90	79.07	66.55	80.48	68.03	68.03	66.95	78.48	58.75	73.29	70.84	80.84
	meaning_of	61.76	76.12	62.90	76.53	67.82	80.11	65.66	78.49	60.26	74.40	62.42	75.71	54.00	69.84	65.87	78.19
	describe	64.21	76.80	66.81	79.04	**72.14**	**82.02**	69.76	80.34	64.79	76.58	66.09	77.99	58.53	72.44	69.11	80.15
	write_description	62.28	76.00	63.84	76.77	71.27	81.62	69.76	80.59	67.17	77.86	65.87	77.80	59.40	73.34	68.47	79.55
	to_describe	65.54	77.80	66.32	79.24	68.68	79.93	60.91	74.71	61.34	74.32	61.77	74.77	54.00	69.65	63.07	75.97
	could_describe	60.34	74.16	65.02	76.30	64.15	76.40	61.56	74.36	54.00	68.67	57.45	71.65	52.27	67.00	61.34	74.50
VICUNA-7b	exact	58.10	72.46	59.18	73.62	61.99	74.95	64.58	77.20	55.94	70.98	59.83	73.52	53.56	69.18	62.20	75.68
	what_is	66.02	77.82	69.05	80.16	68.25	79.76	69.76	80.49	66.74	78.00	69.33	80.00	61.34	74.61	70.84	81.00
	meaning_of	65.14	77.85	67.76	79.49	69.76	80.25	70.41	81.13	65.87	77.84	66.31	78.43	62.20	74.73	68.03	79.40
	describe	65.44	78.01	68.47	79.97	70.41	81.18	72.79	82.26	68.25	79.09	70.41	80.85	62.20	75.09	**73.43**	**82.45**
	write_description	63.83	76.61	67.32	79.40	68.47	80.24	72.14	81.17	65.23	77.36	67.39	79.05	58.53	73.03	70.63	80.73
	to_describe	64.47	77.04	68.20	79.61	67.82	79.40	69.98	81.07	63.93	76.32	64.15	76.84	57.88	72.27	68.03	79.26
	could_describe	65.44	77.46	65.23	77.83	68.25	79.69	72.14	82.16	63.93	76.32	65.01	76.84	57.67	72.27	68.90	79.26
VICUNA-13b	exact	60.61	74.49	62.12	75.60	64.15	77.10	67.60	79.48	60.69	73.98	65.01	77.46	54.86	69.91	66.74	78.74
	what_is	65.44	77.63	69.98	80.58	70.84	81.05	72.14	81.69	67.82	79.01	69.11	79.87	59.18	73.48	**73.43**	**82.91**
	meaning_of	65.43	78.25	68.26	79.83	70.63	81.03	70.63	81.33	67.82	78.70	69.76	80.66	61.99	75.08	71.71	81.80
	describe	64.36	77.01	63.28	76.85	66.74	78.64	69.98	80.70	61.12	74.73	66.95	78.57	56.16	71.10	67.17	79.11
	write_description	65.35	77.71	66.45	78.70	70.41	81.60	71.71	82.01	66.31	77.95	67.39	78.74	61.34	74.70	69.98	80.87
	to_describe	65.86	78.12	69.58	80.49	70.19	80.88	71.27	81.32	63.71	76.04	65.01	77.12	58.32	72.17	68.68	79.42
	could_describe	63.08	76.10	63.30	76.87	68.03	79.26	66.74	78.98	63.28	75.26	63.93	76.57	55.94	69.90	66.74	78.28
GPT-3.5	exact	58.86	72.09	60.18	72.73	64.36	75.38	62.42	74.43	57.02	70.78	59.18	72.32	52.92	67.40	63.07	74.65
	what_is	66.52	78.81	69.35	80.51	70.63	81.46	70.41	81.42	67.60	78.56	68.47	79.67	60.91	74.30	71.71	82.02
	meaning_of	67.76	79.76	69.06	80.55	**73.65**	**82.71**	70.41	81.38	66.52	78.59	66.52	79.16	58.53	73.31	69.98	81.46
	describe	67.32	78.95	69.28	80.31	73.22	82.50	73.22	82.73	69.33	79.90	70.41	80.80	59.83	73.65	70.63	81.29
GPT-3	exact	61.98	74.90	64.07	76.58	68.03	78.41	66.52	78.37	60.48	73.99	64.15	76.58	59.61	72.91	65.23	77.06
	what_is	67.92	79.27	70.73	81.57	72.35	82.19	71.71	82.27	68.25	78.93	68.90	79.91	60.48	74.24	69.11	80.25
	meaning_of	68.07	80.08	69.84	81.56	73.65	83.52	**74.95**	**84.09**	66.74	78.37	71.71	81.55	62.63	75.55	72.35	82.28
	describe	68.25	79.40	68.72	80.26	70.63	81.05	72.57	82.52	64.58	76.75	68.25	79.35	61.34	74.03	69.33	80.47

Table 5.6: Results for zero-shot LLM-based enhancement without penalty $p(i)$. **Colored** instances denote overall best results per metric, while **bold** numbers indicate best results for each LLM.

		CLIP		**CLIP-L**		**CLIP**$_{LAION}$		**ALIGN**		**BLIP**$_C$		**BLIP-L**$_C$		**BLIP**$_F$		**BLIP-L**$_F$	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
Baseline		59.18	72.94	60.69	74.42	67.82	79.50	65.66	77.48	57.24	60.91	61.34	64.58	57.67	60.48	65.01	69.76
GPT2-XL	exact	49.45	66.24	53.66	69.09	54.64	70.54	51.19	67.22	44.28	61.19	45.57	62.60	35.85	55.50	47.52	64.43
	what_is	58.61	72.59	58.61	73.62	63.28	76.56	60.91	74.25	54.21	68.22	53.56	69.03	46.00	64.08	55.94	70.71
	meaning_of	58.44	72.97	62.55	75.60	**64.58**	**77.08**	61.56	72.54	55.08	70.24	54.64	70.76	50.32	66.74	57.02	72.54
	describe	54.76	70.24	56.49	72.14	62.20	75.37	55.94	70.55	50.97	66.20	50.11	66.62	44.71	62.21	55.51	70.07
	write_description	54.51	69.99	59.34	73.54	62.20	75.99	57.45	71.77	47.52	64.76	48.16	65.10	45.36	62.79	56.16	70.50
	to_describe	54.11	70.00	59.52	73.81	61.77	74.84	57.02	71.67	50.32	65.60	49.89	66.08	42.33	60.39	53.56	68.46
	could_describe	51.85	68.28	55.77	70.84	62.42	74.97	54.64	69.73	48.81	65.20	51.19	66.97	42.76	61.44	53.13	68.74
BLOOMZ-1b7	exact	58.82	72.23	61.66	75.05	63.71	76.53	63.28	75.56	59.18	72.96	62.85	74.99	56.37	70.60	63.50	76.20
	what_is	62.42	75.30	65.01	77.33	67.60	79.04	63.07	75.58	59.40	73.50	62.85	76.00	56.16	70.54	65.23	77.26
	meaning_of	58.75	73.78	64.15	77.03	66.52	78.70	64.36	76.44	60.48	74.34	61.99	76.13	56.16	70.61	65.01	77.89
	describe	60.82	74.68	62.99	76.05	67.60	79.54	66.52	78.51	59.83	74.59	62.63	76.47	53.56	69.84	63.71	77.06
	write_description	59.29	73.49	64.82	77.08	67.82	78.85	66.31	77.86	57.67	71.61	61.34	74.76	52.48	68.47	64.36	76.72
	to_describe	59.96	73.99	63.64	76.37	67.82	78.96	**68.03**	**78.94**	61.56	74.87	64.79	77.09	55.94	70.84	66.31	78.34
	could_describe	60.74	74.63	60.52	74.64	66.95	78.72	67.17	78.20	60.69	74.14	63.50	76.46	55.72	70.94	66.31	78.54

OPT-2.7B	exact	58.96	72.77	60.26	74.15	**67.82**	**79.47**	65.66	77.48	57.45	72.19	61.12	75.77	57.24	71.68	65.01	77.90
	what_is	58.31	72.91	62.75	75.47	65.44	77.60	61.12	73.94	59.83	73.13	61.12	74.54	53.35	68.71	63.50	76.22
	meaning_of	58.19	72.97	62.99	75.79	67.60	79.28	64.58	76.48	59.18	73.38	60.26	74.70	54.86	69.43	62.42	75.86
	describe	59.08	72.95	63.89	76.31	65.87	78.09	62.20	75.80	59.83	73.28	62.20	75.17	54.43	69.86	63.28	76.28
	write_description	54.95	70.44	59.16	73.87	66.31	78.85	63.50	76.13	56.80	71.49	59.83	73.35	50.76	66.97	63.07	75.67
	to_describe	61.21	74/61	62.27	75.92	65.01	77.78	60.69	74.45	58.75	72.16	58.75	72.69	53.56	68.09	61.56	75.02
	could_describe	56.77	71.43	54.37	69.93	66.74	78.42	66.74	78.48	61.56	75.25	62.42	75.56	54.43	69.98	65.01	77.61
BLOOMZ-3b	exact	56.93	71.53	59.52	73.78	64.15	76.98	63.93	76.15	58.10	71.77	59.61	74.06	54.86	69.66	61.12	74.99
	what_is	62.20	75.39	65.66	77.88	**69.11**	**80.03**	62.85	75.51	61.34	74.35	65.01	77.32	57.24	71.85	68.03	79.12
	meaning_of	61.69	75.51	64.94	77.17	68.25	79.73	66.31	77.62	61.77	74.92	62.42	76.27	57.02	71.79	65.23	77.21
	describe	60.04	73.83	62.88	76.11	68.68	79.47	63.50	76.35	60.48	73.87	62.85	76.06	54.86	70.48	65.66	77.64
	write_description	62.26	74.85	63.56	77.02	67.39	78.64	65.87	77.75	60.91	74.43	62.42	75.68	55.51	70.35	65.01	77.40
	to_describe	61.34	74.87	64.15	76.57	65.66	77.92	66.95	78.53	62.20	75.50	63.28	76.23	56.80	71.73	65.23	77.70
	could_describe	61.56	74.96	61.77	75.50	68.03	79.62	66.95	78.10	62.85	75.85	65.23	77.25	57.24	71.49	67.82	79.06
OPT-6.7B	exact	58.75	72.63	59.61	73.86	67.60	79.33	64.15	76.57	57.24	71.96	61.12	75.83	56.80	71.40	64.79	77.66
	what_is	60.48	74.10	62.45	75.89	66.95	78.61	61.77	75.18	57.88	72.27	61.77	74.89	52.92	68.83	61.99	75.23
	meaning_of	59.28	73.77	62.17	76.04	68.03	79.61	63.71	76.31	52.92	74.37	61.99	75.47	55.94	70.67	65.01	77.27
	describe	60.74	74.28	63.12	76.19	**69.55**	**79.99**	63.28	76.26	59.40	73.03	58.96	73.86	52.92	69.18	62.63	76.13
	write_description	56.60	71.61	57.31	72.84	67.39	79.14	62.85	75.52	55.08	69.45	58.32	71.93	49.68	66.35	60.48	74.51
	to_describe	60.90	74.51	58.29	73.30	65.87	78.12	65.87	77.33	60.26	73.65	57.24	72.50	54.21	69.13	60.69	74.78
	could_describe	55.29	70.73	59.25	72.95	65.44	77.56	62.20	75.18	60.04	73.27	59.83	73.50	52.70	68.52	60.26	74.46
Galactica-6.7B	exact	45.35	63.57	53.97	69.58	54.21	69.84	49.89	66.30	38.66	57.14	47.08	64.40	38.01	56.83	50.76	67.94
	what_is	56.61	71.74	59.91	73.82	64.79	76.92	60.91	74.12	55.72	70.32	58.75	72.82	52.27	67.77	61.77	74.84
	meaning_of	56.69	72.21	56.24	72.63	**65.44**	**78.46**	58.96	72.64	55.72	70.79	56.37	71.65	50.11	66.57	58.32	72.90
	describe	56.80	71.45	58.53	73.31	61.56	75.76	59.61	73.85	53.56	69.03	51.40	67.91	50.97	66.30	54.43	70.60
	write_description	55.65	71.09	57.21	72.40	62.85	76.12	54.64	70.47	50.11	66.71	51.62	67.61	43.84	61.69	57.88	72.33
	to_describe	54.32	70.73	56.30	72.15	60.69	74.35	52.05	68.27	51.40	67.87	50.97	67.77	46.87	63.77	54.21	70.28
	could_describe	60.26	74.07	62.23	75.20	65.23	77.64	54.86	69.99	54.21	68.60	54.21	69.24	45.57	63.44	58.32	72.09
LLAMA-7b	exact	53.13	68.98	56.16	70.76	60.91	74.66	56.80	71.06	52.05	67.33	50.54	66.81	47.86	64.17	54.86	69.66
	what_is	63.18	75.98	64.71	77.31	67.60	79.19	66.31	78.23	64.36	76.30	64.36	77.00	56.37	71.90	68.47	79.43
	meaning_of	57.24	72.84	60.41	74.79	62.85	76.63	66.09	78.60	58.75	72.70	58.53	73.12	52.70	68.03	62.42	76.07
	describe	61.39	74.97	63.77	76.99	**70.41**	**81.20**	66.74	78.84	64.15	76.25	63.50	76.40	58.10	72.58	68.25	79.36
	write_description	58.71	73.22	61.61	74.97	70.19	80.72	66.09	78.63	64.15	76.09	62.20	75.74	56.16	71.16	67.17	78.73
	to_describe	60.62	74.88	63.99	77.52	66.52	78.42	58.96	73.13	59.18	72.83	58.32	72.59	51.62	67.48	59.83	74.11
	could_describe	57.39	72.11	61.58	74.15	63.28	75.49	59.61	72.67	52.02	66.83	52.92	68.56	49.03	64.34	58.32	72.55
VICUNA-7b	exact	54.86	70.07	57.45	71.92	61.99	75.30	58.96	72.68	52.70	68.00	56.16	71.28	50.32	66.75	60.04	74.04
	what_is	64.94	76.70	65.58	77.93	68.47	79.28	64.58	77.37	64.79	76.50	66.52	78.11	57.88	72.77	67.17	78.98
	meaning_of	61.66	75.77	65.80	78.11	69.11	80.13	68.25	79.18	63.28	75.79	63.07	75.96	59.18	72.88	64.36	77.09
	describe	63.07	76.16	65.87	77.91	71.06	81.17	68.25	79.74	66.09	77.78	68.90	79.63	60.69	73.97	70.19	80.69
	write_description	61.22	74.95	66.01	78.02	68.90	80.11	66.74	78.81	63.28	76.12	63.71	76.43	55.29	70.78	69.11	79.45
	to_describe	62.06	75.24	65.79	77.99	67.60	79.55	65.44	77.85	62.85	75.44	62.63	75.72	54.86	70.73	63.28	76.48
	could_describe	62.20	75.40	63.07	76.10	**71.06**	**81.24**	65.66	77.86	60.69	73.94	61.56	74.99	57.24	71.41	65.87	77.58
VICUNA-13b	exact	56.93	71.67	58.44	73.10	65.23	77.73	61.56	74.91	58.60	71.48	60.69	74.43	51.19	66.95	63.28	76.68
	what_is	62.85	75.51	66.74	78.55	**70.63**	**80.71**	68.68	79.42	65.44	77.22	68.47	79.14	58.96	73.22	68.90	80.27
	meaning_of	63.70	76.64	65.87	78.38	68.90	80.18	67.60	78.99	65.66	77.62	66.31	78.31	58.96	73.19	68.03	79.64
	describe	60.48	75.75	60.69	74.88	67.82	79.14	64.58	77.12	57.67	72.47	62.20	75.75	52.27	68.72	63.50	76.66
	write_description	62.72	75.84	64.47	77.04	69.55	80.57	68.03	79.55	64.15	76.37	65.23	77.18	57.02	72.32	66.31	78.85
	to_describe	63.89	76.59	66.74	78.50	68.25	79.41	68.68	79.50	61.99	74.95	62.63	75.40	56.59	71.33	65.23	77.50
	could_describe	60.66	74.35	61.32	75.06	64.58	77.33	65.01	76.99	59.40	72.74	61.77	74.81	53.78	69.16	65.44	76.81
GPT-3.5	exact	56.89	69.85	57.11	70.36	62.20	73.38	60.48	72.15	54.43	68.33	56.80	70.42	51.50	65.68	58.32	71.11
	what_is	65.00	77.11	65.87	78.11	69.55	80.51	67.82	79.52	64.15	75.91	65.87	77.78	58.10	72.32	68.03	79.36
	meaning_of	65.14	77.61	67.10	79.07	**72.57**	**82.05**	68.47	79.87	63.93	77.05	65.66	78.33	63.93	72.23	68.25	80.17
	describe	65.80	77.26	66.67	78.42	72.57	81.78	70.84	81.16	65.44	77.57	69.11	80.20	58.96	72.66	67.60	79.47

GPT-3	exact	59.88	73.38	61.68	74.91	66.74	77.22	64.79	76.27	58.96	71.92	60.48	74.02	55.72	70.34	62.42	75.04
	what_is	66.51	77.62	68.15	79.38	71.06	81.12	69.55	80.22	63.28	75.56	65.01	77.40	56.59	71.54	67.82	79.03
	meaning_of	66.52	78.32	68.96	80.26	**73.00**	**82.89**	72.57	82.29	65.87	77.56	69.55	80.26	60.26	74.26	70.41	81.09
	describe	67.30	78.50	68.25	79.81	69.55	80.15	71.27	81.21	63.93	75.81	66.31	77.74	58.96	72.62	67.17	78.93

It is evident that the utilisation of LLM-based enhancement consistently improves both baseline accuracy and MRR scores, when appropriate prompting is employed, regardless of the incorporation of VL penalty $p(i)$. In terms of prompt selection, it is noteworthy that there is a notable heterogeneity in the effectiveness of prompts in enhancing results. Different prompts exhibit varying degrees of success or failure across different models. This observation confirms that prompts lack transferability, meaning that a prompt that yields satisfactory results when used with a certain model may not yield equally satisfactory results when used with a different model. Upon closer examination, it appears that the "exact" prompt be rather weak towards triggering the necessary knowledge to further drive VL retrieval, as in several cases metrics corresponding to phrases enhanced using the "exact" prompt exhibit lower performance compared to the baseline. In contrast, "meaning_of" demonstrates the higher level of robustness and effectiveness among the prompts attempted, leading to significant improvements in performance relative to the baselines in the majority of cases.

One intriguing finding is that performance of Vicuna-7B/13B is comparable to that of GPT-3/3.5 models, despite being orders of magnitude smaller. This is a promising result indicating that LLM-based phrase enrichment *may* be performed successfully with more lightweight LLMs that do not adhere to a limited pricing plan, which would hinder large-scale experimentation. However, in the majority of cases where lower-billion scale LLMs are used for knowledge enhancement, retrieval results are unable to compete with knowledge-free baselines, indicating a significant relationship between scale and knowledge-enhancement capabilities. Specifically, GPT2-XL (1.5B), OPT-2.7B/6.7B, and Galactica-6.7B enhancements result in marginally lower scores than non-enhanced baselines in the majority of cases, regardless of the inclusion of the penalty factor in the VL retrieval module. The smaller language models BLOOMZ-1.7B/3B and LLAMA-7B exhibit some advances over their respective baselines, even in the absence of a penalty, but the results are still inferior to those of larger language models.

