National Technical University of Athens
School of Electrical and Computer Engineering

DIVISION OF SIGNALS, CONTROL AND ROBOTICS
COMPUTER VISION, SPEECH COMMUNICATION AND SIGNAL PROCESSING GROUP

Enhancing Video Question Answering with the use of Scene Graphs

DIPLOMA THESIS

of

Dionysia Danai Brilli

Supervisor: Petros Maragos
Professor NTUA

Co-Supervisor: Vassilis Pitsikalis
Deeplab.ai

Athens, March 2024

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

Enhancing Video Question Answering with the use of Scene Graphs

DIPLOMA THESIS

of

Dionysia Danai Brilli

Supervisor: Petros Maragos
Professor NTUA

Approved by the Examining Committee on the 28th March 2024.

....	....	....
Πέτρος Μαραγκός	Αθανασιος Ροντογιαννης	Ιωάννης Κορδώνης
Καθηγητής ΕΜΠ	Αναπληρωτής Καθηγητής ΕΜΠ	Επίκουρος Καθηγητής ΕΜΠ

Athens, March 2024

....

Dionysia Danai Brilli
Electrical and Computer Engineering Graduate, NTUA

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the National Technical University of Athens.

Περίληψη

Στη σύγχρονη εποχή της ψηφιακής επανάστασης, με την εκθετική αύξηση του περιεχομένου σε βίντεο, είναι πλέον επιτακτική η ανάγκη για αποτελεσματική κατανόηση και ερμηνεία των βίντεο, κάτι ζωτικής σημασίας για πολλές εφαρμογές. Η απάντηση ερωτήσεων σε βίντεο (Video Question Answering) είναι ένα πολύπλοκο πρόβλημα που απαιτεί βαθιά κατανόηση τόσο του οπτικού περιεχομένου όσο και των φυσικών γλωσσικών ερωτήσεων. Παρόλο που έρευνες παρουσιάζουν συνεχή πρόοδο, οι περισσότερες δουλειές μέχρι σήμερα έχουν επικεντρωθεί σε μεθόδους που βασίζονται σε εικονοστοιχεία (pixel), ενώ συχνά δυσκολεύονται να αποτυπώσουν αποτελεσματικά τις πολύπλοκες σχέσεις και δυναμικές εντός του βίντεο. Η παρατήρηση της συμπεριφοράς των μοντέλων αυτών έχει αναδείξει αυτόν τον περιορισμό και την ανάγκη για την ανάπτυξη πιο αποτελεσματικών και με επίγνωση του περιεχομένου συστημάτων Video Question Answering.

Η παρούσα εργασία παρουσιάζει μία νέα προσέγγιση προς αυτή την κατεύθυνση με την ενσωμάτωση των γράφων σκηνής με μία ιεραρχική προσέγγιση για την πιο αποτελεσματική απάντηση ερωτήσεων σε βίντεο. Οι γράφοι σκηνής παρέχουν μία δομημένη αναπαράσταση των οπτικών στοιχείων μέσα σε ένα βίντεο και των μεταξύ τους σχέσεων, προσφέροντας μία πλούσια σημασιολογική βάση για την κατανόηση σύνθετων βίντεο. Μετατρέποντας την ανάλυση βίντεο από τον χώρο των πίξελ στον χώρο των γράφων, δίνουμε τη δυνατότητα αποτελεσματικότερης και σημασιολογικά πλούσιας επεξεργασίας βίντεο.

Προτείνουμε μία αρχιτεκτονική που αξιοποιεί τους γράφους σκηνής, χρησιμοποιώντας Νευρωνικά Δίκτυα Γράφων (GNNs) για την επεξεργασία των γράφων σκηνής, μαζί με ένα ιεραρχικό μοντέλο που λειτουργεί σε διαφορετικά επίπεδα του βίντεο, από μεμονωμένα κλιπ, έως και ολόκληρο το βίντεο για να επιτρέψει πιο ολοκληρωμένη κατανόηση του βίντεο. Η ενσωμάτωση των GNNs επιτρέπει την εξαγωγή σημαντικών πληροφοριών για τους γράφους, αποτυπώνοντας τις σχέσεις και τα χαρακτηριστικά των οπτικών στοιχείων. Το ιεραρχικό μοντέλο, που λειτουργεί σε διαφορετικά επίπεδα, διασφαλίζει ότι λαμβάνονται υπόψη τόσο οι λεπτομέρειες όσο και το ευρύτερο περιεχόμενο, οδηγώντας σε βαθύτερη κατανόηση του βίντεο.

Έτσι, παρουσιάζουμε μία μέθοδο που(1) Ξεκινά με την εξαγωγή γράφων σκηνής από επιλεγμένα κλιπ βίντεο (2) Δημιουργεί διανύσματα χαρακτηριστικών με τη χρήση GNNs και (3) Ενσωματώνει τα διανύσματα χαρακτηριστικών σε ένα ιεραρχικό μοντέλο

Αξιολογούμε τη μέθοδό μας στο Action Genome Question Answering Dataset [14], ένα σύνολο δεδομένων πραγματικού κόσμου που απεικονίζει ανθρώπους σε καθημερινές δραστηριότητες. Τα αποτελέσματά μας δείχνουν ότι η προσέγγισή μας είναι μεταξύ των state-of-the-art μεθόδων, ενώ μάλιστα υπερτερεί σε συγκεκριμένες κατηγορίες ερωτήσεων. Η προσέγγισή μας είναι ένα βήμα προς πιο αποδοτικά και με επίγνωση του περιεχομένου συστήματα Video Question Answering, επιτρέποντας πιο ακριβείς και με ουσία απαντήσεις σε ερωτήσεις φυσικής γλώσσας σχετικά με βίντεο.

Εν κατακλείδι, η παρούσα εργασία παρουσιάζει μία νέα προσέγγιση για την απάντηση ερωτήσεων σε βίντεο, η οποία επικεντρώνεται στην αποτελεσματική κατανόηση και ερμηνεία των βίντεο. Η προσέγγισή μας είναι η πρώτη, εξ όσων γνωρίζουμε, που χρησιμοποιεί γράφους σκηνής μαζί με ιεραρχική προσέγγιση για το πρόβλημα του Video Question Answering, ενώ ακόμα τα αποτελέσματά μας αποδεικνύουν την αποτελεσματικότητα της προσέγγισής μας σε σενάρια πραγματικού κόσμου. Πειραματιζόμαστε ακόμα με διαφορετικές μεθόδους επεξεργασίας των γράφων σκηνής αλλά και επίπεδα του ιεραρχικού μοντέλου, παρέχοντας πληροφορίες σχετικά με την αποτελεσματικότητα διαφορετικών αρχιτεκτονικών.

Λέξεις Κλειδιά Βαθειά Μάθηση, Αυτόματη απάντηση ερωτήσεων σε βίντεο, Γράφοι Σκηνής, Παραγωγή Γράφων Σκηνής, Νευρωνικά Δίκτυα Γράφων, Action Genome Question Answering Dataset, Κατανόηση Βίντεο

Abstract

In the digital era, with the exponential growth in video content, efficiently understanding and interpreting videos has become crucial for numerous applications. Video Question Answering (VQA) is a complex task that requires deep understanding of both visual content and natural language queries.While works have continually shown progress, most of the advances to date have focused on pixel-based methods, often struggling to capture the intricate relationships and dynamics within video content effectively. Observing the behavior of state-of-the-art models has underscored this limitation and the necessity to develop more efficient and context-aware Video Question Answering systems.

This thesis presents a novel approach towards this direction by integrating Scene Graphs with a Hierarchical Conditional Approach to efficiently answer questions about Videos. Scene graphs provide a structured representation of the visual elements within a video and their interrelations, offering a rich semantic foundation for understanding complex video data. By transforming the video analysis from pixel to graph space we enable more efficient and semantically rich video processing.

We propose an architecture that leverages scene graphs, utilizes Graph Neural Networks (GNNs) for processing scene graphs, alongside a hierarchical model that operates at different levels of video granularity, from individual clips to the entire video, to enable a comprehensive understanding of video content. The integration of GNNs allows for the extraction of meaningful graph embeddings that capture the relationships and attributes of the visual elements, leading to a deeper understanding of the video content. The hierarchical model, operating at different levels, ensures that both the details and the broader context are considered, leading to a more holistic understanding of the video content.

So, we introduce a methodology that (1) Begins with the extraction of scene graphs from selected video frames, (2) Generates graph embeddings using GNNs and (3) Incorporates the graph embeddings into a hierarchical model

We evaluate our method on the Action Genome Question Answering Dataset [14], a real-world dataset consisting of videos depicting humans in everyday activities. Our results demonstrate that our approach is among state-of-the-art methods, and even outperforms them in several question categories. Our approach is a step towards more efficient and context-aware Video Question Answering systems, enabling more accurate and meaningful responses to natural language queries about videos.

In conclusion, this study presents a novel approach to Video Question Answering, focusing on the efficient understanding and interpretation of videos. Our approach is the first to our knowledge to use scene graphs along with a hierarchical conditional approach for Video Question Answering, and our results demonstrate the effectiveness of our approach in real-world scenarios. We also experiment with different graph processing methods and levels of the hierarchical model, providing insights into the effectiveness of different architectures.

Keywords Deep Learning, Video Question Answering, Scene Graphs, Scene Graph Generation, Graph Neural Networks, Hierarchical Conditional Relation Networks, Action Genome Question Answering Dataset, Video Understanding

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Petros Maragos, for the opportunity to work on this thesis and his supervision throughout the process. His courses were my first introduction to computer vision and machine learning and sparked my passion for this field.

I am very grateful and would like to give a warm thank you to Vassilis Pitsikalis, Dimitris Mallis, Markos Diomataris and the rest of Deeplab for welcoming me into their team, inspiring me, and providing me with the opportunity to work on this exciting project. I am grateful for their guidance, mentorship, and collaboration, which have been instrumental in the completion of this thesis.

Finally, I would like to thank my family and friends for their support and encouragement throughout this journey.

Contents

List of Figures

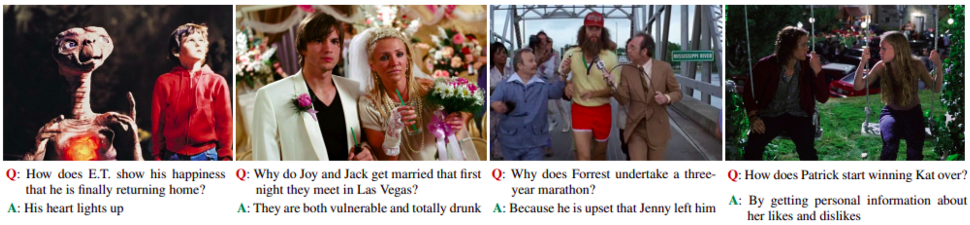
List of Tables

0 Εκτεταμένη Περίληψη στα Ελληνικά

0.1 Εισαγωγή

Στη σύγχρονη ψηφιακή εποχή, η ζωή μας είναι συνυφασμένη με δεδομένα, από την ψηφιακή καταγραφή της καθημερινότητάς μας από τα smartphones, μέχρι τις πλατφόρμες κοινωνικής δικτύωσης που καταγράφουν τις σκέψεις και τις αλληλεπιδράσεις μας [3]. Οι πηγές πληροφορίας πολλαπλασιάζονται συνεχώς, αυξάνοντας σημαντικά τον όγκο των ψηφιακών δεδομένων που παράγουμε και καταναλώνουμε. Κάθε μέρα παράγονται περίπου 328 εκατομμύρια Terrabyte δεδομένων, ενώ τα βίντεο αποτελούν πάνω από το μισό της παγκόσμιας κίνησης δεδομένων [11]! Μέσα σε αυτή την έκρηξη δεδομένων, η πρόκληση της επεξεργασίας και κατανόησης τεράστιων συνόλων δεδομένων γίνεται ολοένα και πιο δύσκολη, καθώς τα συστήματα που είναι σε θέση να τα κατανοήσουν δεν έχουν εξελιχθεί με τον ίδιο ρυθμό.

Για την αλληλεπίδραση με αυτή την οπτική πληροφορία, την εικόνα και το βίντεο, είναι απαραίτητη η εξέλιξη συστημάτων Τεχνητής Νοημοσύνης ικανά να την κατανοήσουν. Συγκεκριμένα, στην προσπάθεια κάλυψης αυτής της ανάγκης έχει αναδυθεί ο τομέας του Video Question Answering (Video QA), συνδυάζοντας την επεξεργασία φυσικής γλώσσας (natural language processing) με την όραση υπολογιστών (computer vision) [1]. Δεδομένου ενός βίντεο και μίας ερώτησης σχετικά με αυτό, το πρόβλημα του Video Question Answering φορά την σωστή απάντηση της ερώτησης με βάση τις διαθέσιμες πολυτυπικές (multimodal) πληροφορίες του βίντεο. Πλέον έχει δημιουργηθεί πληθώρα συνόλων δεδομένων για το Video Question Answering, μεταξύ των οποίων το ActivityNet-QA [54], MovieQA [41], KnowIT VQA [13], TGIF-QA [24], AGQA [14] και πολλά άλλα [58].Το Video Question Answering μπορεί να έχει εφαρμογές σε πολλούς τομείς, όπως στην εκπαίδευση, την ψυχαγωγία, την ασφάλεια, την υγεία, την επιστημονική έρευνα, την αυτόνομη οδήγηση, την ανάλυση βίντεο και πολλά άλλα.



Q: How does E.T. show his happiness that he is finally returning home?
A: His heart lights up

Q: Why do Joy and Jack get married that first night they meet in Las Vegas?
A: They are both vulnerable and totally drunk

Q: Why does Forrest undertake a three-year marathon?
A: Because he is upset that Jenny left him

Q: How does Patrick start winning Kat over?
A: By getting personal information about her likes and dislikes

Figure 0.1: Παραδείγματα Video Question Answering από το dataset MovieQA. Απεικονίζεται ένα μόνο στιγμιότυπο (frame), ενώ κανονικά όλες οι ερωτήσεις-απαντήσεις αντιστοιχούν σε μεγαλύτερης διάρκειας βίντεο κλιπ. Η εικόνα είναι από το [41].

Το Video QA είναι ένα σύνθετο πεδίο, καθώς απαιτεί την αναγνώριση των δράσεων, την χρονική τοποθέτησή τους μέσα στο βίντεο, την κατανόηση φυσικής γλώσσας και την συλλογιστική ικανότητα για την απάντηση των ερωτήσεων. Τα συστήματα Video Question Answering λοιπόν αντιμετωπίζουν σημαντικές υπολογιστικές προκλήσεις, καθώς επεξεργάζονται μεγάλες ποσότητες οπτικών δεδομένων και ερμηνεύουν πολύπλοκες σχέσεις μεταξύ των στοιχείων. Οι παραδοσιακές μέθοδοι επεξεργασίας βίντεο λειτουργούν σε επίπεδο pixel, οπότε τείνουν να εστιάζουν σε οπτικές λεπτομέρειες, συχνά παραλείποντας την κατανόηση του ευρύτερου πλαισίου, απαιτώντας πολλούς υπολογιστικούς πόρους. Επιπλέον, τα βίντεο ποικίλουν σε μεγάλο βαθμό ως προς την ανάλυση, την δειγματοληψία, ενώ περιέχουν πολλές διαφορετικές πληροφορίες με μοναδικά χαρακτηριστικά.

Οι προκλήσεις αυτές αφορούν κυρίως την περίπλοκη φύση των δεδομένων βίντεο που χαρακτηρίζονται από τον αδόμητο και πυκνό χώρο των πίξελ. Τα βίντεο είναι ιδιαίτερα απαιτητικά σε υπολογιστικούς πόρους, καθώς περιέχουν μεγάλο αριθμό στιγμιοτύπων (frames), οδηγώντας πολλές φορές σε μεγάλα μεγέθη αρχείων και δύσκολη επεξεργασία. Ακόμη, διαφορές στην ανάλυση μπορεί να εισάγουν άσχετες λεπτομέρειες ή να δημιουργήσουν παραμορφώσεις, ενώ βίντεο μπορεί να περιέχουν μοναδικά χαρακτηριστικά.

Για την αντιμετώπιση αυτών των ζητημάτων, η παρούσα εργασία εισάγει μία νέα προσέγγιση για την ενίσχυση της απόδοσης του Video Question Answering με τη χρήση γράφων σκηνής και μίας ιεραρχικής αρχιτεκτονικής. Οι γράφοι σκηνής αποτελούν μία δομημένη αναπαράσταση του περιεχομένου ενός βίντεο, παρέχοντας πιο σημαντικές (salient) πληροφορίες για τα οπτικά στοιχεία και τις μεταξύ τους σχέσεις. Προτείνουμε λοιπόν τη μετάβαση από την ανάλυση τωου χώρου των pixel στην ανάλυση του χώρου των γράφων. Παράλληλα χρήση μίας ιεραρχικής προσέγγισης μας επιτρέπει να λειτουργούμε σε δύο επίπεδα, σε επίπεδο σύντομου βίντεο κλιπ και σε επίπεδο ολόκληρου βίντεο, αποτυπώνοντας καλύτερα τις χωροχρονικές σχέσεις στο βίντεο.

Διαμορφώνουμε λοιπόν την προσέγγισή μας γύρω από το ακόλουθο ερευνητικό ερώτημα:

> ”Μπορούμε να αναλύσουμε τα βίντεο σε δομημένους γράφους και να απαντήσουμε σε ερωτήσεις βίντεο χρησιμοποιώντας αυτούς τους γράφους αντί για το βίντεο?”

Με αυτή την εργασία προσπαθούμε να απαντήσουμε στην παραπάνω ερώτηση, εστιάζοντας στο video question answering με βάση τους γράφους (graph-based Video QA). Οι συνεισφορές μας περιλαμβάνουν:

- Είμαστε οι πρώτοι -απ’ όσο γνωρίζουμε- που χρησιμοποιούν ρητούς γράφους σκηνής ως ενδιάμεση αναπαράσταση για το Video QA, μεταβαίνοντας από το χώρο των pixel στο χώρο των γράφων για πιο σημασιολογικά πλούσιες αναπαραστάσεις.

- Πειραματηστήκαμε με διάφορες μεθόδους εξαγωγής διανυσμάτων χαρακτηριστικών (embeddings) για τους γράφων σκηνής, δοκιμάζοντας πολλές αρχιτεκτονικές Graph Neural Networks (GNNs), για την καλύτερη αποτύπωση των σχέσεων και των χαρακτηριστικών των οπτικών στοιχείων.

- Ενσωματώσαμε το παραπάνω με ένα χρονικό νευρωνικό δίκτυο. Χρησιμοποιήσαμε μία αρχιτεκτονική transformer, παραλλαγή του ιεραρχικού δικτύου HCRN [26] που λειτουργεί σε δύο επίπεδα, σε επίπεδο σύντομου βίντεο κλιπ και σε επίπεδο ολόκληρου βίντεο, επιτρέποντας την καλύτερη κατανόηση του περιεχομένου του βίντεο.

- Αξιολογήσαμε τη μέθοδό μας στο Action Genome Question Answering Dataset [14], δείχνοντας ότι η προσέγγισή μας είναι ανταγωνιστική με τις state-of-the-art μεθόδους, ξεπερνώντας τες σε αρκετές κατηγορίες ερωτήσεων.

0.2 Υπόβαθρο

Η Μηχανική Μάθηση είναι ένας κλάδος της Τεχνητής Νοημοσύνης, που επικεντρώνεται στην ανάπτυξη συστημάτων που μπορούν να μαθαίνουν μόνα τους, από την εμπειρία, όπως ο άνθρωπος. Σκοπός της είναι να χρησιμοποιήσει δεδομένα για να μιμηθεί τον τρόπο που μαθαίνουν οι άνθρωποι, εντοπίζοντας μοτίβα και λαμβάνοντας αποφάσεις. Με τη χρήση διαφορετικών μεθόδων, αρχιτεκτονικών και παραμέτρων, σύγχρονες προσεγγίσεις της Μηχανικής Μάθησης μπορούν να επιλύσουν πολύπλοκα προβλήματα, μέσα από τη διαδικασία της εκπαίδευσης. Για την εκπαίδευση αξιοποιείται ένα ευρύ φάσμα τύπων δεδομένων, από αριθμητικά δεδομένα έως πιο σύνθετες μορφές, όπως κείμενο, ήχος ή βίντεο. Το καθένα από αυτά παρουσιάζει μοναδικές προκλήσεις και ευκαιρίες για την ανάπτυξη νέων μοντέλων. Στο επίκεντρο της μεθοδολογίας της Μηχανικής Μάθησης βρίσκονται τρεις βασικοί τύποι μάθησης: η επιβλεπόμενη μάθηση (supervised learning), η μη επιβλεπόμενη μάθηση (unsupervised learning) και η ενισχυτική μάθηση (reinforcement learning), [34] όπως φαίνεται και στο Σχήμα 0.2.

Βασική αρχή της Μηχανικής Μάθησης είναι ο νευρώνας - perceptron. Αποτελεί έναν δυαδικό ταξινομητή, ο οποίος χρησιμοποιείται για γραμμικά διαχωρίσιμα προβλήματα στην επιβλεπόμενη μάθηση. Μπορεί να κατηγοριοποιήσει τις εισόδους με βάση τα βάρη, τα οποία και προσαρμόζει κατά τη διάρκεια της εκπαίδευσης σύμφωνα με τα σφάλματα πρόβλεψης. Για πιο πολύπλοκα, μη γραμμικά διαχωρίσιμα προβλήματα, χρησιμοποιούνται πολλαπλοί νευρώνες, σε πολλαπλά επίπεδα, σχηματίζοντας τα Νευρωνικά Δίκτυα (Neural Networks ή Multi Layer Perceptron - MLP).