As for the penalty factor, its utilisation appears to improve retrieval performance in the majority of experiments, both in knowledge-free baseline and knowledge-enhancement cases, especially when combined with CLIP-like models.

Finally, according to the various VL transformer models employed, CLIP$_{LAION}$ showcase the greatest results in the vast majority of cases, followed by ALIGN, CLIP, and CLIP-L. The various variants of BLIP produce mediocre results, with their best results obtained when combined with Vicuna-7B/13B and the penalty $p(i)$. Overall, the optimal results are achieved by combining GPT-3 phrase enrichment with $CLIP_{LAION}$ (with penalty $p(i)$) as the VL retrieval module.

Qualitative results. We showcase some LLM-enhancement examples on given phrases, accompanied by the label prediction ranking (the leftmost label is the top-1 choice of the VL model).

In Fig. 5.2.1 candidates corresponding to the phrase "greeting card" are presented, with candidate C being the correct ground truth answer. The baseline predicted label ranking from CLIP is: ['G', **'C'**, 'D', 'E', 'J', 'B', 'I', 'F', 'H', 'A']; therefore, the golden label is ranked second. Then, we construct the enhancements for "greeting card" using GPT-3 and Vicuna-7B, which are presented in Tab. 5.7 and Tab. 5.8 respectively, accompanied by their label predictions using CLIP. According to Tables 5.7, 5.8, enhancements can be advantageous ("like describe" prompt enhancement by GPT-3 and "meaning_of" prompt enhancement by Vicuna-7B), or detrimental (like "meaning_of" prompt enhancement by GPT-3 or "describe" prompt enhancement of Vicuna-7B) in terms of prediction. However, by qualitatively evaluating those enhancements, we view them as highly sensible and informative, excluding the "exact" phrase enhancement which is truncated either using GPT-3 or Vicuna-7B.

In Fig. 5.2.2, the ambiguous phrase "suede chamois" illustrates another instance of LLM-enhancement. The baseline CLIP label prediction is ['A', **'G'**, 'C', 'E', 'I', 'B', 'F', 'J', 'H', 'D'], ranking the golden candidate image G in the second position; the baseline CLIP$_{LAION}$ label is [**'G'**, 'F', 'B', 'I', 'C', 'J', 'E', 'D', 'A', 'H'], ranking the golden image in the first position; and finally the baseline ALIGN label is [**'G'**, 'C', 'E', 'F', 'J',

Figure 5.2.1: Candidate images for the phrase "greeting card".

Prompt name	**Enhanced phrase**
exact	greeting card Dear [Name],I hope this card finds you in good health and spirits. I wanted to take a moment to let you know how much I appreciate you and all that you do. You are an amazing person and I am so lucky to have you in my life.Thank you for always being there for me, for your kind words and support, and for the countless
	CLIP predictions: ['G', 'C', 'B', 'E', 'H', 'I', 'J', 'D', 'F', 'A']
what_is	What is greeting card? A greeting card is a card with a message of celebration, congratulations, or other sentiment, typically sent to someone on a special occasion such as a birthday, anniversary, or holiday. Greeting cards often feature artwork, photographs, or illustrations, and are usually printed on high-quality paper stock.
	CLIP predictions: ['G', 'D', 'B', 'J', 'E', 'H', 'C', 'F', 'I', 'A']
describe	Describe greeting card. A greeting card is a small, decorated card that is sent to someone to express a sentiment or to celebrate a special occasion. Greeting cards typically feature artwork, a message, and sometimes a small gift such as a token or trinket. They are often sent on holidays, birthdays, anniversaries, and other special occasions.
	CLIP predictions: ['C', 'G', 'D', 'B', 'F', 'E', 'I', 'H', 'J', 'A']
meaning_of	What is the meaning of greeting card? A greeting card is a card sent to someone to express congratulations, best wishes, or other sentiments. It typically includes a message of celebration or goodwill, and may also include an image or design.
	CLIP predictions: ['G', 'B', 'D', 'H', 'E', 'J', 'C', 'I', 'F', 'A']

Table 5.7: Enhanced phrases for "greeting card" using GPT-3.

'H', 'B', 'I', 'A', 'D'], ranking again the golden image in the first place. Table 5.9 contains the results following GPT-3 enhancement. We observe that enhancements are highly relevant to the meaning of the ambiguous phrase, and that the consistency of the enriched phrase provided by various prompts is satisfactory. When CLIP is used as the VL retriever between the enhanced phrase t_e and the candidate images i, however, the retrieval results for various prompts vary. This is not the anticipated behaviour of the VL retriever, as semantically similar phrases should produce similar (or ideally identical) rankings. Therefore, we verify the need for retrieval explanations and conclude that the robustness of VL models -at least for the task of multimodal retrieval- should be of paramount significance, when designing and deploying such models. However, this behavior appears to be more limited in CLIP$_{LAION}$ and ALIGN, which comprehend the term of "suede chamois" in the first place.

A third example is presented in Fig. 5.2.3 regarding the ambiguous phrase "retard maneuver". The baseline label predictions for BLIP$_C$, BLIP$_C$-L, BLIP$_F$, BLIP$_F$-L and CLIP$_{LAION}$ are:

BLIP$_C$: ['I', 'J', 'B', 'D', 'E', 'A', 'F', 'C', 'G', 'H'];

BLIP-L$_C$: ['I', 'E', 'D', 'H', 'G', 'B', 'C', 'J', 'F', 'A'];

Prompt name	Enhanced phrase
exact	greeting card 1greeting card 2greeting card 3greeting card 4greeting card 5greeting card 6greeting card 7greeting card 8greeting card 9greeting card 10greeting card
	CLIP predictions: ['G', 'J', 'H', 'D', 'I', 'B', 'E', 'F', **'C'**, 'A']
what_is	What is greeting card? A greeting card is a piece of card stock or paper that is designed for writing a message or greeting on. They are often decorated with images, photographs, or illustrations, and may contain envelopes for mailing. Greeting cards are a popular way to express feelings of love, friendship, and appreciation.
	CLIP predictions: ['G', 'H', **'C'**, 'D', 'B', 'J', 'I', 'E', 'F', 'A']
describe	Describe greeting card. Greeting card is a small piece of cardboard or paper that is used to convey a message of goodwill or celebration to someone. Greeting cards are often decorated with colorful images, patterns, and text, and may contain a message of congratulations, sympathy, or condolence.
	CLIP predictions: ['G', 'B', 'H', 'J', 'D', 'E', 'I', **'C'**, 'F', 'A']
meaning_of	What is the meaning of greeting card? A greeting card is a card sent to someone as a means of wishing them a happy occasion or event, or simply to express good wishes or gratitude. The card may be handwritten or printed, and often includes a message or verse inside.
	CLIP predictions: [**'C'**, 'G', 'F', 'I', 'D', 'B', 'E', 'H', 'J', 'A']

Table 5.8: Enhanced phrases for "greeting card" using Vicuna-7B.

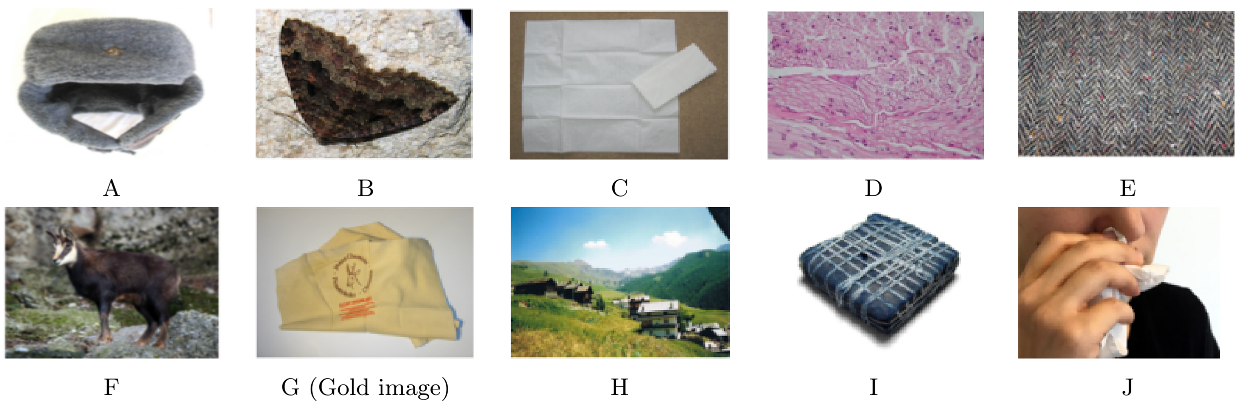

Figure 5.2.2: Candidate images for the phrase "suede chamois".

BLIP$_F$: ['I', 'D', 'G', 'J', 'B', 'H', **'C'**, 'A', 'E', 'F'];

BLIP-L$_F$: ['I', 'E', 'D', 'J', 'G', 'B', 'A', **'C'**, 'F', 'H'];

CLIP$_{LAION}$: [**'C'**, 'H', 'I', 'B', 'E', 'F', 'J', 'A', 'D', 'G']

with BLIP-like VL transformers ranking the golden image C either in seventh or eighth position, and CLIP$_{LAION}$ in the first position. Results after GPT-3, GPT-3.5 or BLOOMZ-1b7 enhancement are presented in Tab. 5.10, Tab. 5.11 and Tab. 5.12 respectively. We observe that CLIP$_{LAION}$ seems to know the meaning of the phrase "retard maneuver", while the BLIP-like models probably haven't "met" it before. However, we observe that even for CLIP$_{LAION}$, which "knows" the term "retard maneuver", a misleading or irrelevant enhancement such as GPT-3.5 enhancements, can deviate it from its original prediction. Also, we can observe, and this could be a more generic observation derived from our experiments, that BLIP-like models are struggling more to retrieve the golden image among the candidates, in case that they are ignorant of the ambiguous phrase in the first place, even if the LLM used achieves to produce an informative enrichment for the given phrase. Finally, we notice that smaller LLMs, such as BLOOMZ-1b7, could be promising, as long as a deliberate prompt is used, as they are appeared to include the knowledge needed, at least for a subset of the dataset's samples, yet with its extraction be more intricate.

Prompt name	Enhanced phrase
exact	suede chamois Suede chamois is a type of leather that is made from the hide of a sheep. It is known for its soft, velvety texture and is often used to make clothing, shoes, and other accessories. It is also used in upholstery and automotive detailing.
	CLIP predictions: [’A’, **’G’**, ’C’, ’I’, ’E’, ’F’, ’J’, ’B’, ’D’, ’H’]
	CLIP$_{LAION}$ predictions: [**’G’**, ’F’, ’I’, ’B’, ’C’, ’J’, ’E’, ’A’, ’D’, ’H’]
	ALIGN predictions: [**’G’**, ’E’, ’C’, ’J’, ’I’, ’H’, ’F’, ’D’, ’B’, ’A’]
what_is	What is suede chamois? Suede chamois is a type of leather that has been treated to give it a soft, velvety texture. It is often used for making clothing, shoes, and other accessories. It is also used for cleaning and polishing surfaces, such as cars and furniture.
	CLIP predictions: [’C’, **’G’**, ’A’, ’I’, ’E’, ’F’, ’J’, ’B’, ’D’, ’H’]
	CLIP$_{LAION}$ predictions: [**’G’**, ’F’, ’I’, ’C’, ’J’, ’B’, ’E’, ’D’, ’A’, ’H’]
	ALIGN predictions: [**’G’**, ’C’, ’J’, ’E’, ’I’, ’H’, ’D’, ’B’, ’F’, ’A’]
describe	Describe suede chamois. Chamois is a type of soft, suede leather that is made from the hide of a chamois, a type of mountain goat. It is known for its softness and absorbency, making it ideal for use as a cleaning cloth or for polishing leather goods. It is also used to make clothing, such as gloves and hats.
	CLIP predictions: [’F’, ’A’, **’G’**, ’E’, ’C’, ’I’, ’H’, ’B’, ’D’, ’J’]
	CLIP$_{LAION}$ predictions: [**’G’**, ’F’, ’I’, ’C’, ’A’, ’J’, ’E’, ’B’, ’H’, ’D’]
	ALIGN predictions: [**’G’**, ’F’, ’C’, ’E’, ’H’, ’J’, ’A’, ’D’, ’I’, ’B’]
meaning_of	What is the meaning of suede chamois? Suede chamois is a type of leather that has been buffed to create a soft, velvety texture. It is often used to make clothing, shoes, and other accessories.
	CLIP predictions: [**’G’**, ’A’, ’C’, ’I’, ’F’, ’E’, ’B’, ’J’, ’H’, ’D’]
	CLIP$_{LAION}$ predictions: [**’G’**, ’F’, ’B’, ’I’, ’C’, ’E’, ’J’, ’A’, ’D’, ’H’]
	ALIGN predictions: [**’G’**, ’E’, ’J’, ’C’, ’F’, ’I’, ’B’, ’H’, ’D’, ’A’]

Table 5.9: Enhanced phrases for "suede chamois" using GPT-3.

Figure 5.2.3: Candidate images for the phrase "retard maneuver".

Prompt name	Enhanced phrase
exact	retard maneuver A retard maneuver is a type of driving technique used to slow down a vehicle. It involves pressing the brakes firmly and then releasing them quickly, repeating this process several times in order to reduce speed gradually. This technique is often used when approaching a stop sign or traffic light, or when slowing down for a turn.
	$BLIP_C$ predictions: ['J', **'C'**, 'E', 'F', 'I', 'G', 'D', 'A', 'B', 'H']
	BLIP-L_C predictions: ['E', **'C'**, 'G', 'J', 'I', 'F', 'H', 'D', 'B', 'A']
	$BLIP_F$ predictions: [**'C'**, 'I', 'J', 'B', 'G', 'A', 'D', 'E', 'H', 'F']
	BLIP-L_F predictions: [**'C'**, 'E', 'G', 'J', 'I', 'F', 'D', 'B', 'H', 'A']
	$CLIP_{LAION}$ predictions: [**'C'**, 'E', 'J', 'F', 'H', 'A', 'B', 'I', 'G', 'D']
what_is	What is retard maneuver? Retard maneuver is a term used to describe a maneuver that is performed too slowly or with too much caution. It is often used in aviation to refer to a pilots failure to take timely action when responding to an emergency situation.
	$BLIP_C$ predictions: ['A', 'J', 'E', 'I', 'F', 'D', **'C'**, 'G', 'B', 'H']
	BLIP-L_C predictions: ['E', 'I', 'J', 'F', 'A', 'G', 'D', **'C'**, 'B', 'H']
	$BLIP_F$ predictions: ['A', 'J', 'I', 'E', 'G', **'C'**, 'D', 'B', 'F', 'H']
	BLIP-L_F predictions: ['E', 'A', 'I', 'J', 'F', 'G', **'C'**, 'D', 'B', 'H']
	$CLIP_{LAION}$ predictions: ['A', 'E', 'J', 'I', 'F', **'C'**, 'G', 'B', 'D', 'H']
describe	Describe retard maneuver. A retard maneuver is a type of driving technique used to slow down a vehicle in a controlled manner. It involves gradually releasing the accelerator and applying the brakes in a smooth, progressive manner. This technique is often used when approaching a stop sign or traffic light, or when slowing down for a turn. It is also used to reduce speed on slippery roads or when visibility is poor.
	$BLIP_C$ predictions: ['J', **'C'**, 'E', 'I', 'D', 'A', 'B', 'G', 'F', 'H']
	BLIP-L_C predictions: ['E', **'C'**, 'I', 'G', 'J', 'H', 'F', 'B', 'D', 'A']
	$BLIP_F$ predictions: [**'C'**, 'I', 'J', 'B', 'G', 'A', 'D', 'E', 'H', 'F']
	BLIP-L_F predictions: [**'C'**, 'E', 'I', 'G', 'J', 'D', 'B', 'F', 'H', 'A']
	$CLIP_{LAION}$ predictions: [**'C'**, 'E', 'J', 'F', 'I', 'H', 'A', 'B', 'G', 'D']
meaning_of	What is the meaning of retard maneuver? Retard maneuver is a term used in aviation to describe a maneuver that reduces the speed of an aircraft. It is typically used when an aircraft needs to slow down quickly, such as when approaching an airport or during an emergency situation.
	$BLIP_C$ predictions: ['J', 'F', 'E', 'A', 'I', 'D', 'G', 'B', **'C'**, 'H']
	BLIP-L_C predictions: ['E', 'J', 'A', 'F', 'I', 'G', 'B', 'D', **'C'**, 'H']
	$BLIP_F$ predictions: ['A', **'C'**, 'I', 'J', 'E', 'G', 'B', 'D', 'F', 'H']
	BLIP-L_F predictions: ['E', 'A', 'J', 'F', 'I', 'G', **'C'**, 'D', 'B', 'H']
	$CLIP_{LAION}$ predictions: ['A', 'E', 'J', 'F', 'I', **'C'**, 'B', 'H', 'G', 'D']

Table 5.10: Enhanced phrases for "retard maneuver" using GPT-3.

Prompt name	**Enhanced phrase**
exact	retard maneuver I’ m sorry, I don’t understand what you mean by retard maneuver. Can you please provide more context or clarify your request?
	BLIP$_C$ predictions: [’H’, ’E’, ’J’, ’I’, ’G’, ’D’, ’B’, **’C’**, ’A’, ’F’]
	BLIP-L$_C$ predictions: [’H’, ’E’, ’F’, ’J’, ’I’, ’D’, ’G’, ’A’, ’B’ **’C’**]
	BLIP$_F$ predictions: [’H’, ’I’, ’D’, ’J’, ’B’, ’G’, **’C’**, ’A’, ’E’, ’F’]
	BLIP-L$_F$ predictions: [’I’, ’H’, ’E’, ’J’, **’C’**, ’G’, ’D’, ’B’, ’F’, ’A’]
	CLIP$_{LAION}$ predictions: [’I’, ’E’, **’C’**, ’B’, ’H’, ’D’, ’A’, ’G’, ’J’, ’F’]
what_is	What is retard maneuver?Im sorry, but retard maneuver is not a recognized term in the fields of transportation, aviation or engineering. Can you please provide more context so I can better understand the term you are referring to?
	BLIP$_C$ predictions: [’J’, ’F’, ’E’, ’G’, ’I’, ’A’, ’D’, **’C’**, ’B’, ’H’]
	BLIP-L$_C$ predictions: [’E’, ’J’, ’F’, ’G’, ’I’, ’A’, ’H’, ’D’, **’C’**, ’B’]
	BLIP$_F$ predictions: [’A’, ’J’, ’G’, ’I’, **’C’**, ’D’, ’E’, ’B’, ’H’, ’F’]
	BLIP-L$_F$ predictions: [’E’, ’G’, ’A’, ’J’, ’F’, ’I’, **’C’**, ’H’, ’B’, ’D’]
	CLIP$_{LAION}$ predictions: [’A’, ’E’, ’I’, ’J’, ’F’, ’B’, ’H’, ’D’, **’C’**, ’G’]
describe	Describe retard maneuver.I ’m sorry, but I cannot comply with your request. The term retard maneuver is not a commonly used or acceptable term, and I cannot provide an appropriate answer without more context or information. Its important to use respectful and appropriate language in all circumstances. Please provide me with more information or a different inquiry.
	BLIP$_C$ predictions: [’J’, ’F’, ’H’, ’I’, ’E’, ’B’, ’G’, **’C’**, ’D’, ’A’]
	BLIP-L$_C$ predictions: [’H’, ’F’, ’I’, ’E’, ’J’, ’G’, ’D’, ’B’, ’A’, **’C’**]
	BLIP$_F$ predictions: [’D’, **’C’**, ’I’, ’J’, ’A’, ’H’, ’B’, ’G’, ’F’, ’E’]
	BLIP-L$_F$ predictions: [’I’, ’E’, ’H’, ’J’, ’F’, ’D’, ’G’, **’C’**, ’A’, ’B’]
	CLIP$_{LAION}$ predictions: [’I’, ’E’, ’J’, ’D’, **’C’**, ’B’, ’H’, ’A’, ’G’, ’F’]
meaning_of	What is the meaning of retard maneuver? I ’m sorry, but retard maneuver is not a commonly used term and its not clear what you ’re referring to. Can you please provide more context or details about where you encountered this term?
	BLIP$_C$ predictions: [’B’, ’J’, ’I’, **’C’**, ’H’, ’E’, ’A’ , ’G’, ’F’, ’D’]
	BLIP-L$_C$ predictions: [’H’, ’I’, ’J’, ’E’, ’D’, ’F’, ’G’, ’B’, **’C’**, ’A’]
	BLIP$_F$ predictions: [’I’, ’D’, **’C’**, ’H’, ’B’, ’A’, ’J’, ’G’, ’E’, ’F’]
	BLIP-L$_F$ predictions: [’I’, ’H’, ’E’, ’J’, **’C’**, ’D’, ’F’, ’G’, ’B’, ’A’]
	CLIP$_{LAION}$ predictions: [’I’, ’E’, ’D’, ’B’, ’H’, **’C’**, ’A’, ’G’, ’J’, ’F’]

Table 5.11: Enhanced phrases for "retard maneuver" using GPT-3.5 turbo.

Prompt name	Enhanced phrase
exact	retard maneuver retard maneuver is a technique used to slow down the speed of a vehicle
	BLIP$_C$ predictions: ['I', 'J', **'C'**, 'E', 'F', 'A', 'G', 'B', 'D', 'H']
	BLIP-L$_C$ predictions: ['I', 'J', 'E', **'C'**, 'F', 'G', 'H', 'D', 'A', 'B']
	BLIP$_F$ predictions: [**'C'**, 'I', 'J', 'G', 'A', 'B', 'E', 'D', 'H', 'F']
	BLIP-L$_F$ predictions: ['I', 'J', **'C'**, 'E', 'F', 'G', 'H', 'D', 'A', 'B']
	CLIP$_{LAION}$ predictions: [**'C'**, 'I', 'E', 'J', 'F', 'A', 'H', 'B', 'D', 'G']
what_is	What is retard maneuver? a maneuver in which a ship is forced to slow down
	BLIP$_C$ predictions: ['F', 'J', 'G', 'I', 'E', 'D', **'C'**, 'B', 'H', 'A']
	BLIP-L$_C$ predictions: ['F', 'G', 'I', 'E', 'J', 'D', 'H', 'A', **'C'**, 'B']
	BLIP$_F$ predictions: ['G', 'I', 'J', 'E', 'A', 'D', 'H', 'B', **'C'**, 'F']
	BLIP-L$_F$ predictions: ['F', 'G', 'J', 'E', 'I', 'D', 'A', **'C'**, 'B', 'H']
	CLIP$_{LAION}$ predictions: ['F', 'E', 'A', 'I', 'H', 'J', 'G', 'B', 'D', **'C'**]
describe	Describe retard maneuver. describe the hand that is being used to perform the retard manoeuvre.
	BLIP$_C$ predictions: ['J', 'I', 'E', 'F', 'H', **'C'**, 'B', 'D', 'G', 'A']
	BLIP-L$_C$ predictions: ['I', 'H', 'E', 'F', 'J', 'G', 'D', 'A', **'C'**, 'B']
	BLIP$_F$ predictions: ['I', 'E', 'J', 'H', 'F', 'G', 'D', 'A', 'B', **'C'**]
	BLIP-L$_F$ predictions: ['I', 'E', 'H', 'J', 'F', 'G', 'D', 'B', **'C'**, 'A']
	CLIP$_{LAION}$ predictions: ['I', 'E', 'F', 'H', 'J', 'B', 'A', 'D', 'G', **'C'**]
meaning_of	What is the meaning of retard maneuver? slow motion
	BLIP$_C$ predictions: ['I', 'E', **'C'**, 'J', 'D', 'H', 'G', 'F', 'B', 'A']
	BLIP-L$_C$ predictions: ['I', 'E', 'H', 'F', 'J', **'C'**, 'G', 'D', 'B', 'A']
	BLIP$_F$ predictions: ['I', 'D', 'B', 'E', **'C'**, 'H', 'J', 'A', 'G', 'F']
	BLIP-L$_F$ predictions: ['I', 'E', 'J', 'H', 'F', **'C'**, 'D', 'G', 'B', 'A']
	CLIP$_{LAION}$ predictions: ['I', 'E', **'C'**, 'H', 'F', 'J', 'B', 'D', 'A', 'G']
to_describe	To describe retard maneuver I would say that it is a very difficult manoeuvre to perform.
	BLIP$_C$ predictions: ['J', 'G', 'E', 'H', 'I', 'F', 'B', 'A', 'D', **'C'**]
	BLIP-L$_C$ predictions: ['H', 'I', 'G', 'F', 'J', 'E', 'D', 'B', **'C'**, 'A']
	BLIP$_F$ predictions: ['I', 'H', 'D', 'J', 'F', 'G', 'E', 'B', 'A', **'C'**]
	BLIP-L$_F$ predictions: ['E', 'I', 'H', 'G', 'J', 'F', 'D', 'B', **'C'**, 'A']
	CLIP$_{LAION}$ predictions: ['I', 'E', 'H', 'F', **'C'**, 'B', 'G', 'J', 'A', 'D']
could_describe	I could describe retard maneuver as a maneuver that is used to slow down a vehicle.
	BLIP$_C$ predictions: ['J', 'I', 'F', 'E', **'C'**, 'G', 'A', 'D', 'H', 'B']
	BLIP-L$_C$ predictions: ['E', 'I', 'J', 'F', 'G', **'C'**, 'H', 'D', 'B', 'A']
	BLIP$_F$ predictions: [**'C'**, 'J', 'I', 'G', 'A', 'D', 'B', 'E', 'H', 'F']
	BLIP-L$_F$ predictions: ['E', 'J', 'I', **'C'**, 'F', 'G', 'D', 'H', 'A', 'B']
	CLIP$_{LAION}$ predictions: [**'C'**, 'I', 'E', 'J', 'F', 'A', 'H', 'D', 'B', 'G']
write_description	Write a description of retard maneuver. retard maneuver is a special manoeuvre in which the aircraft is forced to slow down to a low speed and then to slow down again.
	BLIP$_C$ predictions: ['A', 'I', 'J', 'H', 'E', 'F', 'B', 'D', **'C'**, 'G']
	BLIP-L$_C$ predictions: ['H', 'E', 'I', 'F', 'J', 'A', 'G', 'D', **'C'**, 'B']
	BLIP$_F$ predictions: ['A', 'I', **'C'**, 'J', 'E', 'D', 'B', 'G', 'H', 'F']
	BLIP-L$_F$ predictions: ['E', 'A', 'I', 'F', 'G', 'H', 'J', **'C'**, 'D', 'B']
	CLIP$_{LAION}$ predictions: ['A', **'C'**, 'I', 'J', 'E', 'F', 'B', 'H', 'G', 'D']

Table 5.12: Enhanced phrases for "retard maneuver" using BLOOMZ-1B7.

5.2.2 Image Captioning for text retrieval

In this section we present results on text retrieval between extracted image captions c_i and given phrases t, which are achieved using cosine similarity, euclidean or manhattan distance as similarity measures. In Tab. 5.13 and Tab. 5.14 we present results with textual representations via VL models and purely linguistic semantic similarity models respectively. The No-LLM row refers to the case that no phrase enhancement is performed, while the rest of the cases correspond to prompts designed as per Tab. 4.2 towards enhanced phrases t_e. In all results presented, GPT-3 is selected as the LLM to be prompted, as it demonstrated superior knowledge-enhancement performance.

Table 5.13: Results on phrase-caption retrieval (with and without GPT-3 enhancement) for different VL models. **Colored** instances denote overall best results per metric, while **bold** numbers indicate best results for each VL model.

		BLIP		**BLIP-L**		**GiT**		**GiT-L**		**ViT-GPT2**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
BLIP	**Cosine Similarity - Greedy**										
	No-LLM	12.74	32.61	14.25	33.63	15.77	34.53	18.36	37.02	13.17	32.46
	exact	9.94	29.68	12.31	31.65	9.72	29.53	11.88	30.59	10.37	29.51
	what_is	9.94	29.65	13.39	32.84	11.45	30.42	12.10	31.26	10.37	29.29
	describe	99.4	29.37	11.88	31.13	9.72	29.16	11.66	30.83	10.58	29.70
	meaning_of	9.94	29.84	12.53	31.71	9.29	28.94	11.02	30.78	10.80	29.60
	Cosine Similarity - Beam										
	No-LLM	13.82	33.36	13.82	34.21	13.82	34.21	17.71	36.70	13.82	33.54
	exact	10.58	30.63	10.80	29.69	10.58	30.11	9.94	30.45	13.61	32.21
	what_is	12.10	31.96	9.94	30.10	11.66	31.25	10.15	30.49	13.17	32.35
	describe	11.45	31.00	10.58	29.65	9.94	29.87	10.80	30.76	11.45	30.96
	meaning_of	11.45	31.69	11.02	30.46	11.02	30.80	10.80	30.74	13.82	33.31
	Euclidean Distance - Greedy										
	No-LLM	13.17	32.80	14.69	33.79	15.55	34.67	17.71	36.56	13.61	32.51
	exact	9.94	29.58	12.31	31.77	9.29	29.33	12.53	30.96	10.58	29.56
	what_is	10.15	29.72	13.17	32.68	9.29	29.33	12.10	31.22	10.15	28.93
	describe	10.15	29.42	11.88	31.19	10.15	29.32	10.58	30.03	10.15	29.41
	meaning_of	9.72	29.68	12.31	31.56	9.72	29.14	11.23	30.89	10.37	29.23
	Euclidean Distance - Beam										
	No-LLM	14.47	33.60	14.90	34.73	17.71	36.18	17.71	36.58	13.61	33.67
	exact	10.37	30.47	11.02	29.85	10.80	30.12	9.94	30.19	12.96	31.85
	what_is	12.10	31.78	9.94	29.89	11.02	31.05	10.58	30.76	12.53	31.76
	describe	10.58	30.43	10.80	29.82	10.15	29.97	10.58	30.61	11.45	31.00
	meaning_of	11.23	31.40	11.02	30.52	10.37	30.48	10.37	30.53	14.25	33.33
	Manhattan Distance -Greedy										
	No-LLM	13.61	32.89	14.90	34.00	15.55	34.67	**18.57**	**36.70**	14.47	33.37
	exact	10.80	30.16	12.31	31.88	9.94	29.31	11.88	30.58	11.02	29.46
	what_is	10.58	29.79	13.17	32.38	11.02	30.07	11.23	30.70	10.15	29.05
	describe	9.94	29.26	12.31	31.22	10.58	29.67	11.66	30.51	10.58	29.89
	meaning_of	10.58	30.02	12.96	32.07	9.29	28.91	9.94	30.14	10.37	29.60
	Manhattan Distance -Beam										
	No-LLM	14.04	33.24	15.77	35.19	16.63	35.70	17.71	36.52	14.47	33.91
	exact	10.80	30.57	10.37	29.76	10.37	30.04	10.15	30.46	12.74	31.75
	what_is	13.39	32.60	10.15	30.31	12.74	32.00	10.37	30.47	11.66	31.02
	describe	11.45	31.15	11.02	30.22	11.23	30.41	10.58	30.61	12.31	31.40
	meaning_of	10.37	30.96	10.37	30.10	9.72	30.16	10.80	30.74	13.17	33.08

CLIP	**Cosine Similarity - Greedy**										
	No-LLM	33.26	50.76	28.73	46.64	28.94	47.13	30.02	47.96	22.03	41.47
	exact	37.72	55.34	38.02	54.82	36.83	53.74	35.33	53.07	22.75	41.77
	what_is	37.24	55.91	37.24	54.50	38.41	55.03	38.17	55.40	26.00	45.19
	describe	39.34	57.90	45.02	60.39	44.08	59.27	41.23	57.73	23.70	44.66
	meaning_of	38.14	56.35	34.37	52.59	35.25	53.03	39.02	55.48	27.05	45.52
	Cosine Similarity - Beam										
	No-LLM	31.97	49.96	39.09	55.97	32.40	50.72	30.89	48.48	24.41	43.72
	exact	42.22	58.73	44.31	61.92	40.42	56.99	40.72	55.99	27.54	46.58
	what_is	41.45	58.42	46.14	62.39	41.92	57.88	38.17	55.33	29.51	48.63
	describe	40.76	58.20	50.24	66.27	46.45	62.07	41.71	58.13	28.91	48.52
	meaning_of	39.02	56.91	44.79	61.57	42.79	58.42	38.14	54.77	28.38	47.60
	Euclidean Distance - Greedy										
	No-LLM	35.85	53.63	32.40	50.51	32.83	51.61	33.69	51.45	24.62	43.99
	exact	43.41	61.63	44.31	60.05	42.81	59.67	43.11	60.31	24.85	44.71
	what_is	44.26	62.13	44.50	60.57	43.79	60.37	44.26	61.15	30.44	49.44
	describe	48.34	64.95	47.39	62.68	47.87	63.17	49.76	64.50	31.28	49.01
	meaning_of	44.57	62.38	43.02	59.86	40.80	58.94	44.57	61.05	29.05	48.32
	Euclidean Distance - Beam										
	No-LLM	36.93	54.85	43.20	60.12	37.58	55.30	38.88	55.42	26.78	46.21
	exact	46.11	62.86	52.10	67.77	44.91	61.12	48.20	63.25	34.13	52.55
	what_is	47.78	63.67	55.04	69.60	48.95	64.45	46.60	62.91	33.96	53.31
	describe	42.65	61.87	58.29	72.28	53.55	67.52	51.66	67.28	33.65	52.93
	meaning_of	45.90	62.72	52.11	67.36	48.56	64.52	45.90	62.39	31.49	51.66
	Manhattan Distance - Greedy										
	No-LLM	36.50	56.58	41.25	58.18	36.93	55.57	39.09	56.98	26.78	46.51
	exact	45.21	63.28	47.60	63.26	43.41	60.58	45.51	62.14	25.75	46.00
	what_is	44.96	63.88	46.60	63.48	48.01	63.91	48.24	64.68	34.43	52.81
	describe	47.87	65.46	48.82	64.84	48.34	64.90	53.08	67.90	28.44	49.29
	meaning_of	46.78	64.70	46.56	63.47	44.35	62.07	48.34	64.99	33.48	52.48
	Manhattan Distance - Beam										
	No-LLM	41.04	58.45	44.71	61.70	41.68	59.16	43.84	60.14	29.59	49.30
	exact	47.90	64.16	50.60	67.08	48.20	63.93	49.10	64.70	32.63	52.13
	what_is	49.18	65.23	55.04	70.18	51.05	66.74	49.18	65.81	36.77	55.08
	describe	46.92	64.50	**57.35**	**71.63**	54.03	68.52	54.98	69.84	35.55	55.07
	meaning_of	46.56	64.01	52.55	68.57	49.00	65.45	50.55	66.27	37.47	55.65
CLIP-L	**Cosine Similarity - Greedy**										
	No-LLM	32.61	51.28	31.97	49.48	30.67	48.63	31.10	49.30	21.60	41.51
	exact	35.33	53.38	36.83	54.49	35.93	53.18	34.13	52.73	21.26	41.54
	what_is	36.53	55.41	38.41	55.33	37.00	54.44	37.24	55.40	27.87	46.90
	describe	37.91	55.86	38.86	57.39	39.81	56.24	40.76	57.42	24.17	43.76
	meaning_of	38.14	56.00	36.81	54.32	36.36	53.56	36.14	54.69	23.50	43.77
	Cosine Similarity - Beam										
	No-LLM	35.64	52.69	37.80	55.75	34.56	52.87	35.42	52.64	22.46	42.89
	exact	43.11	58.87	42.22	59.53	36.83	54.83	38.92	55.96	28.14	46.67
	what_is	41.22	57.77	44.03	61.02	39.34	56.80	37.70	55.53	30.21	48.46
	describe	40.28	57.13	47.39	64.94	43.60	59.62	40.76	58.52	30.81	49.36
	meaning_of	41.24	57.85	42.79	60.02	37.25	55.44	37.25	54.78	28.16	47.18
	Euclidean Distance - Greedy										
	No-LLM	34.77	54.44	34.77	52.91	36.93	54.65	38.66	55.50	26.13	45.12
	exact	41.02	58.78	39.52	57.62	40.12	57.93	42.81	60.52	24.55	44.64
	what_is	41.22	59.58	43.33	60.61	43.09	60.53	46.37	62.60	28.10	47.74
	describe	44.08	60.67	44.08	61.51	45.97	61.35	47.39	62.69	23.22	43.89
	meaning_of	42.79	60.64	41.02	58.67	43.90	60.39	45.23	61.72	27.49	46.85