Η βαθειά μάθηση (deep learning) είναι μία υποκατηγορία της Μηχανικής Μάθησης, που χρησιμοποιεί αλγορίθμους εμπνευσμένους από τη δομή και τη λειτουργία των νευρωνικών δικτύων του εγκεφάλου. Περιλαμβάνει τη χρήση μεγαλύτερων νευρ-

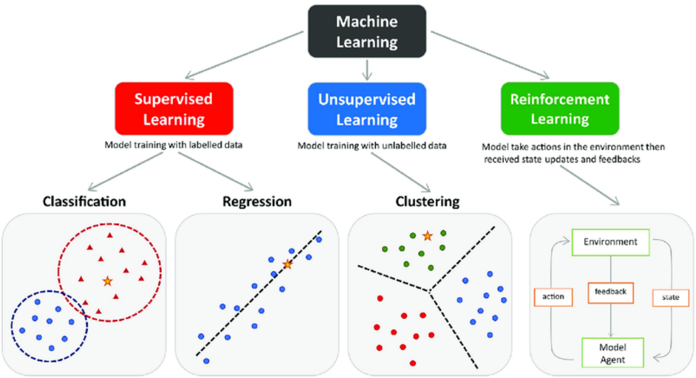


Figure 0.2: Οι κύριοι τύποι μάθησης. Σχήμα από το [34].

ωνικών δικτύων, με περισσότερα επίπεδα, που μπορούν να μάθουν πολύπλοκες αναπαραστάσεις και σύνθετα μοτίβα σε μεγάλες ποσότητες δεδομένων. Το πιο βασικό μοντέλο, FeedForward Neural Network (FFNN), αποτελείται από πολλά επίπεδα νευρώνων, όπου κάθε νευρώνας συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου. Σε αντίθεση με άλλες αρχιτεκτονικές, τα FFNN είναι αναλλοίωτα σε μεταθέσεις (permutation invariant), καθιστώντας τα ιδανικά για προβλήματα με ανεξάρτητα σημεία δεδομένων.

Τα συνελικτικά νευρωνικά δίκτυα (CNNs), τα οποία παρουσιάστηκαν για πρώτη φορά το 1998 [8], υπερέχουν στην επεξεργασία εικόνων, χρησιμοποιώντας πράξεις συνέλιξης που διατηρούν τις χωρικές σχέσεις μεταξύ των pixel. Αυτό επιτυγχάνεται με την ολίσθηση φίλτρων πάνω στην εικόνα εισόδου για την παραγωγή χαρακτηριστικών για την αναγνώριση εικόνων και την εξαγωγή πληροφορίας. Το ResNet, που προτάθηκε το 2015 [18], είναι μία ξεχωριστή προσέγγιση, που χρησιμοποιεί την έννοια των συντομοδοτούμενων συνδέσεων και των υπολειμματικών μπλοκ (residual blocks) για την εκπαίδευση πολύ βαθιών νευρωνικών δικτύων.

Τα αναδρομικά νευρωνικά δίκτυα (RNN) ξεχωρίζουν στην επεξεργασία ακολουθιών δεδομένων, όπως κείμενο, ήχος και βίντεο. Αντιμετωπίζουν βέβαια προκλήσεις με τις μακροπρόθεσμες εξαρτήσεις, την απώλεια πληροφορίας και την αδυναμία της εκπαίδευσης μεγάλων δικτύων, οι οποίες μετριάζονται από προηγμένες παραλλαγές όπως τα LSTM [20] και GRU [6] που εισάγουν μηχανισμούς ελέγχου για την αποθήκευση και την ανάκληση πληροφορίας.

Έπειτα, οι μηχανισμοί προσοχής έχουν εξελίξει τα μοντέλα νευρωνικών δικτύων, επιτρέποντας την εστίαση σε συγκεκριμένα στοιχεία της εισόδου, ανάλογα με την ανάγκη. Τα μοντέλα αυτά είναι ιδιαίτερα χρήσιμα σε προβλήματα με μεγάλες ακολουθίες, όπως κείμενο ή ομιλία. Αυτή η καινοτομία, με παράδειγμα το μοντέλο Transformer [47] χρησιμοποιεί την αυτοπροσοχή (self-attention) και έχει καταφέρει να βελτιώσει την απόδοση σε πολλά προβλήματα επεξεργασίας φυσικής γλώσσας. Μάλιστα, νεότερα μοντέλα, όπως το BERT [10] και το GPT [52], έχουν επιτύχει εντυπωσιακές επιδόσεις μετά από εκπαλίδευση σε μεγάλα σύνολα δεδομένων.

Φυσικά, η αντίληψη του κόσμου απαιτεί την επεξεργασία πολλαπλών τύπων δεδομένων, όπως κείμενο, εικόνες, ήχο και βίντεο. Η πολυτυπική επεξεργασία δεδομένων (multimodal machine learning) στοχεύει στην ενσωμάτωση και ερμηνεία πληροφοριών από πολλαπλές πηγές, προκειμένου να προσφέρει πιο πλήρη και συνεκτική κατανόηση του περιεχομένου. Μάλιστα, η αλληλεπίδραση μεταξύ όρασης και γλώσσας, η οποία έχει κεντρικό ρόλο σε πολλές ανθρώπινες εργασίες, οδηγεί την έρευνα σε τομείς όπως η περιγραφή εικόνων (image captioning), η οπτική απάντηση ερωτήσεων (visual question answering) [1] και πολλά άλλα.

Το Video Question Answering (Video QA) είναι ένας σχετικά νέος τομέας, που στοχεύει στην ανάπτυξη συστημάτων που μπορούν να απαντήσουν σε ερωτήσεις σχετικά με το περιεχόμενο ενός βίντεο. Το Video QA απαιτεί την κατανόηση των οπτικών στοιχείων, των ενεργειών, των σκηνών και των αλληλεπιδράσεων μεταξύ τους, καθώς και την ανάλυση του χρονικού περιεχομένου. Για την αξιολόγηση των συστημάτων Video QA, χρησιμοποιείται η μετρική του accuracy, το οποίο ορίζεται ως το ποσοστό των σωστών απαντήσεων από όλες τις απαντήσεις - για τις multi-choice - και ως το ποσοστό σωστών λέξεων της απάντησης από όλη την απάντηση - για τις open-ended ερωτήσεις. Για την εκπαίδευση των συστημάτων Video QA, χρησιμοποιούνται μεγάλα σύνολα δεδομένων, όπως το MovieQA [41], το TGIF-QA [24], το KnowIt VQA [13] και το Action Genome Question Answering [14].

0.3 Βιβλιογραφική Ανασκόπηση

Το Video Question Answering έχει αναπτυχθεί τα τελευταία χρόνια, με την εμφάνιση πολλών σημαντικών ερευνών και συστημάτων. Οι προσεγγίσεις αυτές παρουσιάζουν μεγάλη ποικιλομορφία, αλλά μπορούν να χωριστούν σε 3 κύριες κατηγορίες: Memory Networks, Transformers, Graph-Based προσεγγίσεις.

Τα Memory Networks είναι μία κατηγορία μοντέλων που χρησιμοποιούν μνήμες για την αποθήκευση πληροφορίας και την ανάκλησή της κατά την απάντηση σε ερωτήσεις. Αυτό τα καθιστά ιδανικά για την κατανόηση βίντεο μεγάλης διάρκειας και σύνθετων αφηγήσεων. Το πιο σχετικό με τη δουλειά μας είναι το Heterogeneous Memory Enhanced Multimodal Attention Model (HME) [5], το οποίο χρησιμοποιεί μνήμες για την αναπαράσταση των οπτικών και γλωσσικών στοιχείων, τα οποία και επεξεργάζεται με τη χρήση LSTMs, όπως βλέπουμε στο Σχήμα 3.1. Η παράλληλη επεξεργασία της γλώσσας και της οπτικής πληροφορίας, μαζί με την ενσωμάτωση μνημών, βελτιώνει την απόδοση στο Video QA.

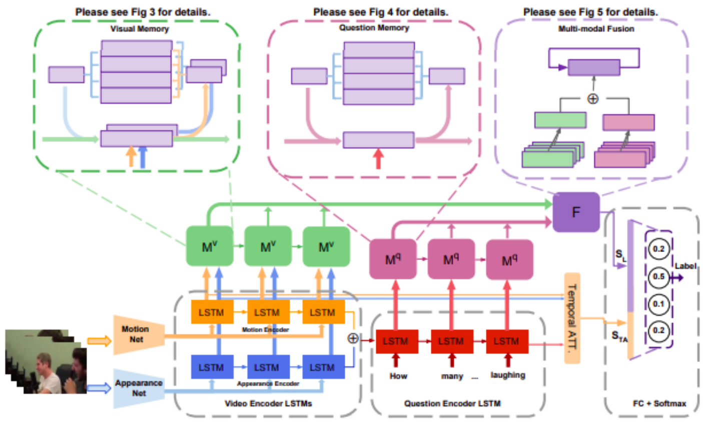


Figure 0.3: Η αρχιτεκτονική του HME. Σχήμα από το [5].

Οι Transformers είναι ένας πολύ ισχυρός μηχανισμός για την επεξεργασία ακολουθιακών δεδομένων. Ο βασικός μηχανισμός τους είναι η αυτοπροσοχή (self-attention), με την οποία ξεχωρίζουν περιοχές της εισόδου που είναι σημαντικές για την ανάλυση της ερώτησης. Η πιο σχετική ποσέγγιση με τη δουλειά μας είναι το Hierarchical Conditional Relation Network (HCRN) [26], το οποίο αποτέλεσε και έμπνευση για εμάς. Την αρχιτεκτονική του βλέπουμε στο Σχήμα 0.4. Με τη χρήση νευρώνων Conditional Relational Neurons (CRNs), για την αναπαράσταση των σχέσεων μεταξύ των οπτικών στοιχείων, το HCRN επιτυγχάνει εξαιρετικά αποτελέσματα στο Video QA. Παράλληλα, είναι το πρώτο μοντέλο που ενσωματώνει την ιεραρχική δομή των βίντεο, με τη χρήση δύο επιπέδων, σε επίπεδο σύντομου βίντεο κλιπ και σε επίπεδο ολόκληρου βίντεο, για την καλύτερη κατανόηση του περιεχομένου του βίντεο.

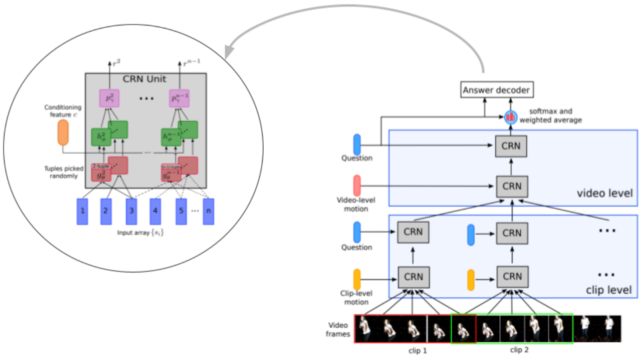


Figure 0.4: Η αρχιτεκτονική του HCRN. Σχήμα από το [26].

Μία ακόμα σημαντική κατηγορία είναι οι προσεγγίσεις με βάση τους γράφους, graph-based approaches. Οι γράφοι και τα Νευρωνικά Δίκτυα Γράφων (GNNs) έχουν την ικανότητα να αναπαριστούν τις σχέσεις μεταξύ των στοιχείων, να εντοπίζουν τα σημαντικά στοιχεία και να εξάγουν πληροφορία από την δομή των δεδομένων. Το πιο σχετικό με τη δουλειά μας είναι το Situation Hyper-Graph based Video Question Answering [46], το οποίο χρησιμοποιεί διανύσματα υπερ-γράφων (hyper-graphs embeddings). Το ενδιαφέρον με αυτή τη προσέγγιση είναι η εκμάθηση των σχέσεων μεταξύ των οπτικών στοιχείων, όχι με τη χρήση ρητών γράφων σκηνής, αλλά με την χρήση ενδιάμεσης διανυσματικής αναπαράστασης.