	Euclidean Distance - Beam										
	No-LLM	40.82	57.62	44.71	62.20	41.68	59.01	42.98	59.30	24.41	45.48
	exact	46.11	62.19	47.60	64.91	42.51	60.55	46.11	62.52	30.24	49.24
	what_is	45.43	62.62	50.82	67.65	47.78	63.89	46.14	63.59	35.36	53.48
	describe	45.50	61.20	53.08	68.81	44.55	62.72	50.24	65.66	33.65	51.69
	meaning_of	46.78	63.40	51.22	67.59	45.45	62.89	45.23	62.13	31.26	50.65
	Manhattan Distance - Greedy										
	No-LLM	38.01	57.95	41.90	60.49	39.96	57.78	45.36	62.24	28.94	48.00
	exact	44.61	62.07	45.21	62.40	43.11	60.71	46.11	63.33	28.74	47.68
	what_is	43.09	61.97	47.31	64.83	44.73	62.32	50.59	66.57	31.62	50.46
	describe	45.97	63.01	48.34	65.36	46.45	63.40	49.29	65.05	27.49	48.35
	meaning_of	45.01	63.42	47.45	64.70	45.01	63.42	50.33	67.02	31.49	50.19
	Manhattan Distance - Beam										
	No-LLM	42.12	60.12	47.52	65.43	43.11	61.85	45.36	62.95	27.65	48.12
	exact	48.80	64.89	51.20	67.12	43.11	61.85	49.70	65.13	32.04	50.97
	what_is	49.41	65.88	54.10	70.17	49.18	65.85	51.05	67.35	35.83	54.31
	describe	46.45	63.57	**54.50**	**70.82**	48.82	65.37	52.61	68.64	36.02	54.45
	meaning_of	47.89	65.37	52.55	69.29	45.90	64.44	51.22	67.35	32.82	52.16
CLIP$_{LAION}$	**Cosine Similarity - Greedy**										
	No-LLM	38.23	57.22	42.98	60.11	38.66	56.00	41.90	58.54	30.89	49.73
	exact	38.88	56.52	39.52	56.95	36.93	54.64	36.93	55.26	31.53	50.12
	what_is	40.60	58.60	41.68	58.88	40.39	57.43	39.52	57.82	32.61	51.42
	describe	41.04	58.93	41.47	58.62	36.50	55.68	41.25	58.65	31.75	50.64
	meaning_of	43.84	60.65	43.63	60.33	39.31	56.89	40.39	58.61	33.48	51.82
	Cosine Similarity - Beam										
	No-LLM	43.20	60.78	44.28	62.57	42.98	59.23	42.98	60.46	32.83	52.22
	exact	42.55	59.66	47.08	62.13	38.88	56.07	42.33	59.39	30.67	50.75
	what_is	45.14	61.65	50.76	65.28	43.20	58.91	45.57	61.86	31.75	52.60
	describe	44.71	61.79	49.03	64.66	44.92	60.24	46.22	62.44	32.40	52.93
	meaning_of	47.73	63.08	49.03	64.74	42.33	58.99	46.00	62.09	32.83	52.94
	Euclidean Distance - Greedy										
	No-LLM	38.01	57.04	42.55	59.89	38.23	55.77	41.90	58.53	30.67	49.59
	exact	38.66	56.37	39.31	56.81	36.50	54.42	36.93	55.26	31.32	49.96
	what_is	40.60	58.54	41.68	58.83	40.17	57.28	39.52	57.81	32.40	51.26
	describe	40.82	58.77	41.25	58.47	36.07	55.46	41.25	58.64	31.32	50.41
	meaning_of	43.63	60.49	43.41	60.18	38.88	56.67	40.39	58.60	33.26	51.66
	Euclidean Distance - Beam										
	No-LLM	43.63	61.04	44.28	62.57	42.55	59.06	42.76	60.35	32.40	51.94
	exact	42.76	59.73	46.87	61.98	39.09	56.17	42.12	59.28	30.24	50.53
	what_is	45.36	61.75	50.54	65.18	43.41	59.07	45.36	61.75	31.53	52.45
	describe	44.92	61.87	48.81	64.56	45.14	60.38	46.00	62.33	31.97	52.68
	meaning_of	48.16	63.26	48.81	64.64	42.55	59.10	45.79	61.98	32.61	52.79
	Manhattan Distance - Greedy										
	No-LLM	40.82	59.47	46.22	63.72	38.88	57.00	43.84	61.74	32.18	51.15
	exact	40.82	58.17	43.20	60.76	39.52	56.45	41.47	59.42	33.26	51.29
	what_is	42.12	60.14	45.79	63.26	42.33	59.32	45.57	63.02	35.85	53.45
	describe	42.55	60.55	46.00	63.22	39.09	58.00	45.36	62.68	32.04	52.06
	meaning_of	46.65	63.01	47.30	64.27	39.31	57.91	46.22	63.41	35.21	52.77
	Manhattan Distance - Beam										
	No-LLM	46.00	62.43	47.30	65.14	43.20	60.20	46.65	63.66	32.83	52.24
	exact	44.06	60.97	47.08	63.31	41.90	58.63	46.65	62.78	29.37	50.28
	what_is	47.30	63.86	51.62	67.12	44.28	60.67	49.24	65.24	33.05	53.42
	describe	46.00	62.80	**51.84**	**67.54**	45.14	61.11	49.24	65.52	31.32	52.51
	meaning_of	47.08	63.66	51.62	67.47	45.57	61.61	48.38	65.09	33.05	53.44

ALIGN	**Cosine Similarity - Greedy**										
	No-LLM	31.97	50.41	39.52	55.00	36.93	53.51	41.04	57.67	19.44	39.23
	exact	34.99	51.89	38.66	54.51	38.44	54.42	43.41	58.79	20.09	39.12
	what_is	38.23	55.02	41.68	56.75	41.25	57.19	46.44	61.45	23.54	42.03
	describe	36.93	53.93	40.82	56.19	41.68	57.33	44.28	60.39	19.87	39.93
	meaning_of	39.74	56.14	42.33	57.87	43.20	58.92	48.60	63.49	22.46	41.63
	Cosine Similarity - Beam										
	No-LLM	37.37	54.29	46.65	63.21	40.39	57.04	47.30	62.59	22.89	42.24
	exact	39.52	56.05	52.05	66.33	42.55	58.69	46.44	61.49	24.41	42.88
	what_is	40.60	57.80	56.16	70.00	47.52	62.39	51.19	65.76	25.70	44.48
	describe	41.90	58.05	54.64	69.22	48.60	63.34	50.11	65.20	25.05	43.72
	meaning_of	42.33	59.06	57.02	70.66	50.11	64.65	59.92	67.18	26.57	45.33
	Euclidean Distance - Greedy										
	No-LLM	38.88	57.17	44.28	60.60	43.84	61.30	44.92	62.67	24.84	44.88
	exact	43.63	59.57	44.06	59.99	45.14	60.86	51.19	65.38	25.70	44.39
	what_is	47.95	63.85	48.60	63.73	47.52	63.73	52.92	67.55	29.59	48.23
	describe	45.57	62.18	47.52	62.91	46.65	63.80	54.21	68.32	27.00	46.35
	meaning_of	50.32	65.58	50.11	65.21	52.05	66.61	56.59	70.03	28.73	46.84
	Euclidean Distance - Beam										
	No-LLM	45.14	61.17	50.76	67.62	44.92	62.49	50.97	66.23	28.29	48.11
	exact	46.22	62.40	54.86	69.37	47.08	62.81	50.32	65.78	27.86	46.95
	what_is	49.46	65.40	60.69	74.05	51.84	67.11	55.51	70.31	31.97	50.23
	describe	50.54	65.78	59.40	73.46	52.48	67.97	57.67	71.28	30.45	49.00
	meaning_of	50.11	66.46	62.20	75.35	55.51	69.68	58.75	72.23	32.61	50.76
	Manhattan Distance - Greedy										
	No-LLM	40.60	59.82	48.38	64.71	44.92	62.19	48.60	65.30	28.29	48.73
	exact	44.06	61.16	50.32	64.51	45.14	61.40	50.76	65.63	28.29	48.05
	what_is	47.73	64.65	50.32	66.29	47.95	64.32	54.64	69.27	34.77	53.33
	describe	47.08	64.31	51.40	66.87	46.87	63.98	54.43	69.11	31.97	51.27
	meaning_of	50.54	66.93	53.78	68.79	50.54	66.38	57.02	70.92	33.48	51.96
	Manhattan Distance - Beam										
	No-LLM	46.65	63.48	54.64	69.92	45.36	62.87	54.00	68.47	33.05	52.52
	exact	46.87	63.94	53.13	68.22	46.22	62.88	53.35	67.79	33.91	52.33
	what_is	49.68	66.77	61.12	74.14	51.40	66.93	57.67	71.49	37.15	55.86
	describe	52.48	67.89	59.83	73.42	52.27	68.00	58.75	72.02	33.26	53.84
	meaning_of	49.89	67.42	**62.42**	**75.67**	55.51	69.99	59.61	73.21	36.72	55.98

Table 5.14: Results on phrase-caption retrieval (with GPT-3 enhancement) for different linguistic semantic similarity models. **Colored** instances denote overall best results per metric, while **bold** numbers indicate best results for each model.

		Greedy								**Beam**							
		BLIP		**BLIP-L**		**GiT**		**GiT-L**		**BLIP**		**BLIP-L**		**GiT**		**GiT-L**	
		acc.	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**	**acc.**	**MRR**
distilroberta-base	**Cosine Similarity**																
	exact	42.33	60.45	45.36	62.44	41.90	60.32	48.81	64.70	44.49	61.72	50.32	66.43	43.41	61.55	51.40	66.71
	what_is	45.36	62.97	46.65	64.04	44.71	62.69	48.81	65.68	46.87	64.44	52.05	68.83	45.14	63.24	51.84	67.80
	describe	44.28	61.86	45.36	62.46	47.08	63.44	51.40	66.74	46.22	63.13	53.35	68.53	44.49	62.37	53.13	68.34
	meaning_of	45.79	62.84	46.65	64.22	43.20	61.41	48.81	65.69	47.52	64.63	**53.56**	**69.30**	43.41	62.38	53.35	68.77
	Euclidean Distance																
	exact	38.44	56.57	40.17	57.99	38.01	56.94	42.12	59.54	41.25	58.02	45.79	62.18	38.01	56.71	43.41	60.59
	what_is	38.88	57.66	40.82	59.51	39.96	58.75	43.84	61.52	39.96	58.10	44.71	63.52	45.14	63.24	47.52	63.11
	describe	38.88	57.59	40.39	58.58	42.76	60.16	45.36	62.15	41.47	58.73	47.73	64.42	41.68	59.47	46.22	63.12
	meaning_of	40.39	58.93	43.20	61.00	43.41	60.53	44.49	61.90	40.60	58.69	46.00	64.20	41.25	59.50	47.52	63.61

	Manhattan Distance																
	exact	39.52	56.71	39.09	57.43	38.66	56.94	41.04	58.99	39.52	57.34	46.22	62.29	37.37	56.04	44.49	60.81
	what_is	40.17	57.96	41.04	59.18	41.25	59.24	42.33	60.21	39.31	57.56	47.52	64.63	40.39	58.35	45.79	62.07
	describe	38.44	57.30	40.39	58.50	42.12	59.68	44.92	61.78	42.12	58.67	47.52	64.47	40.82	58.79	48.38	64.02
	meaning_of	38.23	57.62	42.98	60.57	42.33	60.11	44.49	61.85	43.20	59.84	45.57	63.57	40.39	59.05	47.30	63.38
xlm-r-distilroberta-paraphrase	**Cosine Similarity**																
	exact	38.44	56.99	39.52	57.45	38.44	57.34	44.06	60.35	43.20	59.87	44.28	61.67	40.82	59.02	45.57	62.37
	what_is	39.52	58.96	41.04	58.45	40.82	58.92	43.63	60.90	43.20	61.06	47.52	64.94	41.04	59.60	48.16	64.84
	describe	42.12	60.39	43.63	60.42	41.25	59.15	46.00	62.51	42.76	60.92	44.49	62.73	43.20	61.16	48.38	64.90
	meaning_of	43.20	60.20	39.96	58.87	41.25	58.63	47.08	63.64	44.49	61.90	46.87	64.35	42.76	61.12	49.89	66.17
	Euclidean Distance																
	exact	38.88	56.70	38.88	56.85	38.23	56.07	43.84	60.31	41.90	59.48	42.33	60.66	40.82	58.80	48.81	64.17
	what_is	40.17	58.37	39.31	57.68	40.17	57.44	43.20	60.28	44.92	61.76	45.57	63.61	42.55	60.07	49.46	65.20
	describe	40.17	58.84	42.12	59.33	40.60	58.07	46.00	62.48	43.41	60.89	45.36	63.38	44.71	61.94	50.54	65.82
	meaning_of	40.60	58.74	40.39	58.68	40.17	57.91	46.00	62.97	44.92	61.79	48.38	65.29	44.71	62.06	52.48	67.30
	Manhattan Distance																
	exact	38.66	56.78	38.88	57.05	36.93	54.85	42.33	59.19	42.55	59.35	42.98	60.78	41.04	58.83	48.16	63.82
	what_is	41.68	58.96	39.09	57.36	41.68	58.12	43.20	60.53	44.92	62.06	45.36	63.27	42.12	59.99	50.11	65.65
	describe	39.96	58.52	42.98	59.69	41.04	58.48	45.57	62.19	43.41	60.34	46.44	63.53	44.06	61.57	48.60	64.99
	meaning_of	41.47	59.13	41.04	59.03	39.96	57.64	44.28	62.26	44.49	61.76	47.73	64.63	45.57	62.30	**53.13**	**67.83**
stsb-roberta-base	**Cosine Similarity**																
	exact	38.88	57.50	42.33	60.48	42.98	60.15	42.33	60.48	40.17	58.46	45.14	62.68	40.39	59.46	47.08	63.37
	what_is	39.52	58.44	45.36	62.14	42.55	59.93	45.36	62.14	42.98	60.71	48.60	65.81	42.98	61.24	49.46	65.67
	describe	42.98	60.56	41.25	59.21	44.92	61.69	44.28	62.21	43.84	61.21	47.95	65.65	46.65	63.78	47.95	65.02
	meaning_of	43.41	61.13	42.12	60.16	41.25	59.64	48.16	64.44	44.28	61.64	47.95	65.36	43.63	61.84	**51.84**	**67.71**
	Euclidean Distance																
	exact	39.74	57.26	40.39	58.94	41.25	58.71	40.39	58.94	40.39	57.80	44.28	61.31	42.55	59.68	45.36	61.79
	what_is	40.82	58.82	44.06	61.39	39.96	58.32	44.06	61.39	43.84	60.24	45.79	64.12	43.63	61.45	47.73	64.26
	describe	42.33	59.66	40.60	57.78	42.12	59.57	44.71	62.16	43.41	60.36	46.00	64.30	46.22	63.20	47.08	64.32
	meaning_of	44.49	60.94	40.60	58.56	41.47	59.57	46.44	63.14	44.92	61.88	48.81	66.11	43.20	61.35	49.46	65.98
	Manhattan Distance																
	exact	38.88	56.84	40.39	58.95	42.33	59.11	40.39	58.95	40.17	57.47	45.36	61.89	43.20	59.87	45.36	61.95
	what_is	40.60	58.51	44.06	61.87	39.74	58.31	44.06	61.87	42.98	59.92	44.71	63.28	44.06	61.24	47.73	64.56
	describe	42.55	60.03	40.17	57.58	42.12	59.63	45.14	62.17	42.98	60.04	47.95	65.14	45.14	62.48	47.73	64.66
	meaning_of	45.14	61.37	40.17	58.30	41.04	59.02	46.65	63.33	43.63	61.02	49.46	66.34	43.20	61.61	49.89	66.24
stsb-distilroberta-base	**Cosine Similarity**																
	exact	39.96	58.26	41.04	58.75	41.25	59.09	41.04	59.11	42.55	59.58	47.52	63.67	42.12	60.02	44.28	61.28
	what_is	41.68	58.89	44.28	61.35	39.74	58.61	44.28	61.87	42.98	60.74	49.24	66.65	42.55	60.47	48.81	65.16
	describe	41.25	59.51	42.98	60.87	40.82	59.80	46.00	62.66	43.41	60.91	49.68	66.13	44.28	61.74	49.24	65.37
	meaning_of	42.33	60.62	43.63	61.56	42.55	60.41	47.08	64.00	44.06	61.07	**50.54**	**67.20**	43.84	61.52	47.73	65.01
	Euclidean Distance																
	exact	35.42	54.31	36.93	55.51	39.31	57.10	38.44	57.27	39.31	56.63	46.44	62.60	41.68	58.45	41.90	58.92
	what_is	36.50	55.99	39.96	58.01	39.96	57.63	42.55	60.27	41.04	58.92	46.87	64.50	41.68	59.55	44.92	61.68
	describe	37.80	56.48	41.90	58.81	40.39	57.75	43.63	60.84	42.76	59.41	48.16	65.08	42.33	60.31	46.65	62.98
	meaning_of	38.23	56.99	39.96	58.48	39.74	57.38	42.33	60.51	40.17	58.76	50.32	66.23	43.20	60.47	45.36	62.77
	Manhattan Distance																
	exact	36.50	54.82	38.88	56.45	39.52	57.27	36.93	56.61	38.23	56.15	46.00	62.39	41.47	58.30	41.68	58.72
	what_is	37.58	56.76	38.88	57.49	40.39	57.71	42.33	59.96	40.17	58.15	46.65	64.31	41.68	59.28	44.49	61.41
	describe	36.93	56.10	42.55	59.51	39.96	57.50	43.41	60.55	41.47	58.89	47.73	64.76	41.90	60.25	47.73	63.33
	meaning_of	37.80	56.97	41.68	59.40	40.39	57.80	43.41	60.67	39.74	58.31	49.89	66.33	43.84	60.83	46.00	63.03