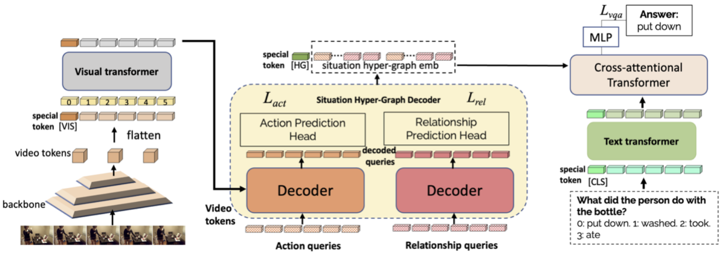


Figure 0.5: Η αρχιτεκτονική του SHG. Σχήμα από το [46].

0.4 Η μέθοδος μας

0.4.1 Επισκόπηση

Στην παρούσα εργασία, αντιμετωπίζουμε το Video QA ως ένα πρόβλημα ταξινόμησης. Συγκεκριμένα, δεδομένου ενός βίντεο V, δηλαδή μίας ακολουθίας από K στιγμιότυπα -frames $V = [f_0, f_1, f_2, ..., f_i, ..., f_k]$ και μίας ερώτησης, q, ο στόχος μας είναι να προβλέψουμε την σωστή απάντηση, a^*, από το σύνολο των πιθανών απαντήσεων, A. Δηλαδή, ορίζουμε ένα dataset, $X = (u_i, q_i, q_i)_{i-1}^{N}$, που αποτελείται από N βίντεο, όπου u_i η οπτική είσοδος από τα στιγμιότυπα του βίντεο, $q_i \epsilon Q$ η ερώτηση και $a_i \epsilon A$ η σωστή (ground truth) απάντηση. Ο στόχος μας λοιπόν είναι να μάθουμε την συνάρτηση $f : Q \times X \to A$, που προβλέπει την κατανομή πιθανοτήτων $P(A)$ των πιθανών απαντήσεων.

Με τη μέθοδό μας εισάγουμε ένα ακόμα modality, τους χωροχρονικούς γράφους σκηνής, με σκοπό την πιο δομημένη και πυκνή αναπαράσταση του περιεχομένου του βίντεο. Για κάθε στιγμιότυπο του βίντεο, f_t, το αναπαριστούμε ως ένα γράφο g_t, ο οποίος αποτελείται από τα αντικείμενα του στιγμιότυπου ως κόμβους και τις μεταξύ τους σχέσεις ως ακμές. Ολόκληρο το βίντεο αναπαριστάται ως ένα σύνολο γράφων $G = [g_1, ..., g_T]$. Στη συνέχεια εξάγουμε διανύσματα χαρακτηριστικών (embeddings) για κάθε γράφο (graph embeddings), ώστε να χρησιμοποιηθούν μαζί με τα διανύσματα χαρακτηριστικών της ερώτησης (question embeddings), για την πρόβλεψη της απάντησης.

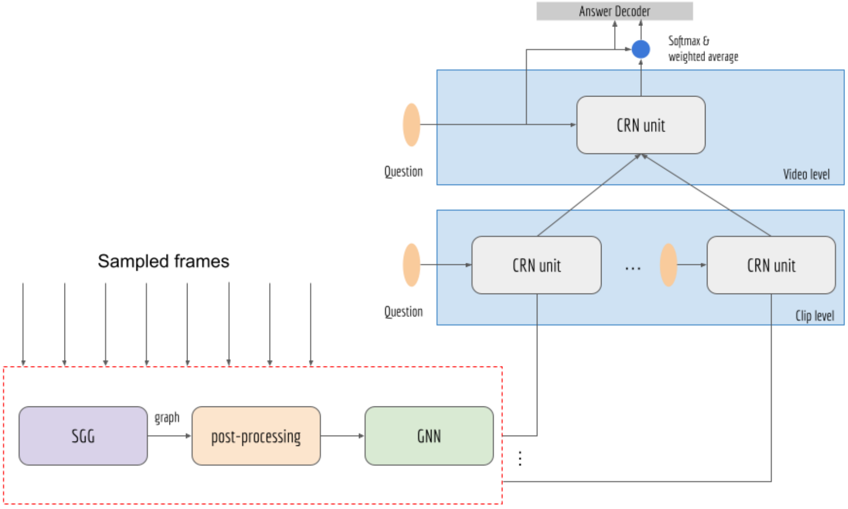


Figure 0.6: Η αρχιτεκτονική της μεθόδου μας. Ξεκινάμε με την επιλογή frames, παράγουμε τους γράφους σκηνής κάθε επιλεγμένο frame και εξάγουμε τα χαρακτηριστικά τους. Στη συνέχεια, επεξεργαζόμαστε τα χαρακτηριστικά αυτά με τη χρήση ενός ιεραρχικού μοντέλου που λειτουργεί σε δύο επίπεδα, το επίπεδο του σύντομου βίντεο κλιπ και το επίπεδο ολόκληρου του βίντεο. Τέλος, το μοντέλο μας προβλέπει την απάντηση στην ερώτηση από το σύνολο των πιθανών απαντήσεων.

Η μέθοδός μας, λοιπόν, ξεκινάει με την επιλογή (sampling) κάποιων frames. Συγκεκριμένα, χωρίζουμε το βίντεο σε 5 βίντεο κλιπ ίσης διάρκειας, ενώ στη συνέχεια επιλέγουμε τυχαία 3 frames από το καθένα. Στη συνέχεια, εξάγουμε το γράφο σκηνής g_i για κάθε frame f_i. Οι γράφοι αυτοί διέρχονται από ένα GNN για να αποτυπωθεί η τοπολογία και η δομή του κάθε γράφου, g_i, σε ένα διάνυσμα χαρακτηριστικών, eg_i. Τα διανύσματα αυτά συνδυάζονται με ένα διάνυσμα χαρακτηριστικών για την ερώτηση, eq_i, και με τη χρήση ενός ιεραρχικού μοντέλου, προβλέπουμε την απάντηση, a_i.

Βασισμένοι στον επαναχρησιμοποιούμενο νευρώνα CRN [26], εισάγουμε ένα νευρώνα προσοχής για τα graph embeddings, με βάση την ερώτηση. Χρησιμοποιούμε αυτούς τους νευρώνες για να χτίσουμε μία πιο βαθιά αρχιτεκτονική, με βάση την ιεραρχική δομή των βίντεο, όπως βλέπουμε και στο Σχήμα 0.6. Συγκεκριμένα, χρησιμοποιούμε δύο επίπεδα, το επίπεδο του σύντομου βίντεο κλιπ και το επίπεδο ολόκληρου του βίντεο. Σε κάθε επίπεδο ιεραρχίας, χρησιμοποιούμε ένα στάδιο CRN νευρώνων, οι οποίοι είναι εξαρτώμενοι (conditioned) στην ερώτηση. Από όσο γνωρίζουμε, αυτή είναι η πρώτη φορά που χρησιμοποιούνται ρητοί γράφοι σκηνής σε συνδυασμό με ιεραρχική δομή για το Video QA.

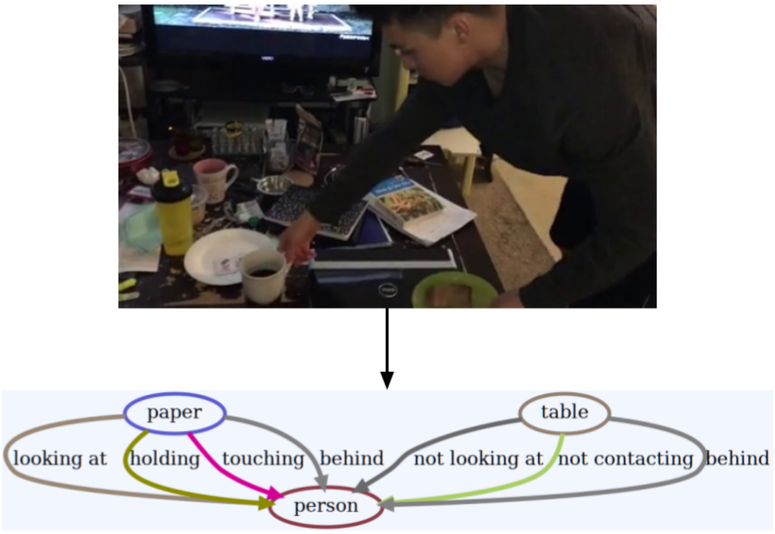


Figure 0.7: Η δημιουργία των γράφων σκηνής από τα frames του βίντεο.

0.4.2 Εξαγωγή Χαρακτηριστικων

Στο Video Question Answering, ένα sample αποτελείται από μία ερώτηση και ένα βίντεο. Για την επεξεργασία του βίντεο με τα baseline μοντέλα μας, εξάγουμε διανύσματα οπτικών χαρακτηριστικών (appearance features) και διανύσματα κίνησης (motion features) με την χρήση προεκπαιδευμένων μοντέλων CNN, και ειδικότερα του ResNeXt-101 [17] για την κίνηση και του ResNet-101 [18] για τα οπτικά χαρακτηριστικά. Για την αναπαράσταση της ερώτησης, περνάμε την ερώτηση από το BERT [10] και χρησιμοποιούμε το CLS token της εξόδου του.

0.4.3 Δημιουργία Γράφων Σκηνής

Σε συνέχεια της επιλογής συγκεκριμένων frames από το βίντεο, δημιουργούμε τους γράφους σκηνής χρησιμοποιώντας ένα προεκπαιδευμένο μοντέλο Scene Graph Generation (SGG). Η διαδικασία αυτή περιλαμβάνει την ανίχνευση αντικειμένων σε μία εικόνα και στη συνέχεια την εύρεση των μεταξύ τους σχέσεων με τη μορφή τριπλετών πχ ” “ ” άνδρας κρατάει ποτήρι” , $< man - holding - glass >$,ή ”πιάτο πάνω σε τραπέζι”, $< dish - ontopof - table >$. Η διαδικασία περιγράφεται στο Σχήμα 0.7.

Στην εργασία μας χρησιμοποιούμε το προεκπαιδευμένο μοντέλο MOTIFS [56] για την δημιουργία των γράφων σκηνής, αλλά μπορούμε να χρησιμοποιήσουμε οποιοδήποτε άλλο SGG μοντέλο στη θέση του. Συγκεκριμένα, επιλέγουμε μία εξελιγμένη μορφή του MOTIFS, με τη χρήση της μεθόδου του [39] για την ελαχιστοποίηση του bias. Το μοντέλο αυτό, με την είσοδο μίας εικόνας εξάγει εκατοντάδες τριπλέτες αντικειμένων και των μεταξύ τους σχέσεων, μαζί με το αντίστοιχο ποσοστό σιγουριάς. Από αυτές, επιλέγουμε τις πιο σίγουρες τριπλέτες, καταλήγοντας σε λιγότερες από 50 ανά frame. Με αυτές τις τριπλέτες δημιουργούμε ένα γράφο, g_i, για κάθε frame, f_i, όπως φαίνεται στο Σχήμα 0.7.