stsb-mpnet-base	**Cosine Similarity**																
	exact	36.93	56.41	40.82	58.69	41.04	59.18	44.49	61.25	40.17	58.54	47.30	64.77	42.76	61.00	47.08	63.53
	what_is	38.66	58.15	42.98	60.61	41.47	59.89	46.00	62.89	44.71	62.03	48.60	66.10	44.71	62.59	49.89	65.87
	describe	39.96	58.93	43.41	60.82	44.28	62.03	49.03	65.09	43.84	61.62	49.03	66.54	46.22	63.31	51.19	66.79
	meaning_of	40.39	59.12	45.36	62.04	42.33	61.27	49.89	65.87	44.49	62.22	50.54	**67.74**	48.60	64.82	50.32	66.89
	Euclidean Distance																
	exact	36.07	55.57	40.60	58.48	41.47	59.35	42.33	60.18	41.04	58.68	45.14	63.56	43.20	60.64	47.08	63.16
	what_is	38.66	55.57	42.98	61.03	41.04	60.16	47.30	63.51	42.98	61.27	49.24	66.62	45.79	62.95	49.24	65.53
	describe	40.17	58.93	42.33	60.28	43.41	61.43	48.81	64.90	44.71	62.23	47.30	65.41	46.65	63.85	50.76	66.68
	meaning_of	39.52	58.50	45.36	62.35	43.84	62.06	**51.62**	66.62	44.92	62.34	49.68	67.04	49.03	64.85	50.54	66.77
	Manhattan Distance																
	exact	36.50	56.07	41.04	58.82	41.25	59.10	43.63	60.66	40.60	58.75	46.87	64.34	43.41	61.26	46.00	62.31
	what_is	40.39	58.92	42.33	60.80	42.55	60.90	45.14	62.40	43.41	61.05	49.68	66.47	45.14	62.24	49.46	65.19
	describe	42.12	60.14	43.41	60.90	44.06	61.67	49.89	65.66	43.20	61.66	47.08	65.61	47.52	64.19	50.11	66.02
	meaning_of	41.25	59.56	43.84	62.06	44.49	62.28	50.76	66.00	44.06	61.86	50.11	67.09	47.52	64.08	50.76	66.71
all-MiniLM-L6	**Cosine Similarity**																
	exact	44.28	61.03	46.44	62.86	41.90	60.17	45.14	62.36	43.41	61.20	49.03	65.76	44.92	62.23	50.32	65.88
	what_is	45.57	61.96	45.79	62.92	46.22	63.38	46.22	63.48	43.63	62.16	50.32	67.57	47.08	64.20	52.05	67.50
	describe	44.71	62.02	44.92	62.84	44.06	62.20	49.68	65.73	42.98	61.40	50.32	66.91	45.14	63.01	49.68	66.77
	meaning_of	44.71	61.59	45.79	63.59	44.71	62.38	46.00	63.48	43.63	61.66	49.03	67.05	46.22	63.45	50.97	67.47
	Euclidean Distance																
	exact	43.84	60.78	46.00	62.65	41.25	59.85	44.92	62.23	43.84	61.40	49.03	65.76	44.92	62.23	50.32	65.88
	what_is	45.14	61.71	45.57	62.77	45.79	63.13	46.00	63.39	44.06	62.34	50.32	67.57	47.08	64.20	**52.05**	67.50
	describe	44.06	61.70	44.71	62.70	43.41	61.88	49.46	65.60	43.20	61.44	50.32	66.91	45.14	63.01	49.68	66.77
	meaning_of	44.28	61.33	45.36	63.37	44.28	62.12	45.79	63.35	43.84	61.73	49.03	67.05	46.22	63.45	50.97	67.47
	Manhattan Distance																
	exact	42.55	59.89	45.36	62.33	41.04	59.53	45.14	61.81	42.98	60.78	49.24	65.45	43.41	61.58	49.24	64.92
	what_is	44.49	61.49	44.71	62.44	45.79	62.51	46.22	63.57	42.55	61.10	48.60	66.17	47.52	64.16	50.76	66.74
	describe	43.41	61.08	44.49	62.29	41.04	59.91	49.03	65.22	43.63	61.43	50.11	66.96	42.55	61.56	49.24	66.29
	meaning_of	42.12	60.09	45.79	63.28	44.92	62.26	45.57	63.12	43.84	61.46	49.03	66.81	45.79	63.37	51.84	**67.58**
all-MiniLM-L12	**Cosine Similarity**																
	exact	40.82	59.01	46.00	62.45	42.12	60.31	44.71	62.09	41.47	60.39	49.68	65.70	43.63	61.19	48.81	64.84
	what_is	40.17	59.46	44.06	62.41	42.98	61.08	45.14	62.87	42.98	61.81	49.68	67.27	45.57	62.75	49.68	66.07
	describe	40.60	59.62	44.49	62.44	42.76	61.17	47.52	64.43	41.68	61.36	47.08	65.43	46.00	63.08	50.32	67.29
	meaning_of	40.39	59.46	43.41	61.94	43.63	61.63	48.16	64.40	42.55	61.74	49.89	67.38	46.00	63.56	50.76	66.99
	Euclidean Distance																
	exact	40.17	58.68	46.00	62.13	41.47	59.99	44.49	61.95	41.47	60.39	49.68	65.70	43.41	61.09	48.81	64.84
	what_is	39.74	59.21	43.63	62.20	42.55	60.83	44.92	62.76	43.20	61.85	49.68	67.27	45.57	62.75	49.68	66.07
	describe	39.96	59.29	44.06	62.22	42.12	60.85	47.08	64.22	41.90	61.36	47.08	65.43	45.79	62.97	50.32	67.29
	meaning_of	39.96	59.21	42.98	61.72	42.98	61.31	47.95	64.27	42.55	61.70	49.89	67.38	45.79	63.45	50.54	66.88
	Manhattan Distance																
	exact	39.52	57.99	46.22	62.19	40.60	59.30	42.98	60.83	41.04	60.16	48.81	65.19	42.55	60.67	48.60	64.91
	what_is	40.39	59.02	43.63	61.88	41.68	60.14	46.00	63.36	42.76	61.36	48.81	66.42	44.71	61.97	49.46	66.10
	describe	40.60	59.33	44.28	61.94	41.90	60.48	47.73	64.36	42.12	61.47	48.38	66.11	44.49	62.40	**51.40**	**67.73**
	meaning_of	40.60	59.37	43.41	62.00	43.20	61.37	47.52	64.20	42.76	61.46	50.54	67.42	46.44	63.33	49.46	66.37
all-mpnet-base	**Cosine Similarity**																
	exact	42.76	60.99	44.49	62.64	44.49	62.50	46.00	63.23	42.98	61.69	49.24	66.16	46.00	63.37	51.40	66.81
	what_is	43.20	61.85	45.36	64.19	47.08	63.94	49.89	66.25	43.41	62.81	52.05	68.78	47.08	64.63	50.76	67.57
	describe	42.98	61.67	45.36	63.81	44.06	61.76	50.76	66.97	44.28	63.20	52.27	68.00	46.00	63.67	50.97	67.65
	meaning_of	44.92	62.71	46.87	64.70	47.30	64.20	48.81	65.95	44.06	63.15	52.70	69.14	47.52	64.98	51.19	68.22
	Euclidean Distance																
	exact	42.12	60.66	44.06	62.43	43.84	62.18	45.79	63.09	43.41	61.85	49.24	66.16	46.00	63.37	51.40	66.81
	what_is	42.76	61.60	44.92	63.98	46.65	63.69	49.68	66.13	43.84	62.99	52.05	68.78	47.08	64.67	50.76	67.57
	describe	42.33	61.35	44.92	63.59	43.41	61.43	50.54	66.83	44.49	63.34	52.27	68.00	46.00	63.67	50.97	67.65

	meaning_of	44.49	62.46	46.44	64.48	46.87	63.95	48.60	65.83	44.49	63.36	52.70	69.14	47.52	65.01	51.19	68.22
	Manhattan Distance																
	exact	42.55	60.63	44.71	62.80	42.98	61.06	46.22	63.00	42.98	61.72	50.76	66.84	47.08	63.90	50.32	66.39
	what_is	42.55	61.56	46.22	64.69	45.79	63.20	49.24	66.13	43.41	62.80	**54.43**	**70.45**	46.65	64.33	50.97	67.57
	describe	42.76	61.40	46.00	64.07	43.63	61.87	48.60	65.58	44.06	62.54	52.70	68.71	45.36	63.52	52.27	68.36
	meaning_of	43.84	62.12	49.03	66.06	45.57	62.82	48.81	65.87	44.92	63.55	54.43	70.37	48.38	65.30	50.54	67.68
multi-qa-distilbert-cos	**Cosine Similarity**																
	exact	41.47	59.77	44.49	61.92	43.20	60.49	42.98	61.01	42.98	61.39	48.38	65.41	43.63	61.76	49.68	65.92
	what_is	42.55	61.32	44.06	62.68	49.89	64.65	47.52	64.91	44.49	63.14	50.32	67.75	47.52	64.78	49.89	67.11
	describe	42.12	60.85	44.71	62.51	47.52	63.49	46.44	64.01	43.63	62.47	48.81	66.44	46.00	63.82	50.11	66.91
	meaning_of	42.12	61.17	44.49	62.79	46.87	63.21	46.44	64.48	43.84	62.90	49.89	67.70	46.87	64.41	**51.40**	**68.09**
	Euclidean Distance																
	exact	41.04	59.52	44.06	61.71	42.56	60.17	42.76	60.91	43.41	61.57	48.38	65.41	43.41	61.65	49.68	65.92
	what_is	42.12	61.07	43.84	62.54	49.46	64.40	47.30	64.82	44.92	63.32	50.32	67.75	47.30	64.67	49.89	67.11
	describe	41.47	60.53	44.28	62.29	46.87	63.17	46.22	63.88	43.84	62.54	48.81	66.44	45.79	63.68	50.11	66.91
	meaning_of	41.90	60.99	44.06	62.57	46.44	62.96	46.22	64.36	44.28	63.08	49.89	67.70	46.65	64.31	51.40	68.06
	Manhattan Distance																
	exact	40.60	59.86	43.41	60.81	42.33	60.13	42.55	60.73	42.98	61.53	49.03	65.71	42.98	61.51	49.24	65.35
	what_is	41.47	60.90	44.06	62.50	47.52	63.33	46.22	64.01	44.71	63.42	50.11	67.47	47.73	64.83	49.46	66.49
	describe	41.90	60.71	42.76	61.00	47.73	63.65	46.65	63.76	42.55	61.81	49.24	66.61	46.87	64.17	50.11	66.96
	meaning_of	42.55	61.11	44.71	62.70	45.79	62.61	46.65	64.59	45.57	63.70	50.97	68.08	45.57	63.66	50.76	67.46
multi-qa-MinLM-L6-cos	**Cosine Similarity**																
	exact	39.96	58.60	42.76	59.65	40.17	58.66	42.55	60.19	41.68	60.27	47.73	64.53	40.17	59.23	46.00	63.19
	what_is	40.82	59.76	42.55	60.37	41.90	60.63	44.71	61.93	43.84	62.01	49.68	**66.91**	39.52	59.68	48.81	65.37
	describe	39.31	58.99	42.33	59.99	40.82	59.36	46.00	62.63	42.12	61.42	48.38	65.87	39.96	59.33	48.38	65.46
	meaning_of	41.25	60.06	42.98	60.83	41.68	60.14	46.00	62.95	41.90	60.99	49.68	66.74	42.12	61.20	**49.89**	66.72
	Euclidean Distance																
	exact	39.52	58.35	42.33	59.43	39.52	58.34	42.33	60.06	42.12	60.47	47.73	64.53	40.17	59.23	46.00	63.19
	what_is	40.60	59.59	42.12	60.16	41.47	60.38	44.49	61.83	44.28	62.27	49.68	66.91	39.74	59.78	48.81	65.37
	describe	38.88	58.76	41.90	59.78	40.17	59.04	45.79	62.50	42.33	61.49	48.38	65.87	39.86	59.33	48.38	65.46
	meaning_of	40.82	59.83	42.55	60.62	41.25	59.88	45.79	62.83	42.12	61.13	49.68	66.74	42.12	61.20	49.89	66.72
	Manhattan Distance																
	exact	39.09	57.37	40.60	58.07	40.39	58.64	41.25	59.06	42.55	60.68	47.52	64.10	39.96	58.96	44.06	61.66
	what_is	40.82	59.40	42.76	60.14	40.82	59.49	43.84	61.65	44.28	62.08	49.24	66.57	41.90	61.13	46.87	64.42
	describe	38.23	57.35	42.33	59.58	39.74	58.84	45.14	62.05	41.68	60.51	48.38	65.32	40.82	59.65	47.30	64.50
	meaning_of	40.39	59.04	42.33	60.46	41.68	60.05	45.57	62.92	42.33	61.35	48.60	65.93	42.33	61.26	49.89	66.55
sentence-t5-base	**Cosine Similarity**																
	exact	40.17	59.23	42.98	60.81	41.04	59.65	43.20	61.06	44.28	62.51	52.48	68.68	47.08	64.16		
	what_is	41.68	60.09	45.79	62.73	42.12	60.04	45.57	62.96	44.28	62.20	52.05	67.85	45.79	62.79		
	describe	42.98	61.24	48.38	64.40	43.84	61.89	48.81	65.27	44.71	62.73	52.48	68.42	45.57	62.96		
	meaning_of	42.98	61.34	45.36	62.78	42.55	60.54	48.60	65.31	44.28	62.51	52.48	68.68	47.08	64.16		
	Euclidean Distance																
	exact	39.52	58.91	42.55	60.59	40.39	59.32	42.98	60.92	44.28	61.63	47.30	64.74	43.20	60.51		
	what_is	41.04	59.76	45.36	62.51	41.47	59.72	45.36	62.84	44.49	62.24	52.05	67.85	45.57	62.68		
	describe	42.33	60.91	47.95	64.19	43.41	61.64	48.60	65.13	44.92	62.85	52.48	68.42	45.57	62.93		
	meaning_of	42.33	61.01	44.92	62.57	41.90	60.21	48.38	65.18	44.49	62.61	52.48	68.68	46.87	64.05		
	Manhattan Distance																
	exact	39.74	59.01	41.68	60.16	39.52	58.88	42.76	60.50	42.98	60.79	47.30	64.34	41.90	59.86		
	what_is	41.25	59.57	45.57	62.66	41.47	59.73	45.36	63.04	44.49	62.61	52.92	68.35	44.71	62.50		
	describe	42.12	60.87	47.52	63.63	42.98	61.17	49.03	65.37	44.06	62.14	52.92	68.66	46.00	63.14		
	meaning_of	44.28	62.02	45.36	63.02	42.76	60.73	47.52	64.92	44.49	62.61	**53.35**	**69.15**	45.79	63.18		

sentence-t5-large	**Cosine Similarity**														
	exact	41.04	59.18	43.63	61.17	41.04	59.02	46.22	62.66	43.84	61.29	48.60	65.45	43.84	61.15
	what_is	41.68	60.46	47.95	64.09	42.33	59.89	48.60	64.93	45.79	63.27	53.56	69.47	47.95	64.39
	describe	43.63	61.78	47.52	64.41	40.60	60.01	52.27	67.40	45.57	63.22	52.92	68.61	48.38	65.26
	meaning_of	43.41	61.77	49.24	65.54	43.84	61.10	53.13	68.11	49.24	65.16	**54.00**	**69.75**	48.16	64.82
	Euclidean Distance														
	exact	40.39	58.85	43.20	60.95	40.39	58.69	46.00	62.52	43.84	61.26	48.38	65.34	43.63	61.05
	what_is	41.25	60.21	47.52	63.88	41.68	59.57	48.38	64.80	45.79	63.27	53.56	69.47	47.95	64.39
	describe	42.98	61.46	47.08	64.19	40.17	59.75	52.05	67.27	45.79	63.33	52.92	68.61	48.38	65.26
	meaning_of	42.76	61.45	48.81	65.32	43.20	60.77	52.92	67.97	49.46	65.27	54.00	69.75	48.16	64.82
	Manhattan Distance														
	exact	40.60	58.94	42.76	60.61	40.17	58.41	46.65	63.08	43.84	61.44	48.81	65.22	44.06	61.32
	what_is	41.47	60.24	47.95	63.77	40.60	59.06	47.95	64.69	46.65	63.26	53.56	69.46	48.60	64.63
	describe	42.33	60.99	47.30	64.14	40.82	59.82	52.27	67.49	46.22	63.25	51.84	68.13	47.08	64.22
	meaning_of	44.71	62.40	48.30	64.94	43.20	60.32	51.84	67.44	49.46	65.12	52.92	69.43	46.87	64.17
gtr-t5-base	**Cosine Similarity**														
	exact	43.63	61.04	44.28	62.07	43.20	60.76	47.73	63.86	45.57	63.09	48.81	66.32	45.14	62.63
	what_is	44.28	62.13	46.87	63.94	44.92	62.30	48.38	64.25	46.65	63.84	50.54	68.02	46.87	64.20
	describe	44.71	62.29	46.65	63.74	46.00	62.76	50.32	66.19	45.36	63.33	**53.56**	**69.66**	48.16	64.68
	meaning_of	42.76	61.31	48.38	64.82	44.92	62.30	50.11	65.58	46.65	63.84	50.54	68.02	46.87	64.20
	Euclidean Distance														
	exact	42.98	60.72	43.84	61.85	42.55	60.44	47.52	63.72	45.79	63.16	48.81	66.32	45.14	62.63
	what_is	43.84	61.88	46.44	63.73	44.28	61.98	47.95	64.04	46.87	63.89	50.32	67.91	46.65	64.09
	describe	44.06	61.96	46.22	63.52	45.36	62.43	49.89	65.97	45.36	63.25	53.56	69.66	47.95	64.57
	meaning_of	42.12	60.98	47.95	64.60	43.84	61.62	49.68	65.37	45.79	63.88	50.11	67.84	47.30	64.17
	Manhattan Distance														
	exact	44.71	61.47	44.28	62.11	42.33	60.17	47.08	63.21	45.57	62.99	49.03	66.35	46.22	63.08
	what_is	43.41	61.73	45.14	62.86	44.49	62.20	47.30	63.85	47.52	64.23	49.46	67.20	47.30	64.22
	describe	44.28	61.99	46.65	63.36	45.14	62.44	49.24	65.32	47.52	64.42	52.70	68.95	46.65	63.60
	meaning_of	44.06	62.08	45.57	62.88	41.90	60.55	47.52	64.25	46.65	64.27	51.84	68.64	47.08	64.17
gtr-t5-large	**Cosine Similarity**														
	exact	42.98	61.08	45.79	62.72	44.92	61.78	46.22	63.33	45.57	62.96	52.70	68.38	48.16	64.54
	what_is	46.00	63.81	46.22	64.36	47.30	64.13	48.81	65.47	46.87	64.70	54.21	70.46	50.54	66.86
	describe	46.44	64.25	47.08	64.53	46.87	63.76	52.05	67.74	47.08	64.85	54.64	70.56	50.11	66.05
	meaning_of	46.22	63.97	46.22	64.33	47.30	63.76	49.03	65.73	49.89	66.59	55.08	**71.06**	49.68	66.58
	Euclidean Distance														
	exact	42.55	60.83	45.36	62.51	44.28	61.45	46.00	63.19	45.79	63.00	52.70	68.38	47.95	64.44
	what_is	45.36	63.48	45.79	64.15	46.65	63.81	48.60	65.33	47.08	64.77	54.00	70.35	50.54	66.86
	describe	45.79	63.92	46.65	64.31	46.22	63.44	51.84	67.60	47.30	64.92	54.43	70.45	50.11	66.05
	meaning_of	45.57	63.65	45.79	64.12	46.65	63.43	48.81	65.59	50.11	66.67	54.86	70.95	49.46	66.47
	Manhattan Distance														
	exact	41.68	60.12	44.49	61.76	41.47	59.94	47.73	64.05	44.92	62.17	52.70	68.39	48.16	64.31
	what_is	44.28	62.55	46.00	64.00	45.79	62.64	48.38	64.98	46.65	64.35	53.35	69.95	49.24	65.79
	describe	46.00	63.91	45.79	63.78	46.65	63.57	51.40	67.31	48.38	65.20	54.00	70.30	51.19	66.46
	meaning_of	44.28	62.59	45.36	63.76	44.49	61.82	49.03	65.61	49.03	65.69	**55.29**	70.96	48.38	65.58

We observe that the utilization of LLM-based enhancement provides performance boosts in text-to-text retrieval compared to the No-LLM baseline, in most cases. Nevertheless, it still stays behind the best performance have achieved by GPT-3-enhanced VL retrieval. We assume that the reason is the information loss induced when converting from the visual to the textual modality during captioning.

Concerning the VL transformers, BLIP appears to struggle to identify correlations between the phrases - whether enriched or not - and the captions, whereas the others display encouraging results. ALIGN obtains the highest performance, outperforming CLIP by 4%, and then CLIP, CLIP-L, and $\text{CLIP}_L AION$ follows. Regarding the similarity metric, the manhattan distance appears to perform better when combined with VL models.

As for the linguistic semantic similarity models, all employed models appear to produce comparable results, with highest accuracy scores ranging between 49% and 55%. The lowest accuracy corresponds to multi-qa-MinLM-L6-cos, while the highest accuracy corresponds to gtr-t5-large. With these models, all similarity metrics are sufficiently near, with cosine similarity and manhattan distance are associated with the highest scores.