0.4.4 Graph Neural Networks (GNNs)

Για τον ορισμό του γράφου g_i χρησιμοποιούμε τα αντικείμενα ως κόμβους (nodes) και τις μεταξύ τους σχέσεις ως ακμές (edges). Κάθε αντικείμενο ή σχέση αναπαρίσταται από ένα διάνυσμα χαρακτηριστικών, το οποίο είναι το 1-hot vector της κατηγορίας τους, δηλαδή ένα διάνυσμα μήκους όσες και οι κατηγορίες με την τιμή 1 στην θέση της κατηγορίας τους και 0 αλλού.

Οι γράφοι αυτοί στη συνέχεια επεξεργάζονται από ένα GNN, το οποίο μαθαίνει την αναπαράσταση των κόμβων και των ακμών του γράφου. Έτσι, το GNN παράγει ιδιαίτερα πληροφοριακά χαρακτηριστικά για κάθε γράφο,τα οποία και επιτρέπουν την κατανόηση της σκηνής και του βίντεο. Ο σκοπός των GNN είναι να εξάγουν χαρακτηριστικά χαμηλών διαστάσεων, τα οποία συνοψίζουν την δομή και την πληροφορία του γράφου. Αυτό το καταφέρνουν με μία μορφή μετάδοσης μηνυμάτων (message passing), όπου κάθε κόμβος ανανεώνει την αναπαράστασή του με βάση την αναπαράσταση των γειτόνων του.

Στο πλαίσιο της εργασίας μελετήθηκαν πολλές αρχιτεκτονικές GNN, όπως τα Graph Attention Networks [48], Graph Isomorphism Networks [22], αλλά και state-of-the-art μέθοδοι κωδικοποίησης σκηνών, όπως το SCENE [31]. Σε κάθε μία από αυτές τις κατηγορίες αρχιτεκτονικών επιλέχθηκε μοντέλο που επεξεργάζεται τόσο τους κόμβους όσο και τις ακμές του γράφου, καθώς οι γράφοι μας περιλαμβάνουν σημαντική πληροφορία και στα δύο.

0.4.5 Ιεραρχικό Μοντέλο & Απάντηση Ερώτησης

Το τελευταίο στάδιο της μεθόδου μας περιλαμβάνει μία ιεραρχική αρχιτεκτονική για την πρόβλεψη της απάντησης. Συγκεκριμένα, χρησιμοποιούμε δύο επίπεδα, ένα για το σύντομο βίντεο κλιπ και ένα για το ολόκληρο βίντεο. Κάθε ένα από αυτά τα επίπεδα αποτελείται από ένα στάδιο CRN νευρώνων, οι οποίοι είναι εξαρτώμενοι (conditioned) στην ερώτηση. Οι CRN νευρώνες είναι ένας τύπος νευρώνων προσοχής, οι οποίοι είναι σε θέση να εστιάσουν σε συγκεκριμένα στοιχεία της εισόδου, με βάση την ερώτηση. Η ιεαραρχική αρχιτεκτονική μας επιτρέπει να εξάγουμε πιο πλούσια χαρακτηριστικά για το βίντεο, καθώς αυτό επεξεργάζεται σε δύο επιμέρους επίπεδα. Όπως βλέπουμε στο Σχήμα 0.6, το χαμηλότερο επίπεδο, clip level, επεξεργάζεται τα graph embeddings με βάση την ερώτηση, εξάγοντας ένα διάνυσμα χαρακτηριστικών για κάθε βίντεο κλιπ. Στο επόμενο επίπεδο, video level, τα διανύσματα των βίντεο κλιπ επεξεργάζονται από έναν νευρώνα CRN, ο οποίος εξάγει ένα συνολικό διάνυσμα χαρακτηριστικών για το ολόκληρο βίντεο (video-graph embedding, ev_i). Το διάνυσμα αυτό συνδυάζεται με το διάνυσμα της ερώτησης eq_i και με τη χρήση ενός ταξινομητή, προβλέπουμε την απάντηση a_i.

0.4.6 Διαδικασία Εκπαίδευσης

Η διαδικασία εκπαίδευσης του μοντέλου μας περιλαμβάνει την χρήση προ-εξαγμένων γράφων σκηνής. Το μοντέλο μας σειριακά επεξεργάζεται τμήματα (batches) δεδομένων από το dataset, όπου κάθε batch αποτελείται από τους γράφους των επιλεγμένων frames, μία ερώτηση και την αντίστοιχη απάντηση. Το GNN επεξεργάζεται τους γράφους και εξάγει τα graph embeddings, τα οποία με τη σειρά τους εισέρχονται στο ιεραρχικό μας μοντέλο για την πρόβλεψη της απάντησης.

Το optimization πραγματοποιείται με χρήση του CrossEntropyLoss και του AdamW optimizer, ενώ οι μετρικές accuracy και loss καταγράφονται στην πλατφόρμα ανάλυσης πειραμάτων Weights and Biases [2]. Μεταξύ δύο εποχών εκπαίδευσης (train epochs) περιλαμβάνεται μία διαδικασία επαλήθευσης (validation). Η επιλογή του καλύτερου μοντέλου γίνεται με βάση την μετρική accuracy στο βήμα validation.

0.4.7 Προκλήσεις και Περιορισμοί

Η απόδοση της μεθόδου μας εξαρτάται από την ποιότητα των γράφων σκηνής, την ανίχνευση αντικειμένων και την εξαγωγή των σχέσεων μεταξύ τους. Το μοντέλο SGG που επιλέξαμε, αποτελεί ένα παλαιότερο μοντέλο, το οποίο είναι εκπαιδευμένο σε άλλο σύνολο δεδομένων. Μπορούμε εύκολα να λύσουμε αυτό το πρόβλημα, χρησιμοποιώντας ένα state-of-the-art μοντέλο με ελάχιστες αλλαγές στη μέθοδό μας, ενώ η εφαρμογή fine-tuning στο σύνολο δεδομένων μας αναμένεται να ενισχύσει σημαντικά την απόδοσή μας.

Το σύνολο δεδομένων που χρησιμοποιούμε για να αξιολογήσουμε τη μέθοδό μας, το Action Genome Question Answering αποτελείται από τεράστιο αριθμό δεδομένων, περίπου 3 εκατομμύρια ερωτήσεις σε 9.5 χιλιάδες βίντεο. Αυτό το μέγεθος δεδομένων απαιτεί μεγάλη υπολογιστική ισχύ, καθώς και μεγάλη χωρητικότητα μνήμης. Για να αντιμετωπίσουμε αυτό το πρόβλημα, χρησιμοποιούμε μικρότερα υποσύνολα δεδομένων, διατηρώντας τις ίδιες κατανομές με το αρχικό dataset. Έτσι, μπορούμε να εκπαιδεύσουμε το μοντέλο μας σε μικρότερη κλίμακα, ενώ ταυτόχρονα διατηρούμε την αναπαραστατική ικανότητα του.

Είναι σημαντικό να αναφέρουμε την ανάγκη για περαιτέρω μελέτη της ικανότητας γενίκευσης της μεθόδου μας. Η μέθοδός μας υποθέτει ότι οι ερωτήσεις μπορούν να απαντηθούν μόνο με τη χρήση γράφων. Στο σύνολο δεδομένων που χρησιμοποιούμε, AGQA, οι ερωτήσεις εξάγονται από τους γράφους σκηνής, οπότε η υπόθεση αυτή είναι λογική. Ωστόσο, σε πραγματικά σενάρια, οι ερωτήσεις μπορεί να απαιτούν πληροφορίες που δεν υπάρχουν στους γράφους. Για παράδειγμα, η μέθοδός μας δεν μπορεί να απαντήσει σε ερωτήσεις όπως ”Τι χρώμα είναι το ρούχο του άνδρα;” ή ”Πόσα ποτήρια με βυσσινάδα υπάρχουν στο τραπέζι;”.

0.5 Πειράματα & Αποτελέσματα

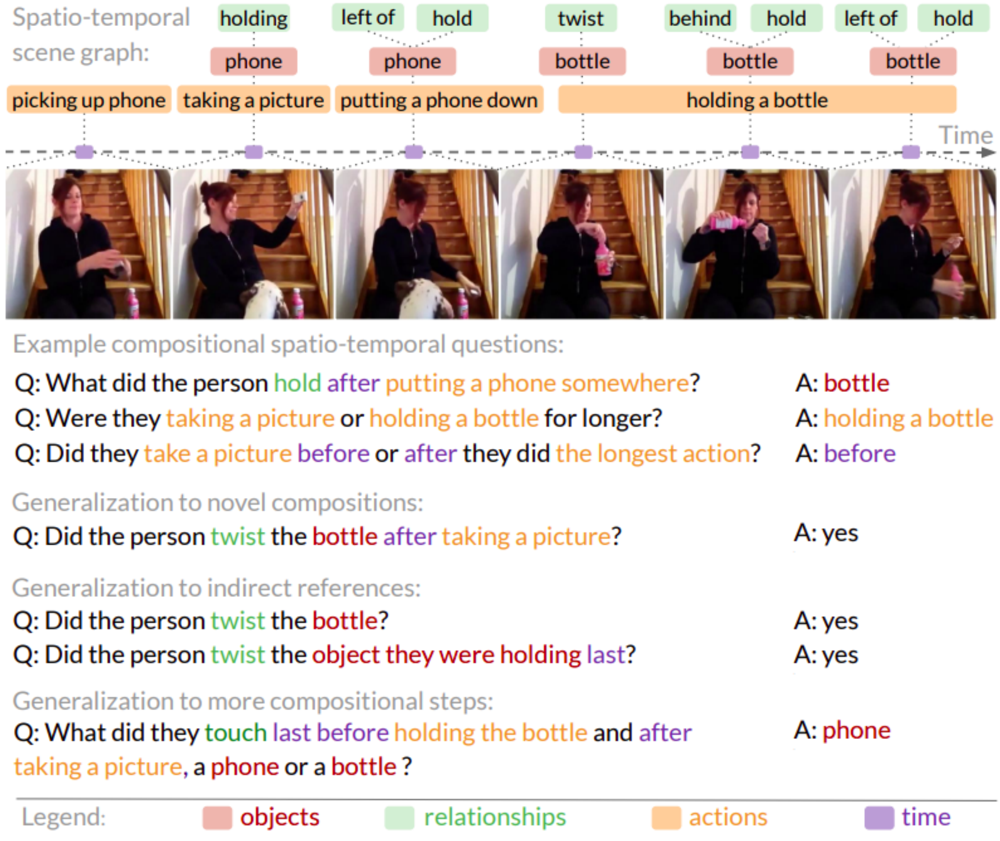
Σε αυτή την υποενότητα θα παρουσιάσουμε τα πειράματα που πραγματοποιήσαμε για την αξιολόγηση της μεθόδου μας. Η αξιολόγηση αυτή πραγματοποιήθηκε στο σύνολο δεδομένων Action Genome Question Answering (AGQA) [14], το πρώτο dataset μεγάλης κλίμακας (large-scale) για το Video Question Answering. με επισημειωμένους γράφους σκηνής. Μετά την παρουσίαση του dataset, περιγράφουμε τις λεπτομέρειες της υλοποίησής μας, προχωρώντας με τις αρχιτεκτονικές των baseline μοντέλων που αναπτύξαμε για την συγκριτική αξιολόγηση (benchmarking) της μεθόδου μας. Τα baseline μοντέλα μας εισάγονται με σκοπό την σταδιακή προσθήκη πολυπλοκότητας μέσα από νέα modalities. Συγκεκριμένα, εισάγουμε:

- **Language Bias Baseline**
- **Language-Vision Baseline**
- **Language & Scene Graphs Baseline**

Βασιζόμενοι σε αυτά τα baseline μοντέλα, παρουσιάζουμε την αρχιτεκτονική της μεθόδου μας, η οποία περιλαμβάνει μία ιεραρχική αρχιτεκτονική με ικανότητα χρονικής μοντελοποίησης, προχωρώντας στην αξιολόγηση της. Πραγματοποιούμε συγκριτική ανάλυση της μεθόδου μας απέναντι στις state-of-the-art μεθόδους του πεδίου, παρουσιάζοντας τα αποτελέσματα των πειραμάτων μας. Τέλος, παρουσιάζουμε μία σύνοψη των πειραματικών μας αποτελεσμάτων.