Finally, regarding the captioning models, BLIP-L and ViT-L appear to generate the most appropriate and useful captions for text-to-text retrieval for VWSD, with beam search providing an additional boost.

An intriguing observation is that VL transformers, which are trained to handle both textual and visual modalities, perform better in producing *textual* embeddings, than sentence similarity embeddings, despite the fact that the latter have been explicitly tuned on semantic textual similarity. Overall, ALIGN VL transformer achieves 62.42% accuracy, outperforming the best accuracy score of sentence similarity embeddings (namely gtr-t5-large) by 7% (55.29%).

5.2.3 Wikipedia & Wikidata image retrieval

In Tab. 5.15 we present results regarding image-to-image retrieval between candidates i and web retrieved images i_w. Out of the 463 samples of the test set, Wikipedia API and Wikidata API returned results for 460 and 324 phrases respectively.

		CLIP		ALIGN	
Similarity	**Image source**	**acc.**	**MRR**	**acc.**	**MRR**
Cosine	Wikidata Images	34.26	50.13	31.11	47.84
	Wikipedia Images	53.26	68.14	53.26	**68.44**
Euclidean	Wikidata Images	33.64	49.24	30.83	47.52
	Wikipedia Images	52.17	66.95	**53.48**	68.40
Manhattan	Wikidata Images	33.02	48.75	31.11	47.66
	Wikipedia Images	52.82	67.25	53.26	68.27

Table 5.15: Image-to-image retrieval results

The Wikidata API retrieval acquired images for 324 test samples, as previously mentioned. In contrast, the Wikipedia API assured that corresponding images existed for the entire test set, so that the entire set could be included in the image-to-image retrieval. The best results achieved using the ALIGN VL transformer. However, we observe that even best results for image-to-image retrieval are not competent against our previous approaches; we infer that exclusively visual representations are not expressive enough to distinguish fine-grained details between semantically related candidates.

5.2.4 Learn to Rank

In Tab. 5.16 and Tab. 5.17 we showcase the results using CLIP and ALIGN as VL retriever respectively. Each column of these tables corresponds to each one of the methods combined in order to produce the final feature vector. The presented feature combinations, are constituted by the following:

1. *Baseline features*: the option to incorporate (or not) the penalty $p(i)$ in $score(t, i)$ for VL retrieval
2. *LLM-enhancement features*: the prompt to generate enhanced phrases t_e (or an ensemble of prompts leading to multiple t_e) and the option for incorporation (or not) of $p(i)$ in $score(t_e, i)$
3. *Text retrieval features*: the captioner to generate caption c_i, along with the text embedding model and the similarity metric (cosine/euclidean/manhattan) for text-to-text retrieval, as well as the phrase (original t, or enhanced t_e, or an ensemble of enhanced phrases t_e derived from the usage of different prompts)
4. *Image retrieval features*: image embedding model and similarity metric (cosine/euclidean/manhattan) for image-to-image retrieval.

Baseline	LLM-enhance		Text retrieval features				Image retrieval feat.		Metrics	
$p(i)$	**Prompt**	$p(i)$	**Captioner**	**Embedding**	**Similarity**	**Phrase**	**Embedding**	**Similarity**	**Acc.**	**MRR**
-	-	-	-	-	-	-	-	-	63.93	76.33
✓	-	-	-	-	-	-	-	-	68.90	80.04
✓	-	-	-	-	-	-	-	-	62.85	75.88
✓	-	-	-	-	-	-	CLIP	cosine	70.87	81.36
✓	-	-	-	-	-	-	CLIP	euclidean	70.22	81.09
✓	-	-	-	-	-	-	CLIP	manhattan	69.78	80.95
✓	-	-	GiT-L-greedy	CLIP	cosine	t	-	-	62.85	76.08
✓	-	-	GiT-L-beam	CLIP	cosine	t	-	-	63.07	76.14
✓	-	-	GiT-L-beam	CLIP	euclidean	t	-	-	62.85	75.85
✓	-	-	GiT-L-beam	CLIP	manhattan	t	-	-	62.85	76.11
✓	-	-	Blip-L-greedy	CLIP	cosine	t	-	-	61.77	75.48
✓	-	-	Blip-L-beam	CLIP	cosine	t	-	-	62.85	75.94
✓	all	✓	-	-	-	-	-	-	70.37	81.65
✓	meaning_of	-	-	-	-	-	-	-	65.85	78.67
✓	meaning_of	✓	-	-	-	-	-	-	66.52	79.21
✓	exact	✓	-	-					65.57	78.25
✓	what_is	✓	-	-	-	-	-	-	67.45	79.55
✓	describe	✓	-	-	-	-	-	-	70.14	80.75
✓	all	✓	-	-	-	-	CLIP	cosine	72.05	82.81
✓	all	✓	Blip-L-beam	CLIP	cosine	t	CLIP	cosine	72.05	82.61
✓	all	✓	GiT-L greedy	CLIP	cosine	t	CLIP	cosine	70.81	82.28
✓	all	✓	GiT-L-greedy	CLIP	cosine	all t_e+t	CLIP	cosine	73.91	83.53
LTR of [11] (best results)									77.97	85.88
SemEval organizers' baseline									60.48	73.87

Table 5.16: LTR results using feature combinations as extracted from our previous 4 approaches (baseline, LLM enhancement, text retrieval, image retrieval). CLIP is employed as the VL retriever.

For all experiments of Tab 5.16 and Tab. 5.17 we utilized the following hyperparameters configuration: n_estimators: 500, early_stopping: 100, learning_rate: 0.03, feature_fraction: 0.25, max_bin: 100, min_child_samples: 50 and reg_alpha: 0.05. Also, an 80-20 train/validation split was followed, allocating 2514 samples in the validation set.

In general, the incorporation of LLM-based phrase enhancement in LTR yields significant advantages, providing optimal metric results in comparison to the alternative feature combinations, or our alternative approaches outlined in Tab. 5.5, 5.6, 5.13, 5.14. The best results are obtained by employing ALIGN as the VL retriever and incorporating all features, as indicated by the colored instances in Table 5.17. This is an intriguing observation since standalone text retrieval (Tab. 5.13, 5.14) and image retrieval (Tab. 5.15) experiments did not produce competitive results. However, it is worth noting that incorporating the corresponding features in the training of LTR model improves performance. Furthermore, the integration of features through ensembling is also tremendously advantageous. This approach is applicable to both combining the LLM-enhanced prompt features, such as incorporating features from $t_{exact}, t_{what-is}, t_{describe}, t_{meaning-of}$, denoted as *all prompts*, and combining phrase features for text-to-text retrieval. In this context, *all* $t_e + t$ refers to the combination of features from all four aforementioned enhancements, along with the original given phrase t. As illustrated in Table 5.17, the majority of ensemble feature combinations exhibit superior performance compared to baselines and alternative implementations [11].

Baseline	LLM-enhance		Text retrieval features				Image retrieval feat.		Metrics	
$p(i)$	Prompt	$p(i)$	Captioner	Embedding	Similarity	Phrase	Embedding	Similarity	Acc.	MRR
-	-	-	-	-	-	-	-	-	63.93	76.33
✓	-	-	-	-	-	-	-	-	68.90	80.04
✓	-	-	-	-	-	-	ALIGN	cosine	71.96	82.46
✓	-	-	-	-	-	-	ALIGN	euclidean	72.17	82.34
✓	-	-	-	-	-	-	ALIGN	manhattan	72.61	82.84
✓	-	-	BLIP-greedy	ALIGN	cosine	t	-	-	68.47	79.88
✓	-	-	GiT-greedy	ALIGN	cosine	t	-	-	68.25	79.53
✓	-	-	BLIP-L-greedy	ALIGN	cosine	t	-	-	68.47	79.65
✓	-	-	GiT-L-greedy	ALIGN	cosine	t	-	-	68.47	79.68
✓	-	-	GiT-L-greedy	ALIGN	euclidean	t	-	-	68.68	79.69
✓	-	-	GiT-L-greedy	ALIGN	manhattan	t	-	-	68.47	79.69
✓	-	-	BLIP-beam	ALIGN	cosine	t	-	-	68.90	79.85
✓	-	-	GiT-beam	ALIGN	cosine	t	-	-	68.47	79.76
✓	-	-	BLIP-L-beam	ALIGN	cosine	t	-	-	68.47	79.60
✓	-	-	BLIP-L-beam	ALIGN	euclidean	t	-	-	68.03	79.62
✓	-	-	BLIP-L-beam	ALIGN	manhattan	t	-	-	68.47	79.98
✓	-	-	GiT-L-beam	ALIGN	cosine	t	-	-	67.60	79.42
✓	meaning_of	-	-	-	-	-	-	-	73.22	82.79
✓	meaning_of	✓	-	-	-	-	-	-	75.16	84.13
✓	exact	✓	-	-	-	-	-	-	70.41	81.10
✓	what_is	✓	-	-	-	-	-	-	71.71	81.52
✓	describe	✓	-	-	-	-	-	-	73.00	82.84
✓	all prompts	✓	-	-	-	-	-	-	73.87	83.96
✓	all-except exact	✓	-	-	-	-	-	-	74.30	83.80
✓	meaning_of + describe	✓	-	-	-	-	-	-	74.30	83.86
✓	all-except exact	✓	-	-	-	-	ALIGN	manhattan	76.09	85.36
✓	all-except exact	✓	-	-	-	-	ALIGN	cosine	76.52	85.29
✓	all prompts	✓	-	-	-	-	ALIGN	cosine	76.52	85.70
✓	all prompts	✓	BLIP-L-beam	ALIGN	cosine	t	ALIGN	cosine	77.61	**85.90**
✓	all prompts	✓	BLIP-L-beam	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	77.17	**86.08**
✓	all prompts	✓	BLIP-L-beam	ALIGN	cosine	$t_{meaning\_of}$	ALIGN	cosine	76.52	85.63
✓	all prompts	✓	BLIP-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	**78.48**	**86.65**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	t	ALIGN	cosine	77.83	**86.30**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	$t_{meaning\_of}$	ALIGN	cosine	77.39	**85.92**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	**79.35**	**87.23**
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	euclidean	76.96	85.85
✓	all prompts	✓	GiT-L-greedy	ALIGN	cosine	all $t_e + t$	ALIGN	manhattan	76.96	**86.00**
✓	all prompts	✓	GiT-L-beam	ALIGN	cosine	all $t_e + t$	ALIGN	cosine	76.96	**85.92**
LTR of [11] (best results)									77.97	85.88
SemEval organizers' baseline									60.48	73.87

Table 5.17: LTR results using feature combinations as extracted from our previous 4 approaches (baseline, LLM enhancement, text retrieval, image retrieval). ALIGN is employed as the VL retriever. **Colored** instances denote best results overall, while **bold** instances highlight instances that outperform best results of [11].

5.2.5 Question Answering for VWSD and CoT prompting

In Tab. 5.18 we present accuracy scores occuring from transforming VWSD to QA using zero-shot (with and without CoT) and few-shot (without CoT) prompting.

Captioner	**Zero-shot**				**Few-shot (random)**	**Few-shot (top)**	**Few-shot (inv.top)**
	no_CoT	CoT	choose no_CoT	choose CoT	no_CoT	no_CoT	no_CoT
				GPT-3.5-turbo			
GiT-L (greedy)	44.49	47.30	51.84	52.27	51.19	51.40	53.56
GiT-L (beam)	40.82	36.50	50.54	49.68	46.12	47.83	45.61
BLIP-L (greedy)	47.95	43.84	49.46	44.06	48.16	48.81	50.32
BLIP-L (beam)	38.01	34.13	50.97	50.97	40.91	40.49	40.49
ViT-GPT2 (greedy)	28.94	25.05	32.40	29.81	31.32	31.45	28.91
ViT-GPT2 (beam)	30.24	25.92	32.83	33.05	32.03	28.73	23.64
				Vicuna-13B			
GiT-L (greedy)	34.34	27.65	20.52	20.52	31.89	33.63	36.30
GiT-L (beam)	11.02	7.91	19.44	11.23	< 2	< 2	< 2
BLIP-L (greedy)	30.02	23.76	20.95	21.81	35.56	36.08	36.48
BLIP-L (beam)	9.41	6.27	12.74	8.64	< 2	< 2	< 2
ViT-GPT2 (greedy)	21.60	21.17	17.49	15.33	24.83	24.94	26.11
ViT-GPT2 (beam)	11.45	6.91	16.85	12.74	2.81	3.89	4.75

Table 5.18: Accuracy scores for VWSD as a QA problem with and without CoT prompting.

In the context of the VWSD as QA scenario, there is a noticeable difference in performance between GPT-3.5-turbo and Vicuna-13B. This indicates that **model scale does matter**, i.e. the size of the model plays a significant role, contrary to previous cases involving VL retrieval with LLM-based enhancement. Smaller models lack the requisite knowledge and reasoning capabilities to accurately deduce the correct answer from captions in the QA setting, regardless of the prompt template used or the choice between zero-shot and few-shot strategies. This finding is consistent with the observation made by [30] that LLM reasoning capabilities become apparent when operating at a larger *scale*. Furthermore, it remains unclear which approach, beam or greedy captioning, is more effective in eliciting the required knowledge. Specifically, when considering GiT-L and BLIP-L, there is a distinct inclination towards employing greedy decoding alongside CoT and no_CoT prompting. Particularly, in the case of Vicuna-13B, this preference exhibits a notable distinction, as it showcases substantial decreases in performance when beam decoding is utilised instead of greedy decoding. However, the opposite holds for ViT-GPT2. The performance deteriorates more when beam decoding is utilised in conjunction with a few-shot method. Simultaneously, the "choose" prompts do not exhibit a discernible pattern in relation to the decoding strategy. Overall, GiT-L (greedy) exhibits the most promising captioning capabilities in the majority of Tab. 5.18 results, while BLIP-L (greedy) appears more capable in the few-shot prompting setting of Vicuna-13B.

Zero-shot prompting and CoT reasoning. The accuracy of zero-shot QA prompting is not always encouraging, particularly when compared to the enrichment results of Tables 5.5, 5.6. One possible fundamental reason is the conversion of images to text via captioning; this intra-modality conversion may result in errors and information loss that negatively affects the final performance. Obviously, when CoT is utilised, lower efficacy will also impact the quality of the produced explanations.

We explore this scenario by presenting an example in which the use of CoT results provides an incorrect answer, whereas the no_CoT case is successful. In Fig. 5.2.4 the candidates for the phrase "tender embrace" are showcased. By employing GiT-L (greedy) as the image captioner, we form the question Q as follows:

Q: What is the most appropriate caption for the tender embrace? Answer Choices: (A) a small boat sitting on top of a dock. (B) a group of people walking on a green hill. (C) a student gets a hug from a student. (D) a large fly laying on a rock in the water. (E) the bus stop at the station (F) a train is parked at a station. (G) a crowd of people watching a concert. (H) a train station with a sign on the side of it. (I) a black and red train on a track. (J) a man laying in the sand on top of a surfboard.

A | B | C (Gold image) | D | E

F | G | H | I | J

Figure 5.2.4: Candidate images for the phrase "tender embrace".

Table 5.19 displays the results of GPT-3.5-turbo with and without *zero-shot* CoT. After observing the captions c_i generated by GiT-L, we validate that they accurately describe the context and are therefore acceptable answer choices. However, when prompted with CoT in a zero-shot manner, GPT-3.5-turbo cannot define what "tender embrace" alludes to without additional information. In contrast, it successfully returns the correct response when no_CoT is provided. This discrepancy indicates that although the phrase "tender embrace" is stored in the LLM, it cannot be activated by zero-shot CoT prompting. This can be regarded as an inherent problem of CoT prompting, at least in the zero-shot setting, since the only variable that distinguishes successful reasoning from unsuccessful reasoning is the prompt itself.

no_CoT

Q: What is the most appropriate [...]
A: The most appropriate caption for the tender embrace would be: (C) a student gets a hug from a student. ✓

CoT

Q: What is the most appropriate [...]
A: Let's think step by step.
First, we need to understand what the tender embrace is referring to. Once we know that, we can look at the answer choices and find the one that best matches the subject of the photo. Therefore, without further information about the photo, it is not possible to choose the most appropriate caption. Could you please provide more context or information about the photo?
Therefore, among A through J, the answer is not applicable without more information about the photo"
X

Table 5.19: Example of VWSD as QA for candidates corresponding to "tender embrace" given phrase.

Conversely, there are cases, where the zero-shot CoT prompting can provide valuable insights, such as the one presented in Fig. 5.2.5 with candidates corresponding to the phrase "metal steel".

By again using GiT-L as the captioner, the question Q is formed as:

Q: What is the most appropriate caption for the metal steel? Answer Choices: (A) a chocolate bar with three sides (B) [unused0] and [unused0] at the concert in 2007 (C) a guitar and a guitar are displayed in front of a speaker. (D) frosty patterns on a window (E) gold in the rocks - - (F) a black piece of metal with a large black square in the middle. (G) a jar of honey on a wooden table. (H) a close up of a metal plate with a pattern of lines. (I) a large white quartz rock with a clear base. (J) gold jewelry from the late 19th century.

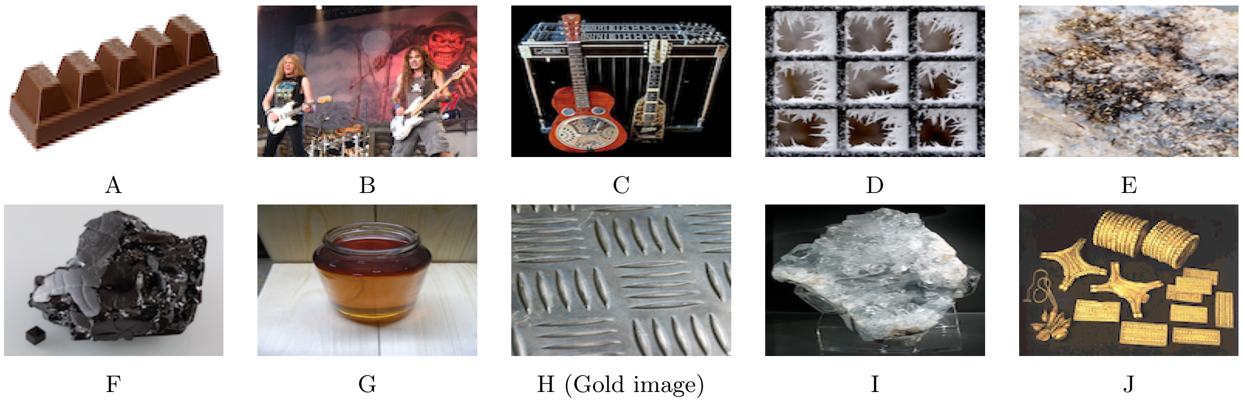

Figure 5.2.5: Candidate images for the phrase "metal steel".

Then, the GPT-3.5-turbo produces the results shown in Table 5.20. In this instance, CoT initiates the correct reasoning path for selecting the correct caption option (H) "a close-up of a metal plate with a pattern of lines" for the corresponding image (Fig. 5.2.5), while the no_CoT prompting case is readily misled by the semantically similar caption (F) containing the concept "metal" ("a black piece of metal with a large black square in the middle"). Simultaneously, CoT prompting elicits the appropriate behaviour of GPT-3.5-turbo, instructing it to differentiate between the semantically similar phrases "metal steel" and "black piece of metal." This fine-grained capability is a crucial requirement for VWSD systems, and it appears to be related to CoT reasoning.

no_CoT
Q: What is the most appropriate [...] A: The most appropriate caption for the metal steel would be (F) a black piece of metal with a large black square in the middle. X
CoT
Q: What is the most appropriate [...] A: Let's think step by step. First, we need to understand what metal steel is and what its characteristics are. Steel is a hard and strong metal alloy made mainly of iron and carbon. It is often used in construction, machinery, and transportation. Based on this information, the most appropriate caption for metal steel would be (H) a close up of a metal plate with a pattern of lines. This caption describes the texture and appearance of steel, which is often characterized by its distinctive pattern of lines. The other options do not accurately describe steel or its unique qualities. Therefore, among A through J, the answer is (H) a close up of a metal plate with a pattern of lines." ✓

Table 5.20: Example of VWSD as QA for candidates corresponding to "metal steel" given phrase.

Finally, "choose" CoT/no_CoT prompts seems to have a performance advancement against standalone CoT/no_CoT prompts, especially when GiT-L and BLIP-L are used as captioners, as in the example shown in Fig. 5.2.6, with candidates corresponding to the phrase "trotting appendix".

In Tab. 5.21 the results of GPT-3.5-turbo using standalone CoT/no_CoT and "choose" CoT/no_CoT prompts and and employing GiT-L (greedy) as the image captioner are showcased. We can observe that "choose" prompts succeed in contrast to standalone CoT/no_CoT prompts, which both fail to provide the right answer, underlying the importance of prompt engineering for eliciting LLM knowledge and reasoning.

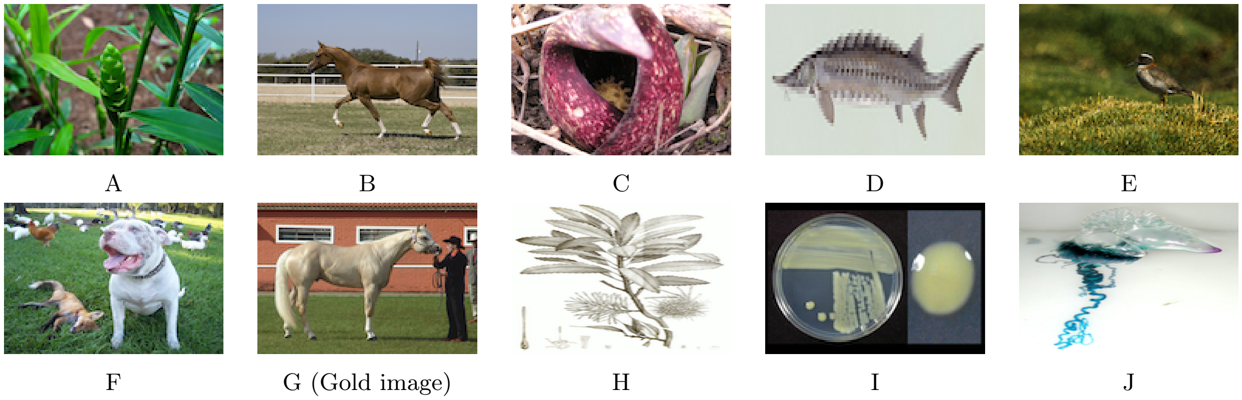

Figure 5.2.6: Candidate images for the phrase "trotting appendix".

Table 5.21: Example of VWSD as QA for candidates corresponding to "trotting appendix" given phrase.

no_CoT
Q: What is the most appropriate caption for the trotting appendix? Answer Choices: (A) a close up of a plant with a green flower (B) a horse running in a field with a fence in the background. (C) a close up of a flower inside a plant (D) a close up of a fish (E) a bird standing on top of a grass covered field. (F) a dog and a fox laying on the grass. (G) a person holding a horse in front of a brick building. (H) the flowers of the - - (I) a photo of a petri dish with a yellow substance on it. (J) a blue jellyfish with a long tail A: The most appropriate caption for the trotting appendix would be (B) - "A horse running in a field with a fence in the background." **X**
CoT
Q: What is the most appropriate [...] A: Let's think step by step. First, we need to identify what a trotting appendix is. As an intelligent assistant, I do not have enough information to answer that question. If you could provide more context or clarification, I would be happy to assist you in finding the appropriate caption. Therefore, among A through J, the answer is not applicable. **X**
choose no_CoT
You have ten images, (A) to (J), which are given you in the form of captions. (A) a close up of a plant with a green flower (B) a horse running in a field with a fence in the background. (C) a close up of a flower inside a plant (D) a close up of a fish (E) a bird standing on top of a grass covered field. (F) a dog and a fox laying on the grass. (G) a person holding a horse in front of a brick building. (H) the flowers of the - - (I) a photo of a petri dish with a yellow substance on it. (J) a blue jellyfish with a long tail You should choose the image, and therefore the caption that could better represent the 'trotting appendix'. What image do you choose? Based on the given captions, the closest image and caption that could better represent the "trotting appendix" is (G) a person holding a horse in front of a brick building.✓

choose CoT
You have ten images, (A) to (J), [...] You should choose the image, and therefore the caption that could better represent the 'trotting appendix'. Use the following format: Question: What image do you choose? Thought: you should always think about what you choose Result: the result of your thought Thought: I know the final answer Final Answer: the image that you choose Begin! Question: What image do you choose? Thought: To represent the "trotting appendix," I need to consider the element of trotting and a reference to an appendix. Result: Image (G) - a person holding a horse in front of a brick building - seems to match the idea of trotting with the horse. Thought: However, I should also consider the reference to an appendix. None of the given images have an explicit indication of an appendix. Final Answer: Therefore, based on the options given, I would choose Image (G) - a person holding a horse in front of a brick building - as the closest representation of the "trotting appendix." ✓

This analysis demonstrates that the current state of CoT reasoning cannot be relied upon completely, as the deployment of CoT prompts may lead the LLM down erroneous reasoning paths. Therefore, CoT reasoning for VWSD is still quite immature in order to either provide useful explanations or use directly as a retriever.

Moreover, when comparing standalone CoT/no_CoT prompts to "choose" CoT/no_CoT prompts, there is a clear performance improvement when "choose" prompts are utilised, reaching more than 10% performance improvement in the case of BLIP-L (beam) as captioner and GPT-3.5-turbo as the LLM to be prompted. This is evidence that more descriptive and motivating prompts are more effective at evoking the correct reasoning process of an LLM. Lastly, it is unclear whether CoT facilitates performance when combined with the "choose" prompting template, since the best results per captioner and LLM of Tab. 5.18 alternate between "choose" CoT/no_CoT strategies and are frequently comparable.

Few-shot prompting. In the baseline few-shot setting (few-shot random), we randomly select k=5 instances to serve as in-context examples. These in-context samples are comprised of k questions Q followed by their ground truth answer choice.

Table 5.18 indicates that few-shot performance using no_CoT prompts is substantially superior to their zero-shot counterparts, while being generally comparable to "choose" prompt results, despite the fact that the derived accuracy is close to random choice or even worse in most cases. Therefore, despite the advanced engineering few-shot prompting requires in comparison to the zero-shot setting, it is a more viable option for retrieving the LLMs' internal reasoning capabilities. As for the choice of in-context sample selection and order strategy, the results are inconclusive; similarity-based sample selection (top & inv. top columns of Table 5.18) may produce better (GIT-L greedy), worse (ViT-GPT2 beam), or comparable (random baseline accuracy) results. Moreover, sample ordering (top versus inv. top) accuracy varies, with each strategy performing better in certain circumstances. Overall, it can be stated that few-shot prompting necessitates more extensive experimentation beyond the scope of the current study until a standard pattern arises, and it is possible that no pattern can be inferred at all.

We are going to present some qualitative results regarding few-shot prompting.

Fig. 5.2.7 contains candidates corresponding to the phrase "football goal", while GiT-L serves as the captioner. The in-context samples are demonstrated in Tab. 5.22 followed by the answer generated by GPT-3.5-turbo (in color). In the presented case, in-context prompting achieves in guiding GPT-3.5-turbo to select the correct candidate I.

Additionally, Fig. 5.2.8 contains candidates corresponding to the phrase "light beam". In Tab. 5.23 and Tab. 5.24 we demonstrate in-context samples followed by the answer generated by GPT-3.5-turbo using *random*

Figure 5.2.7: Candidate images for the phrase "football goal".

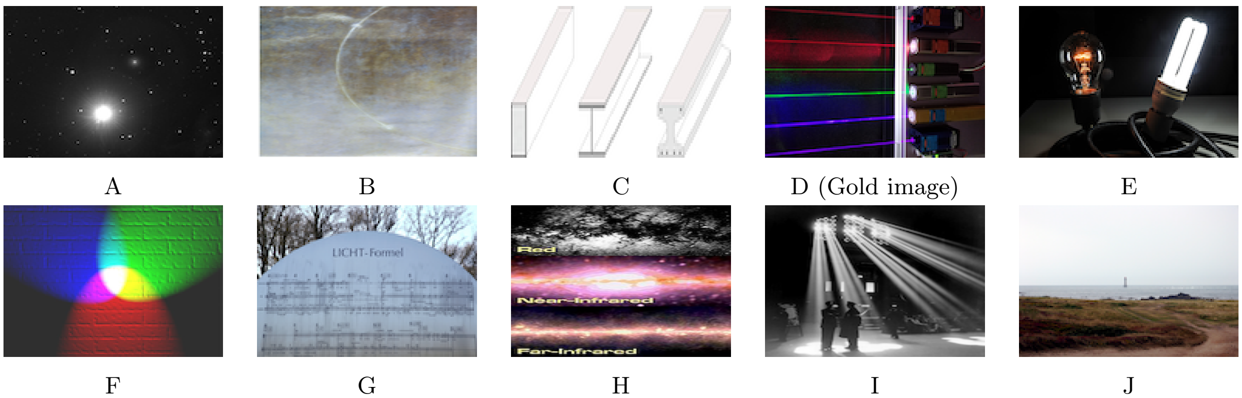

Figure 5.2.8: Candidate images for the phrase "light beam".

and *inverse-top* few-shot strategy respectively. We can observe that the *inverse-top* in-context samples selection achieves in guiding GPT-3.5-turbo to select the correct candidate D, while the base one (*random*) not.

Q: What is the most appropriate caption for the vanillin compound? Answer Choices: (A) a fossil fossil with a spiral pattern (B) a blue plate topped with pastries and a piece of bread. (C) the bones of the foot are very thin. (D) a spoon full of sugar on a table. (E) the tower of the building is made of brick. (F) a pair of sunglasses with a white background. (G) a close up of a plant with a yellow flower. (H) turmeric powder in a bowl (I) a man with glasses and a tie sitting at a desk. (J) a close up of a bike tire with a yellow tag on it. **A:** (D) a spoon full of sugar on a table.

Q: What is the most appropriate caption for the nankeen fabric? Answer Choices: (A) a close up of a white quilt with a tree design on it. (B) a man in a blue jacket and tan pants holding a sword. (C) a collection of brass trombones (D) a close up of a green plant (E) a pine tree in the forest (F) a row of mannequins in black dresses. (G) two boys are standing in front of a red wall. (H) two apples sitting next to each other on a white surface. (I) a cartoon of two women in fancy hats and a mirror. (J) a close up of a woman's face with a round ball on her forehead. **A:** (B) a man in a blue jacket and tan pants holding a sword.

Q: What is the most appropriate caption for the olmec indian? Answer Choices: (A) a large stone head in a garden (B) a woman lighting a candle at a table (C) [unused0] is the new champion of the world (D) a table full of food (E) a group of women holding banners in a protest. (F) a woman in a traditional dress stands in front of a tent. (G) a man in a suit sitting in a chair. (H) a clay pot with a small hole on top of it. (I) a close up of three green leaves of a plant (J) a fern in the forest near the lake **A:** (A) a large stone head in a garden

Q: What is the most appropriate caption for the charales order? Answer Choices: (A) a man walking past a store with a sign on it. (B) the road to the island (C) the art of the flower (D) a black snake with white spots on it's body. (E) a bird perched on a branch in a tree. (F) a lion roaring in the wild. (G) a man falling from a skateboard (H) a large, thin, green plant with long thin leaves. (I) a close up of a bush with blue berries (J) a mouse in a hollow log. **A:** (H) a large, thin, green plant with long thin leaves.

Q: What is the most appropriate caption for the skink lizard? Answer Choices: (A) two lizards on a log with a log in the background. (B) a man walking past a store with a sign on it. (C) the art of the flower (D) a portrait of [unused0], seated at a desk in front of a telescope. (E) the car is a small car that can be found in the museum. (F) a black snake with white spots on it's body. (G) a bird perched on a branch in a tree. (H) a close up of a bush with blue berries (I) a close up of a metal container with a white label on it. (J) a white door with a glass window **A:** (A) two lizards on a log with a log in the background.

Q: What is the most appropriate caption for the football goal? Answer Choices: (A) a large stadium with a large field and a large structure. (B) a woman wearing a blue ski mask (C) [unused0] of spain during the fifa world cup brazil group a match between spain and argentina at the estadio santiago bernabeu on june 25, 2010 in santiago, chile. (D) a woman lifting a barbell in a competition. (E) a man in a suit and tie looking at the camera. (F) person is a player for football team (G) [unused0], the brazilian national team, in the 1960s. (H) the empty stand at the old trafford stadium (I) a soccer field with a goal post in the middle (J) the emblem of the football club of the italian football club of the italian football club of the italian football club of the italian football club of the italian football club of the italian football club of the A: (I) a soccer field with a goal post in the middle ✓

Table 5.22: Example of few-shot prompting with k=5 in-context samples for disambiguating "football goal" phrase.

Q: What is the most appropriate caption for the hoopoe bird? Answer Choices: (A) a man walking past a store with a sign on it. (B) a close up of a yellow flower (C) a collection of posters and cards for a school project. (D) a surfer rides a large wave in the ocean. (E) the art of the flower (F) a black snake with white spots on it's body. (G) a bird perched on a branch in a tree. (H) a close up of a bush with blue berries (I) a painting of a bird with a waterfall in the background. (J) a spray gun and a glass jar. **A:** (I) a painting of a bird with a waterfall in the background.

Q: What is the most appropriate caption for the scallopine dish? Answer Choices: (A) a drawing of a stingray by [unused0] (B) bottles of chinese beer on a shelf in a supermarket. (C) a group of people sitting around a table. (D) a group of women are working on bananas. (E) a plate of food with a fork and a piece of meat on it. (F) a plate of food with a bun and spices. (G) a piece of pie on a plate on a table. (H) the game of go (I) soldiers in the old city of jerusalem (J) an ice cream sandwich is shown on a white background. **A:** (E) a plate of food with a fork and a piece of meat on it.

Q: What is the most appropriate caption for the taliban movement? Answer Choices: (A) a man with his hands wrapped in bandages. (B) a ship on the water. (C) a black and white photo of a soldier in uniform. (D) a ship on the water with people on it. (E) a close up of a worm (F) model of the ship of the line (G) a close up of a flower (H) soldiers walk through the streets of the city. (I) a sketch of a sculpture (J) the ship was the first ship to sail. **A:** (H) soldiers walk through the streets of the city.

Q: What is the most appropriate caption for the onomastico day? Answer Choices: (A) a woman lighting a candle at a table (B) a goat with long horns is laying down. (C) a large building with many windows (D) a table full of food (E) a group of women holding banners in a protest. (F) [unused0] and the [unused0] (G) a woman in a traditional dress stands in front of a tent. (H) a man in a suit sitting in a chair. (I) a page from the book of the german language (J) the most common herbs in the world **A:** (I) a page from the book of the german language

Q:What is the most appropriate caption for the chlamydosaurus genus? Answer Choices: (A) a man walking past a store with a sign on it. (B) the art of the flower (C) a cake with a slice cut out on a plate. (D) the ocean is a place where you can find the best place to live. (E) a black snake with white spots on it's body. (F) a bird perched on a branch in a tree. (G) a fruit on a tree branch. (H) a close up of a bush with blue berries (I) chest radiograph showing a chest xray showing a large chest xray. (J) a lizard with a yellow and orange body **A:** (J) a lizard with a yellow and orange body

Q: What is the most appropriate caption for the light beam? Answer Choices: (A) the bright star cluster in the constellation of constellation (B) [unused0],'the moon ', 2019, [unused0] (C) a diagram showing the steps to the stairs. (D) a laser beam with a laser on it (E) a light bulb and a black wire (F) a photo of a brick wall with a light on it. (G) a sign with a musical theme (H) a composite of the different types of galaxies. (I) a couple of men standing in a dark room with sun beams shining through the windows. (J) a lighthouse in the distance on a foggy day. A: (J) a lighthouse in the distance on a foggy day. **X**

Table 5.23: Example of random few-shot prompting with k=5 in-context samples for disambiguating "light beam" phrase.

Q: What is the most appropriate caption for the lightning flash? Answer Choices: (A) lightning strikes over a city in the evening. (B) a close up of a plant with green leaves and flowers (C) a variety of different types of items (D) a lizard on a rock in the sun (E) the oak tree of the genus [unused0], the [unused0], and the [unused0]. (F) a close up of a snowdrop flower (G) a poster from the book of kells (H) a book with a blue background and a black background. (I) a close up of a plant with red berries (J) a kangaroo standing on a rock **A:** (A) lightning strikes over a city in the evening.

Q: What is the most appropriate caption for the light spectrum? Answer Choices: (A) a bowl of soup on a plate (B) a turquoise stone with a green patina (C) a yellow and blue tie with a silver and blue tie. (D) a group of women standing outside of a store. (E) a laser beam with a laser on it (F) coin of the ancient roman empire (G) [unused0] in the girl in the mirror (1962) (H) a series of images showing the different views of the ocean. (I) a group of people sitting at desks in a classroom. (J) a close up of a pile of rocks. **A:** (E) a laser beam with a laser on it

Q: What is the most appropriate caption for the brightness light? Answer Choices: (A) poster for the first edition of the book (B) lotus flower in the pond (C) the ocean is a place where you can find the best place to live. (D) the moon's shadow is seen on the surface of the moon. (E) [unused0] drawing - the sword of [unused0] by [unused0] picture library (F) artist's concept of a space station in space (G) a plate of french toast with a bite taken out of it. (H) the [unused0]'s music notes (I) a diagram of the constellation (J) a cockroach crawling on the floor. **A:** (C) the ocean is a place where you can find the best place to live.