0.5.1 Action Genome Question Answering

Το σύνολο δεδομένων Action Genome Question Answering (AGQA) [14] αποτελεί ένα Video Question Answering dataset μεγάλης κλίμακας, με σχεδόν 4 εκατομμύρια ερωτήσεις σε 9.5 χιλιάδες βίντεο. Το AGQA επεκτείνει το σύνολο δεδομένων Charades [55], το οποίο περιλαμβλάνει βίντεο από καθημερινές δραστηριότητες σε εσωτερικούς χώρους, προσθέτοντας επισημειωμένους γράφους σκηνής και ερωτήσεις βασισμένες σε αυτούς. Οι ερωτήσεις, λοιπόν, είναι επικεντρωμένες στα αντικείμενα, τις μεταξύ τους σχέσεις και τις ενέργειες που απεικονίζονται στο βίντεο. Παράδειγμα ερωτήσεων του AGQA μπορούμε να δούμε στο Σχήμα 0.8.



Example compositional spatio-temporal questions:

Q: What did the person hold after putting a phone somewhere? A: bottle

Q: Were they taking a picture or holding a bottle for longer? A: holding a bottle

Q: Did they take a picture before or after they did the longest action? A: before

Generalization to novel compositions:

Q: Did the person twist the bottle after taking a picture? A: yes

Generalization to indirect references:

Q: Did the person twist the bottle? A: yes

Q: Did the person twist the object they were holding last? A: yes

Generalization to more compositional steps:

Q: What did they touch last before holding the bottle and after taking a picture, a phone or a bottle ? A: phone

Figure 0.8: Παραδείγματα ερωτήσεων από το σύνολο δεδομένων AGQA. Σχήμα από το [14].

0.5.2 Λεπτομέρειες Υλοποίησης

Για την υλοποίηση των πειραμάτων μας χρησμοποιήσαμε τη βιβλιοθήκη PyTorch [33] για την ανάπτυξη και εκπαίδευση των μοντέλων μας, τη βιβλιοθήκη Deep Graph Library (DGL) [49] για τη μορφοποίηση και επεξεργασία των γράφων, καθώς και τα GNNs.

Για την εκπαίδευση των μοντέλων μας χρησιμοποιήσαμε δύο μηχανήματα, εξοπλισμένα με κάρτες γραφικών NVIDIA RTX 3090 και GTX 1080 Ti αντίστοιχα. Αυτή η υποδομή επέτρεψε την παραλληλοποίηση της εκπαίδευσης των μοντέλων μας και την αποδοτική εξαγωγή δεδομένων.

Για την διαχείριση των υπολογιστικών πόρων, δημιουργήσαμε τρία υποσύνολα του συνόλου δεδομένων AGQA, το tiny, το small και το medium, οπως φαίνεται στον

Πίνακα 0.1. Το tiny αποτελείται από 10k δείγματα εκπαίδευσης (train samples) και 2k δείγματα test (test samples) και το small αποτελείται από 100k train samples και 20k test samples. Και τα δύο υποσύνολα διατηρούν τις ίδιες κατανομές με το αρχικό dataset, καθώς επιλέχθηκαν με τυχαίο τρόπο από το αρχικό σύνολο train, ενώ ακόμα διασφαλίστηκε η χρήση διαφορετικών βίντεο στα υποσύνολα train και test, ώστε να μην υπάρχει διαρροή δεδομένων.

Υποσύνολα Δεδομένων

Split	Train Size	Test Size
tiny	10.000	2.000
small	100.000	20.000

Table 0.1: Υποσύνολα δεδομένων που δημιουργήσαμε στο AGQA για την διαχείριση του όγκου του.

Για την υλοποίηση των πειραμάτων μας, χρειάστηκε η εξαγωγή των frames κάθε βίντεο, κάτι που πραγματοποιήθηκε με τη χρήση του FFmpeg [44]. και ένα παρεχόμενο πρόγραμμα από τους συγγραφείς του AGQA.

0.5.3 Baseline Μοντέλα

0.5.3.1 Language Bias Baseline

Στο πρώτο μας πείραμα, εστιάσαμε στην αξιολόγηση της ικανότητας των μοντέλων μας να απαντήσουν σε ερωτήσεις μόνο με βάση το κείμενο της ερώτησης, χωρίς κάποια πληροφορία για το βίντεο, όπως βλέπουμε και στο Πίνακα 0.2. Χρησιμοποιήσαμε το CLS token από το μοντέλο BERT, το οποίο εισέρχεται σε ένα μοντέλο MLP για την πρόβλεψη της απάντησης, όπως φαίνεται στο Σχήμα 0.9. Εκπαιδεύσαμε το μοντέλο μας στα υποσύνολα tiny και small, όπου και παρουσιάζουμε τα αποτελέσματα μας στο Πίνακα 0.3. Το σκορ του μοντέλου μας στο υποσύνολο tiny ανέρχεται στο 21.5%, ενώ στο υποσύνολο small ανέρχεται στο 34.1%. Με αυτό το πείραμα, μπορούμε να εκτιμήσουμε τις συσχετίσεις μεταξύ των ερωτήσεων και των απαντήσεων, χωρίς την παρουσία πληροφορίας για το βίντεο. Μάλιστα, state-of-the-art προσεγγίσεις μπορούν να εκμεταλλευτούν το γλωσσικό bias και να επιτύχουν μέχρι και 47% accuracy βλέποντας μόνο την ερώτηση.

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-

Table 0.2: Modalities & Ικανότητες του Language Bias Baseline.

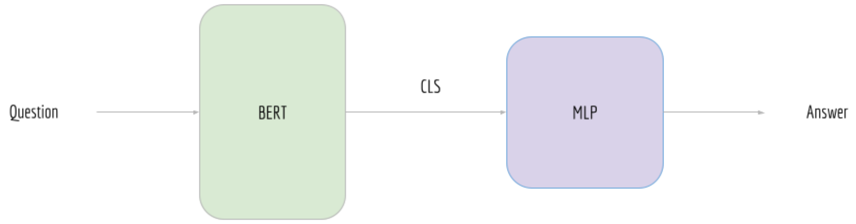


Figure 0.9: Αρχιτεκτονική του Language Bias Baseline με σκοπό την απάντηση ερωτήσεων μόνο με βάση την ερώτηση.

Υποσύνολα	**Accuracy**
tiny	21.5%
small	34.1%

Table 0.3: Αποτελέσματα στη μετρική accuracy του Language Bias Baseline στα υποσύνολα του AGQA.

0.5.3.2 Language-Vision Baseline

Στο δεύτερο μας πείραμα, επεκτείναμε το non-temporal μοντέλο μας προσθέτοντας την εικόνα του βίντεο, χωρίς τη χρήση των γράφων σκηνής, όπως φαίνεται στο Σχήμα 0.10. Ερευνήσαμε το πως η προσθήκη appearance features των frames βελτιώνει την απόδοση του μοντέλου μας, με τα modalities που φαίνονται στον Πίνακα 0.4. Χρησιμοποιήσαμε appearance features από ένα ResNet-101 και γλωσσικά features, όπως και προηγουμένως, το CLS token του BERT. Για να αντισταθμίσουμε την έλλειψη χρονικής πληροφορίας, χρησιμοποιήσαμε τον μέσο όρο των appearance features των frames για την πρόβλεψη της απάντησης.

Model	**Language**	**Vision**	**Scene Graphs**	**Temporal Processing**
Lang_MLP	✓	-	-	-
Lang_Vision	✓	✓	-	-

Table 0.4: Modalities & Ικανότητες του Language-Vision Baseline.

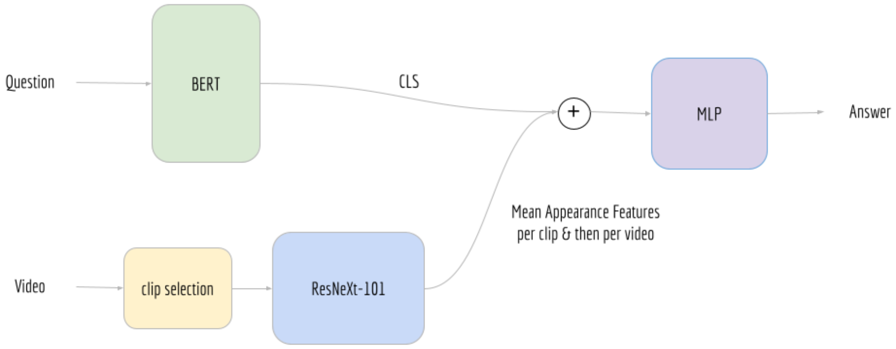


Figure 0.10: Αρχιτεκτονική του Language-Vision Baseline με σκοπό την απάντηση ερωτήσεων με βάση την ερώτηση και το βίντεο με non-temporal μοντέλο.

Τα αποτελέσματα μας στο Πίνακα 0.5 δείχνουν ότι η προσθήκη των appearance features βελτιώνει την απόδοση του μοντέλου μας, με το σκορ του να ανέρχεται στο 24.3% στο υποσύνολο tiny και στο 38.2%. Μπορούμε να μετρήσουμε την ποιοτική βελτίωση του μοντέλου μας, όπως φαίνεται στον Πίνακα 0.6, όπου παρουσιάζονται τα αποτελέσματα σε διάφορα είδη ερωτήσεων. Παρατηρούμε ότι το μοντέλο μας είναι καλύτερο στις ερωτήσεις τύπου exists, obj-act και rel-act, ενώ χρειάζεται βελτίωση στις ερωτήσεις που απαιτούν χρονική πληροφορία, όπως οι ερωτήσεις τύπου duration-comparison και action-recognition.

Experiments	lang	lang+vid
tiny	21.5%	26.1%
small	24.1%	25.8%

Table 0.5: Αποτελέσματα στη μετρική accuracy του Language-Vision Baseline στα υποσύνολα του AGQA.

0.5.3.3 Ανώτατο όριο non-temporal baseline

Με αυτό το πείραμα, εξετάζουμε το ανώτατο όριο του non-temporal μοντέλου προσθέτοντας τους γράφους σκηνής στην αρχιτεκτονική του μοντέλου μας, κάνοντας παράλληλα ένα Proof of Concept (PoC) της μεθόδου μας όπως αναγράφεται και στον Πίνακα 0.7. Με τη χρήση των γράφων σκηνής θέλουμε να δώσουμε στο μοντέλο μας πιο δομημένη πληροφορία για το περιεχόμενο του βίντεο, όπως αντικείμενα, ενέργειες και σχέσεις μεταξύ τους. Χρησιμοποιούμε τους επισημειωμένους γράφους σκηνής

question-type	**accuracy**
exists	32.8%
obj-rel	18.5%
obj-act	24%
sequencing	24.62%
duration-comparison	12%
rel-act	22%
action-recognition	0%

Table 0.6: Αποτελέσματα του Language-Vision Baseline στα διάφορα είδη ερωτήσεων του AGQA.

που παρέχονται από το AGQA, όπως φαίνεται στο Σχήμα 0.11. Για κάθε επιλεγμένο frame δηλαδή, έχουμε τον αντίστοιχο γράφο σκηνής, ο οποίος αποτελεί είσοδο στο GNN μοντέλο μας. Η αρχιτεκτονική GNN που χρησιμοποιούμε εδώ είναι το Graph Attention Network με edge features. Κάθε frame, λοιπόν, μετατρέπεται σε γράφο σκηνής και κωδικοποιείται με το GNN σε ένα graph embedding. Για την πρόβλεψη της απάντησης, συνδυάζουμε τα embeddings όλων των γράφων, βρίσκοντας τον μέσο όρο τους σε ένα video-graph embedding, ώστε να αντισταθμίσουμε την έλλειψη χρονικής πληροφορίας. Το video graph embedding στη συνέχεια συνδυάζεται με την ερώτηση και εισάγονται σε ένα MLP για την πρόβλεψη της απάντησης.

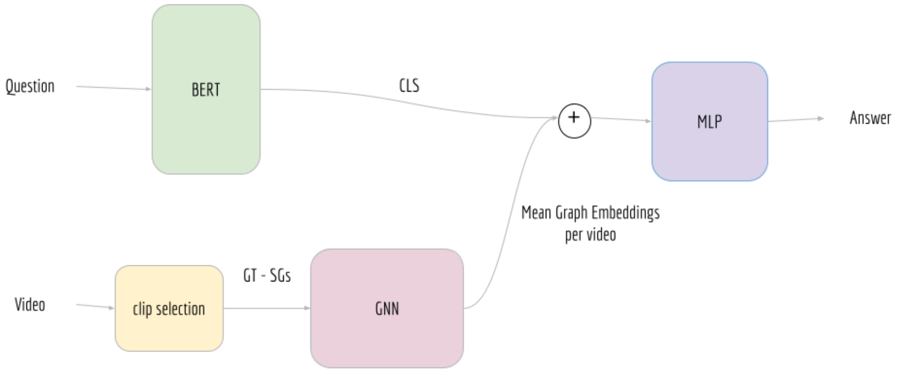


Figure 0.11: Αρχιτεκτονική του Proof of Concept baseline με χρήση των επισημειωμένων γράφων σκηνής (GT - SGs) και ενός non-temporal μοντέλου MLP.

Τα αποτελέσματα μας στον Πίνακα 0.8 δείχνουν ότι η προσθήκη των γράφων σκηνής βελτιώνει σημαντικά την απόδοση του μοντέλου μας, με το σκορ του να ανέρχεται

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-
Vid_Lang_MLP	✓	✓	-	-
PoC	✓	✓	✓	-

Table 0.7: Modalities & Ικανότητες του Proof of Concept Baseline.

στο 49.1% στο υποσύνολο tiny. Μπορούμε να μετρήσουμε την ποιοτική βελτίωση του μοντέλου μας, όπως φαίνεται στον Πίνακα 0.9, όπου παρουσιάζονται τα αποτελέσματα σε διάφορα είδη ερωτήσεων, τα οποία είναι βελτιωμένα σχεδόν σε όλες τις κατηγορίες. Παρατηρούμε ότι το μοντέλο μας είναι καλύτερο στις ερωτήσεις τύπου exists, obj-act και obj-rel, ενώ χρειάζεται βελτίωση στις ερωτήσεις που απαιτούν χρονική πληροφορία, όπως οι ερωτήσεις τύπου duration-comparison και action-recognition, όπως και περιμέναμε. Συμπεραίνουμε, λοιπόν, ότι παρά την έλλειψη χρονικής πληροφορίας, η προσθήκη του video-graph embedding μπορεί να δώσει μία " “ "χρονική" διάσταση, επιτρέποντας στο μοντέλο να καταλάβει το περιεχόμενο του βίντεο. Ακόμη, η προσθήκη των γράφων μας δείχνει ότι η μετατροπή των frames σε πιο δομημένη αναπαράσταση μπορεί να βελτιώσει την απόδοση του μοντέλου μας.

Experiments	Lang_MLP	Vid_Lang_MLP	PoC
tiny	21.5%	26.1%	49.1%

Table 0.8: Πειραματικά αποτελέσματα με την προσθήκη επισημειωμένων γράφων σκηνής στο Proof of Concept baseline μας.

question type	accuracy
superlative	22.6%
obj - rel	27.72%
exists	32.8%
obj-act	36%
sequencing	30.8%
duration-comparison	6%
rel-act	0.3%
action-recognition	0%

Table 0.9: Αποτελέσματα του non-temporal Proof of Concept baseline μας ανά κατηγορία ερώτησης.

0.5.4 Η τελική μας προσέγγιση

0.5.4.1 Non-Temporal Μοντέλο

Αρχικά, πρώτη μας δοκιμή ήταν η εξέλιξη του PoC μοντέλου μας με την χρήση προβλεπόμενων γράφων σκηνής αντί για τους επισημειωμένους, όπως μπορούμε να δούμε και στο Σχήμα 0.12.

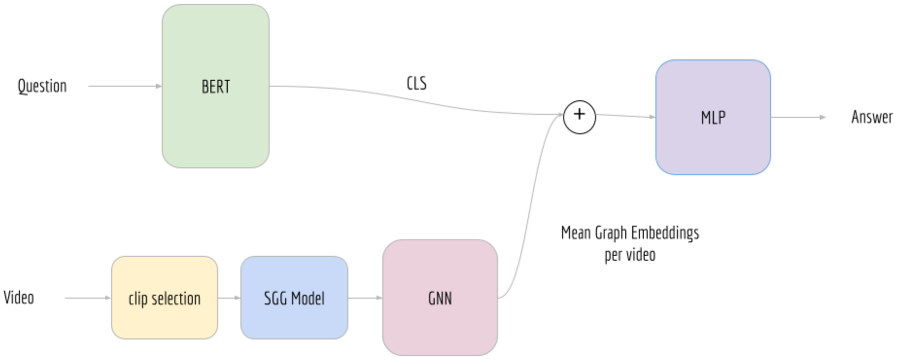


Figure 0.12: Αρχιτεκτονική της non-temporal προσέγγισης μας με τη χρήση γράφων σκηνής που προβλέψαμε.

Όπως παρατηρούμε από τα αποτελέσματα του Πίνακα 0.10, η απόδοση του μοντέλου μας είναι αρκετά υψηλότερη από τα πρώτα δύο baselines, υστερεί όμως σίγουρα σε σχέση με το Proof of Concept baseline. Αυτό ήταν αναμενόμενο, καθώς το μοντέλο SGG μας δεν είναι ρυθμισμένο για το AGQA, οπότε η ποιότητα των γράφων σίγουρα είναι χειρότερη από τους επισημειωμένους.

Experiments	Lang_MLP	Vid_Lang_MLP	PoC	SG_MLP
tiny	21.5%	26.1%	49.1%	31.6%

Table 0.10: Πειραματικά αποτελέσματα με την προσθήκη γράφων σκηνής στο Proof of Concept baseline μας.

0.5.4.2 Temporal Μοντέλο

Στη συνέχεια, αντλήσαμε έμπνευση από το HCRN και καταλήξαμε στην τελική μας αρχιτεκτονική, που φαίνεται στο Σχήμα 0.6. Παρατηρήσαμε ότι το μοντέλο αυτό απαιτεί περισσότερες εποχές για σύγκλιση σε σχέση με τα προηγούμενα, καθώς είναι πιο σύνθετη και βαθιά αρχιτεκτονκή.

Μπορούμε να δούμε ότι το μοντέλο μας έχει πολύ καλή απόδοση, ξεπερνώντας τα προηγούμενα baseline μοντέλα, όπως φαίνεται στον Πίνακα 0.12. Είναι παρόλαυτά

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-
Vid_Lang_MLP	✓	✓	-	-
PoC	✓	✓	✓	-
SG-MLP	✓	✓	✓	-
SG-HCRNx	✓	✓	✓	✓

Table 0.11: Modalities & Ικανότητες των μοντέλων μας.

λίγο πιο χαμηλά σε συνολικό accuracy από το PoC μοντέλο μας, κάτι που οφείλεται στην ποιότητα των γράφων σκηνής που χρησιμοποιούμε. Μάλιστα, από τον Πίνακα 0.13 μπορούμε να δούμε ότι το μοντέλο μας τα πηγαίνει καλά στην αναγνώριση των αντικειμένων και των αλληλεπιδράσεών τους μέσα στο βίντεο, καθώς έχει υψηλό σκορ στις κατηγορίες obj-act και obj-rel. Ακόμη, όπως φαίνεται στον Πίνακα 0.14, η μέθοδός μας παρουσιάζει αποτελέσματα ανταγωνιστικά με τις state-of-the-art μεθόδους, όπως το PSAC, το HME και το HCRN, ενώ βρίσκεται δεύτερη συνολικά, μετά από το SHG-VQA. Η μέθοδός μας μάλιστα πετυχαίνει κορυφαία αποτελέσματα από όλες τις μεθόδους σε μερικές κατηγορίες ερωτήσεων, όπως το obj-rel και το superlative, δείχνοντας έτσι την ικανότητά του μοντέλου μας να αντιληφθεί τις σχέσεις μεταξύ των αντικειμένων στο βίντεο και να κατανοήσει βαθύτερα το περιεχόμενό του. Το μοντέλο μας, ακόμα, έρχεται δεύτερο στις κατηγορίες κατηγορίες ερωτήσεων, πέρα από το exists, κάτι που ίσως οφείλεται στα διαφορετικά λεξιλόγια των γράφων σκηνής και των επισημειωμένων γράφων, καθώς και τη μειωμένη ικανότητα γενίκευσης του μοντέλου μας.

Experiments	Lang_MLP	Vid_Lang_MLP	PoC	SG_MLP	SG_HCRNx
tiny	21.5%	26.1%	49.1%	31.6%	42.5%

Table 0.12: Πειραματικά αποτελέσματα της μεθόδου μας σε σύγκριση με τα baseline μοντέλα μας.

question type	accuracy
superlative	55.1&
obj - rel	49.8%
exists	53.7%
obj-act	56.3%
sequencing	50.4%
duration-comparison	25.5%
rel-act	40.9%
action-recognition	7.4%

Table 0.13: Αποτελέσματα σε accuracy της μεθόδου μας ανά κατηγορία ερώτησης.