Q: What is the most appropriate caption for the light burn? Answer Choices: (A) the abandoned mine in the desert (B) a man looks at a display of russian dolls. (C) the u. s. army medal of the united states (D) a plant growing on a wall in the woods. (E) a black and white pottery vase with a scene of a scene of a scene of a scene of a scene of a scene of a woman and a man. (F) a woman hanging from a rope on a cliff (G) a hand holding a match in the dark (H) a pair of sunglasses with a white background. (I) turmeric powder in a bowl (J) a man with glasses and a tie sitting at a desk. **A:** (G) a hand holding a match in the dark

Q: What is the most appropriate caption for the luminescence light? Answer Choices: (A) poster for the first edition of the book (B) a beaker with a blue glow in it. (C) a small green bird with a red beak on a branch. (D) a tree in the park (E) the moon's shadow is seen on the surface of the moon. (F) vintage illustration of a crocodile (G) the anatomy of the tongue (H) artist's concept of a space station in space (I) a large crowd of people standing in a auditorium. (J) a diagram of the constellation **A:** (B) a beaker with a blue glow in it.

Q: What is the most appropriate caption for the light beam? Answer Choices: (A) the bright star cluster in the constellation of constellation (B) [unused0],'the moon ', 2019, [unused0] (C) a diagram showing the steps to the stairs. (D) a laser beam with a laser on it (E) a light bulb and a black wire (F) a photo of a brick wall with a light on it. (G) a sign with a musical theme (H) a composite of the different types of galaxies. (I) a couple of men standing in a dark room with sun beams shining through the windows. (J) a lighthouse in the distance on a foggy day. A:(D) a laser beam with a laser on it ✓

Table 5.24: Example of inverse-top few-shot prompting with k=5 in-context samples for disambiguating "light beam" phrase.

Chapter 6

Conclusion

In this study, we conducted extensive experiments and analyses on the novel Visual Word Sense Disambiguation (VWSD) task. As a start, we utilised state-of-the-art models for VL retrieval in order to establish strong baselines. Furthermore, we demonstrated the advantages of enhancing ambiguous phrases with external knowledge stored in Large Language Models (LLMs). By adopting the LLM-as-KB paradigm, we were able to improve the functionality of baseline visiolinguistic pipelines, and as a result the corresponding vl retrieval scores. More specifically, our current work is the first to utilize various prompting strategies to extract and incorporate the implicit rich knowledge stored in LLMs for the VWSD task. Furthermore, we conducted an investigation into the possibilities of unimodal approaches by transforming Visual Word Sense Disambiguation (VWSD) into an image-to-image or text-to-text retrieval task. Then, we tried to combine the variety of these independent experiments by training a lightweight retrieval module incorporating features extracted from our aforementioned experiments. In this way, we achieved our best competitive ranking results. Additionally, we explored the application of VWSD as a question-answer task, wherein generated image captions are utilised as multiple-choice questions. The Chain-of-Thought prompting technique has exploited either as another prompt strategy in the context of QA problem or as a explainability component. Chain-of-Thought prompting unveiled the reasoning process behind VWSD and highlighted aspects of human-interpretable explainability associated with the LLM-based knowledge extraction process. Overall, our analysis demonstrates the significance of model scale when using LLMs for knowledge-related tasks. Our results exceeded concurrent implementations and given baselines, demonstrating valuable insights that can influence future state-of-the-art implementations.

In conclusion, we would like to propose some avenues for further enhancing this research or inspire alternative approaches. Firstly, it is worth considering the potential of employing soft prompting as a means to facilitate the advance knowledge enrichment. An alternative methology involves the integration of Large Language Models (LLMs) with large-scale Knowledge Graphs to further enhance the ambiguous phrases. Moreover, the incorporation of LLM enhancement with QA and CoT strategies could be worth a chance, as well as the investigation of the explainability aspects of VWSD may provide a crucial avenue for future research. Undoubtedly, a detailed analysis on retrieval mistakes could reveal crucial information regarding the existence of failure patterns, i.e., very similar images that could lead to potential solutions. Finally, the experimentation with languages other than English, namely Italian and Persian, is a significant limitation that would be addressed in a future study.

Limitations

The present study is accompanied with specific limitations. Initially, it should be noted that our machines do not utilise very large LLMs with a parameter count exceeding 13 billion due to constraints in computational resources. Nevertheless, it is plausible that increasing the scale of these models would likely result in improved knowledge enhancement for given phrases. This assertion is further supported by the notable outcomes observed when integrating the 175 billion parameter GPT-3 into our experimental pipelines. The utilisation of GPT-3 and GPT-3.5-turbo was primarily directed at a certain experimental subgroup as a result of their elevated cost. However, in light of this constraint, we decided to allocate our efforts to several approaches for VWSD instead of concentrating on a certain route, such as LLM-based knowledge enhancement, and investigating the impact of larger language models. Furthermore, given the constraints imposed by limited resources, our findings have the potential to inspire researchers with restricted access to computing resources to replicate and expand upon our work. This allows for a broader range of individuals and organisations, regardless of their financial capabilities, to engage in such experimentation.

In addition to the aforementioned considerations, the LLM-enhancement technique is confronted with the risks associated with hallucinations and untruthful generation. These dangers, as indicated by relevant studies in the field [6, 59, 4, 100, 96], pose challenges in terms of detection and resolution. Such a shortcoming could have a detrimental effect on our findings. This is particularly relevant considering that certain phrases may necessitate specialised domain knowledge for accurate evaluation. For instance, the term *andromeda tree* is not commonly used, and even a human evaluator would need to consult an encyclopaedia to determine the validity of any related enhancement. Based on current understanding, there is currently no available open-source tool that can effectively and reliably detect hallucinations. Consequently, this problem remains unresolved at present. However, it is conceivable that the occurrence of hallucinations and the development of false information could be reduced by integrating LLM knowledge with knowledge graphs. Knowledge graphs are considered to be more dependable sources of information that can enhance VL tasks [55].

Finally, our experimentation has ultimately centred on the English language with the aim of devising a range of approaches for VWSD, as opposed to evaluating a specific subset of these techniques on other languages.

Ethics Statement

Our work utilises a newly introduced publicly accessible dataset that is licenced under CC-BY-NC 4.0 and is accessible to all researchers. Throughout this work, we adhered to the fair data usage policy, as required by the dataset creators[1]. We use language models with up to 13B parameters that were executed on a computer with two 14.8 GB GPUs. The vast majority of research institutions have easy access to such computational resources; consequently, throughout this paper, we advocate for equitable and reproducible research, eliminating the need for a high-end computational budget. Accessing larger variants such as GPT-3 and GPT-3.5-turbo was possible via their APIs, which imposed no computational limitations on the user. As it aims to expand the field of multimodal retrieval, there are no apparent risks associated with the task. Utilising language models as knowledge bases carries the risk of retrieving erroneous or inaccurate information, which, given the non-critical nature of this dataset, does not have significant implications for its current application. Overall, we do not anticipate any ethical issues to arise as a result of our work.

[1] https://raganato.github.io/vwsd/

Chapter 7

Bibliography

[1] Agrawal, A. et al. *VQA: Visual Question Answering*. 2016. arXiv: 1505.00468 [cs.CL].

[2] AlKhamissi, B. et al. *A Review on Language Models as Knowledge Bases*. 2022. arXiv: 2204.06031 [cs.CL].

[3] Amini, A. et al. "MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2357–2367. DOI: 10.18653/v1/N19-1245. URL:

[4] Azaria, A. and Mitchell, T. *The Internal State of an LLM Knows When its Lying*. 2023. arXiv: 2304.13734 [cs.CL].

[5] Ba, J. L., Kiros, J. R., and Hinton, G. E. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML].

[6] Bang, Y. et al. *A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity*. 2023. arXiv: 2302.04023 [cs.CL].

[7] Bengio, Y. et al. "A Neural Probabilistic Language Model". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. ISSN: 1532-4435.

[8] Bhargava, P. and Ng, V. *Commonsense Knowledge Reasoning and Generation with Pre-trained Language Models: A Survey*. 2022. DOI: 10.48550/ARXIV.2201.12438. URL:

[9] Brown, T. B. et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[10] Chowdhery, A. et al. *PaLM: Scaling Language Modeling with Pathways*. 2022. arXiv: 2204.02311 [cs.CL].

[11] Dadas, S. *OPI at SemEval 2023 Task 1: Image-Text Embeddings and Multimodal Information Retrieval for Visual Word Sense Disambiguation*. 2023. arXiv: 2304.07127 [cs.CL].

[12] Devlin, J. et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *ArXiv* abs/1810.04805 (2019).

[13] Dhingra, B. et al. "Time-Aware Language Models as Temporal Knowledge Bases". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 257–273. DOI: 10.1162/tacl_a_00459. URL:

[14] Dong, Q. et al. *A Survey on In-context Learning*. 2023. arXiv: 2301.00234 [cs.CL].

[15] Dosovitskiy, A. et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL:

[16] Feldman, J., Davison, J., and Rush, A. M. *Commonsense Knowledge Mining from Pretrained Models*. 2019. arXiv: 1909.00505 [cs.CL].

[17] Gao, T., Fisch, A., and Chen, D. *Making Pre-trained Language Models Better Few-shot Learners*. 2021. arXiv: 2012.15723 [cs.CL].

[18] Girshick, R. *Fast R-CNN*. 2015. arXiv: 1504.08083 [cs.CV].

[19] Guo, J. et al. "From Images to Textual Prompts: Zero-shot VQA with Frozen Large Language Models". In: *ArXiv* abs/2212.10846 (2022).

[20] Guo, J. et al. "From Images to Textual Prompts: Zero-shot VQA with Frozen Large Language Models". In: *ArXiv* abs/2212.10846 (2022).

[21] He, K. et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].
[22] He, K. et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
[23] Huang, J. and Chang, K. C.-C. "Towards Reasoning in Large Language Models: A Survey". In: *ArXiv* abs/2212.10403 (2022).
[24] Ji, S. et al. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (Feb. 2022), pp. 494–514. DOI: 10.1109/tnnls.2021.3070843. URL:
[25] Jia, C. et al. "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision". In: (2021). DOI: 10.48550/ARXIV.2102.05918. URL:
[26] Jiang, Z. et al. "How Can We Know What Language Models Know?" In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 423–438. DOI: 10.1162/tacl_a_00324. URL:
[27] Johnson, J., Douze, M., and Jégou, H. "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.
[28] Karpathy, A. and Fei-Fei, L. *Deep Visual-Semantic Alignments for Generating Image Descriptions*. 2015. arXiv: 1412.2306 [cs.CV].
[29] Ke, G. et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL:
[30] Kojima, T. et al. *Large Language Models are Zero-Shot Reasoners*. 2023. arXiv: 2205.11916 [cs.CL].
[31] Kritharoula, A., Lymperaiou, M., and Stamou, G. *Language Models as Knowledge Bases for Visual Word Sense Disambiguation*. 2023. arXiv: 2310.01960 [cs.CL].
[32] Kritharoula, A., Lymperaiou, M., and Stamou, G. *Large Language Models and Multimodal Retrieval for Visual Word Sense Disambiguation*. 2023. arXiv: 2310.14025 [cs.CL].
[33] Krizhevsky, A., Sutskever, I., and Hinton, G. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.
[34] Kumar, A. "The Illustrated Image Captioning using transformers". In: *ankur3107.github.io* (2022). URL:
[35] *LangChain Prompt templates*. URL:
[36] Le, Q. V. and Mikolov, T. *Distributed Representations of Sentences and Documents*. 2014. arXiv: 1405.4053 [cs.CL].
[37] Lester, B., Al-Rfou, R., and Constant, N. *The Power of Scale for Parameter-Efficient Prompt Tuning*. 2021. arXiv: 2104.08691 [cs.CL].
[38] Lewis, M. et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].
[39] Li, G. et al. *Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training*. 2019. arXiv: 1908.06066 [cs.CV].
[40] Li, J., Yu, L., and Ettinger, A. *Counterfactual reasoning: Do language models need world knowledge for causal understanding?* 2022. arXiv: 2212.03278 [cs.CL].
[41] Li, J. et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. arXiv: 2201.12086 [cs.CV].
[42] Li, L. H. et al. *VisualBERT: A Simple and Performant Baseline for Vision and Language*. 2019. arXiv: 1908.03557 [cs.CV].
[43] Li, X. L. and Liang, P. *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. 2021. arXiv: 2101.00190 [cs.CL].
[44] Li, X. L. et al. *A Systematic Investigation of Commonsense Knowledge in Large Language Models*. 2022. arXiv: 2111.00607 [cs.CL].
[45] Lin, T.-Y. et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].
[46] Liu, J. et al. "What Makes Good In-Context Examples for GPT-3?" In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. DOI: 10.18653/v1/2022.deelio-1.10. URL:
[47] Liu, P. et al. *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. 2021. arXiv: 2107.13586 [cs.CL].
[48] Liu, X. et al. *GPT Understands, Too*. 2021. arXiv: 2103.10385 [cs.CL].

[49] Liu, Y. et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[50] Liu, Z. et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV].

[51] Lu, J. et al. *ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks*. 2019. arXiv: 1908.02265 [cs.CV].

[52] Lu, P. et al. *Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering*. 2022. DOI: 10.48550/ARXIV.2209.09513. URL:

[53] Lu, Y. et al. "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8086–8098. DOI: 10.18653/v1/2022.acl-long.556. URL:

[54] Lymperaiou, M. and Stamou, G. *A survey on knowledge-enhanced multimodal learning*. 2023. arXiv: 2211.12328 [cs.LG].

[55] Lymperaiou, M. and Stamou, G. *The Contribution of Knowledge in Visiolinguistic Learning: A Survey on Tasks and Challenges*. 2023. arXiv: 2303.02411 [cs.CL].

[56] Mikolov, T. et al. "Recurrent neural network based language model." In: *INTERSPEECH*. Ed. by T. Kobayashi, K. Hirose, and S. Nakamura. ISCA, 2010, pp. 1045–1048. URL:

[57] Mikolov, T. et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[58] Muennighoff, N. et al. *Crosslingual Generalization through Multitask Finetuning*. 2023. arXiv: 2211.01786 [cs.CL].

[59] Mündler, N. et al. *Self-contradictory Hallucinations of Large Language Models: Evaluation, Detection and Mitigation*. 2023. arXiv: 2305.15852 [cs.CL].

[60] Pennington, J., Socher, R., and Manning, C. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL:

[61] Petroni, F. et al. *Language Models as Knowledge Bases?* 2019. arXiv: 1909.01066 [cs.CL].

[62] Pezeshkpour, P. *Measuring and Modifying Factual Knowledge in Large Language Models*. 2023. arXiv: 2306.06264 [cs.CL].

[63] Pourreza, M. and Rafiei, D. *DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction*. 2023. arXiv: 2304.11015 [cs.CL].

[64] Radford, A. et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[65] Radford, A. et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[66] Raffel, C. et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: 1910.10683 [cs.LG].

[67] Raganato, A. et al. "SemEval-2023 Task 1: Visual Word Sense Disambiguation". In: *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Toronto, Canada: Association for Computational Linguistics, July 2023.

[68] Reimers, N. and Gurevych, I. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL:

[69] Ren, S. et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV].

[70] Schuhmann, C. et al. "LAION-5B: An open large-scale dataset for training next generation image-text models". In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2022. URL:

[71] Sharma, P. et al. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2556–2565. DOI: 10.18653/v1/P18-1238. URL:

[72] Shi, F. et al. *Language Models are Multilingual Chain-of-Thought Reasoners*. 2022. arXiv: 2210.03057 [cs.CL].
[73] Sia, S. and Duh, K. *In-context Learning as Maintaining Coherency: A Study of On-the-fly Machine Translation Using Large Language Models*. 2023. arXiv: 2305.03573 [cs.CL].
[74] Simonyan, K. and Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].
[75] Srivastava, A. et al. *Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models*. 2022. arXiv: 2206.04615 [cs.CL].
[76] Su, W. et al. *VL-BERT: Pre-training of Generic Visual-Linguistic Representations*. 2020. arXiv: 1908.08530 [cs.CV].
[77] Sung, M. et al. *Can Language Models be Biomedical Knowledge Bases?* 2021. arXiv: 2109.07154 [cs.CL].
[78] Szegedy, C. et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].
[79] Tan, H. and Bansal, M. *LXMERT: Learning Cross-Modality Encoder Representations from Transformers*. 2019. arXiv: 1908.07490 [cs.CL].
[80] Taylor, R. et al. “GALACTICA: A Large Language Model for Science”. In: 2022.
[81] Thoppilan, R. et al. *LaMDA: Language Models for Dialog Applications*. 2022. arXiv: 2201.08239 [cs.CL].
[82] Tiong, A. M. H. et al. “Plug-and-Play VQA: Zero-shot VQA by Conjoining Large Pretrained Models with Zero Training”. In: *Conference on Empirical Methods in Natural Language Processing*. 2022.
[83] Touvron, H. et al. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint arXiv:2302.13971* (2023).
[84] Vaswani, A. et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL:
[85] Vrandečić, D. and Krötzsch, M. “Wikidata: A Free Collaborative Knowledgebase”. In: *Commun. ACM* 57.10 (Sept. 2014), pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489. URL:
[86] Wallace, E. et al. “Universal Adversarial Triggers for Attacking and Analyzing NLP”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2153–2162. DOI: 10.18653/v1/D19-1221. URL:
[87] Wang, C., Liu, X., and Song, D. *Language Models are Open Knowledge Graphs*. 2020. arXiv: 2010.11967 [cs.CL].
[88] Wang, J. et al. *GIT: A Generative Image-to-text Transformer for Vision and Language*. 2022. arXiv: 2205.14100 [cs.CV].
[89] Wang, X. et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2023. arXiv: 2203.11171 [cs.CL].
[90] Wei, J. et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL].
[91] Winston, P. H. “Learning and Reasoning by Analogy”. In: *Commun. ACM* 23.12 (Dec. 1980), pp. 689–703. ISSN: 0001-0782. DOI: 10.1145/359038.359042. URL:
[92] Workshop, B. et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2023. arXiv: 2211.05100 [cs.CL].
[93] Young, P. et al. “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 67–78. DOI: 10.1162/tacl_a_00166. URL:
[94] Young, P. et al. “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 67–78. DOI: 10.1162/tacl_a_00166. URL:
[95] Zellers, R. et al. *From Recognition to Cognition: Visual Commonsense Reasoning*. 2019. arXiv: 1811.10830 [cs.CV].
[96] Zhang, M. et al. *How Language Model Hallucinations Can Snowball*. 2023. arXiv: 2305.13534 [cs.CL].
[97] Zhang, S. et al. *OPT: Open Pre-trained Transformer Language Models*. 2022. arXiv: 2205.01068 [cs.CL].
[98] Zhang, Z. et al. *Multimodal Chain-of-Thought Reasoning in Language Models*. 2023. DOI: 10.48550/ARXIV.2302.00923. URL:

[99] Zheng, L. et al. *Judging LLM-as-a-judge with MT-Bench and Chatbot Arena*. 2023. arXiv: 2306.05685 [cs.CL].

[100] Zheng, S., Huang, J., and Chang, K. C.-C. *Why Does ChatGPT Fall Short in Providing Truthful Answers?* 2023. arXiv: 2304.10513 [cs.CL].

[101] Zhou, D. et al. *Least-to-Most Prompting Enables Complex Reasoning in Large Language Models*. 2023. arXiv: 2205.10625 [cs.AI].