Method	obj-rel	rel-action	obj-action	superlative	sequencing	exists	duration	activity	Overall
PSAC [30]	37.84	49.95	50.00	33.20	49.78	49.94	45.21	4.14	40.18
HME [5]	37.42	49.90	49.97	33.21	49.77	49.96	47.03	5.43	39.89
HCRN [26]	40.33	49.86	49.85	33.55	49.70	50.01	43.84	5.52	42.11
SHG-VQA [46]	46.42	**60.67**	**64.63**	38.83	**62.17**	56.06	**48.15**	**10.12**	**49.20**
SG_HCRNx(ours)	**49.8**	53.7	55.1	**40.9**	50.4	**56.3**	25.5	7.4	42.5

Table 0.14: Σύγκριση της μεθόδου μας με state-of-the-art μεθόδους στο tiny dataset.

0.5.5 Μελέτες

Μέρος της παρούσας εργασίας αποτελούν και κάποιες επιπλέον μελέτες που πραγματοποιήσαμε στο πλαίσιο της προσέγγισής μας. Αρχικά, μελετήσαμε την επίδραση των διαφορετικών modalities στο HCRN, όπου επιβεβαιώσαμε τα πειραματικά αποτελέσματα που αναφέρονται στο [26]. Στη συνέχεια, μελετήσαμε την επίδραση των διαφορετικών αρχιτεκτονικών GNN στην απόδοση των graph embeddings, όπου πειραματιστήκαμε με διαφορετικές προσεγγίσεις, πολλαπλές στρώσεις και διαφορετικούς συνδυασμούς επιπέδων. Τέλος, μελετήσαμε την επίδραση των διαφορετικών σταδίων της ιεραρχικής αρχιτεκτονικής προσθέτοντας ένα στάδιο μέσα σε κάθε επίπεδο και κάνοντας την αρχιτεκτονική πιο βαθιά.

0.5.6 Πειραματικά Συμπεράσματα

Η παρούσα μελέτη εισάγει μία νέα προσέγγιση στο πρόβλημα του Video Question Answering, χρησιμοποιώντας γράφους σκηνής σε συνδυασμό με ιεαραρχική αρχιτεκτονική. Η προσέγγισή μας είναι πρωτότυπη και παρουσιάζει ανταγωνιστικά αποτελέσματα σε σχέση με τις state-of-the-art μεθόδους. Φυσικά, υπάρχουν πολλά πεδία για μελλοντική έρευνα, όπως η βελτίωση της ποιότητας των γράφων σκηνής, η χρήση πιο προηγμένων μοντέλων GNN ή η χρήση διαφορετικών τύπων γράφων.

0.6 Συμπεράσματα

Η παρούσα εργασία ερευνά το visual-relation driven Video QA μέσω των γράφων σκηνής. Παρουσιάζουμε μία νέα προσέγγιση, την οποία αξιολογούμε στο AGQA, εστιάζοντας σε σενάρια πραγματικού κόσμου και καθημερινές δραστηριότητες. Η μεθοδολογία μας στηρίζεται στην υπόθεση ότι οι γράφοι σκηνής παρέχουν ουσιαστικές πληροφορίες για την κατανόηση του περιεχομένου ενός βίντεο, ενώ η ιεραρχική αρχιτεκτονική με νευρώνες προσοχής μας επιτρέπει να απαντήσουμε σε σύνθετα ερωτήματα. Τα πειραματικά μας αποτελέσματα δείχνουν ότι η προσέγγισή μας είναι ανταγωνιστική με τις state-of-the-art μεθόδους, ενώ παρουσιάζει τον αντίκτυπο των διαφορετικών modalities στην ποιότητα των απαντήσεων.

Η υιοθέτηση scene-graph-driven μεθόδων στο Video QA ενισχύει την κατανόηση πολύπλοκου περιεχομένου των βίντεο, αλλά αντιμετωπίζει προκλήσεις, ιδίως όσον αφορά την εξάρτηση από την ποιότητα των εξαγόμενων γράφων σκηνής. Είναι ύψιστης σημασίας ο σωστός εντοπισμός των αντικειμένων και των σχέσεών τους, καθώς τυχόν ανακρίβειες μπορεί να οδηγήσουν σε λανθασμένες απαντήσεις. Επιπλέον, η ικανότητα του συστήματος να γενικεύει σε διάφορους τομείς περιορίζεται από την ποικιλομορφία του περιεχομένου και τις υπολογιστικές απαιτήσεις της επεξεργασίας των βίντεο, κάτι που δυσκολεύει εφαρμογές πραγματικού χρόνου.

Ακόμα, έχουμε καταγράψει μελλοντικά βήματα για καλύτερη απόδοση της μεθόδου μας . Η βελτίωση της ποιότητας των γράφων σκηνής μπορεί να εξασφαλιστεί με την περεταίρω εκπαίδευση ενός μοντέλου SGG στο συγκεκριμένο dataset για την ενίσχυση του accuracy. Μία εκπαίδευση end-to-end θα μπορούσε να βοηθήσει στην εύρεση καλύτερων αναπαραστάσεων των γράφων, ενώ η χρήση neurosymbolic προσεγγίσεων, όπως το Neural State Machine [23] ή σύγχρονων transformers μεγάλης κλίμακας όπως το CLIP-BERT [29] μπορεί να οδηγήσει σε πιο αποτελεσματική μάθηση. Τέλος, η χρήση διαφορετικών γράφων σκηνής, η ενσωμάτωση οπτικών πληροφοριών ή χωρικών πληροφοριών, όπως συντεταγμένες περιοχών ενδιαφέροντως μπορεί να επιτρέψει στο μοντέλο μας να απαντήσει ερωτήσεις που σχετίζονται με οπτική πληροφορία ακόμα καλύτερα.

1 Introduction

1.1 Introduction

We live in the age of digital information, where everything around us is connected to a data source, and everything in our lives is digitally recorded [3]. With our surroundings and daily activities becoming more and more digitalized, our smartphones capturing our everyday routines, and social media platforms documenting everything, from our thoughts to our interactions with our friends, we are constantly generating and consuming digital data. Our lives are surrounded by data sources, like our smartphones and tablets, wearable devices, smart home systems, online shopping and transactions, entertainment platforms and many more.

The volume of data, particularly from videos, has fundamentally changed the way we interact with the world. Video platforms and streaming services offer access to a vast diversity of content, while video data has also revolutionized communication. Videos have shaped education, marketing, and entertainment, democratizing information sharing, and shifting our lives into a more video-centric world. To give a quantitative overview of this, approximately 328.77 million terabytes of data are created each day, while video content is responsible for over half of all global data traffic [11]. However, while data generation grows rapidly, as seen in Figure 1.1 [43], it is becoming more and more difficult to process and understand it, since systems able to understand it have not shown the same progress yet. This is why the integration of Machine Learning (ML) in our lives becomes critical. Machine Learning Systems have the capability to navigate through large datasets, identify patterns and trends that the human mind could not process. Especially in video content, we need systems that are able to understand videos, extract meaningful insights, crucial for human interactions.

In this study, we focus on efficiently understanding and interpreting videos for more accurate Video Question Answering performance. Our approach takes a novel path in this direction, using scene graphs along with a hierarchical conditional approach. Scene graphs offer a structured representation of video content, while providing an insightful overview of visual elements and the relations between them. Using a hierarchical approach allows us to operate at two levels, clip level and video level, capturing spatial relationships, actions and interactions in different contexts. The above lead us to a more salient understanding of the video content.

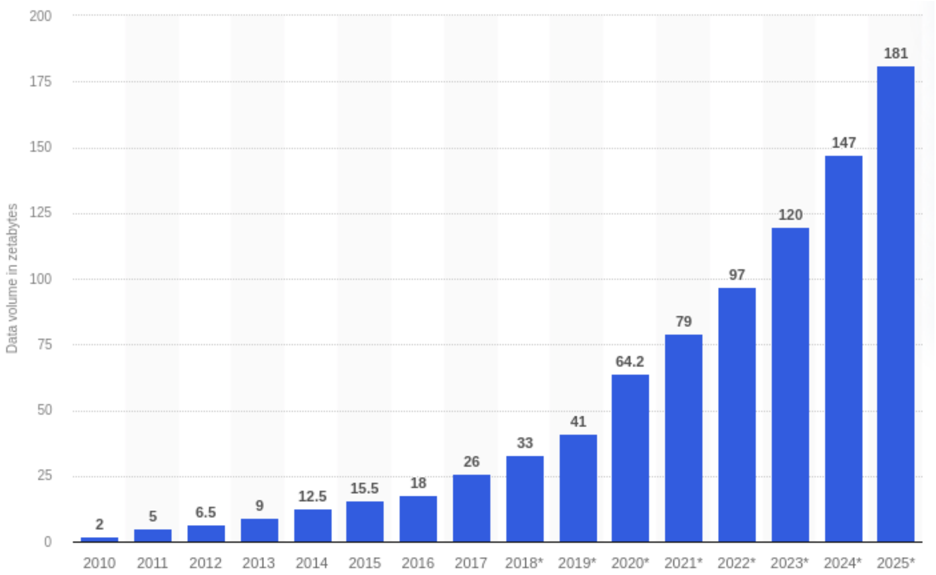
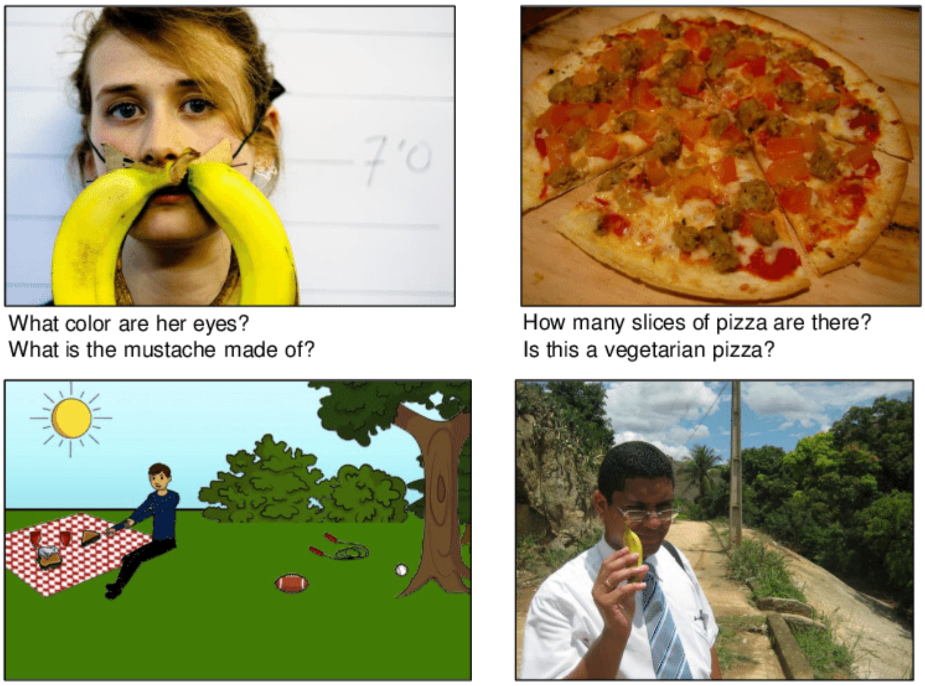


Figure 1.1: Volume of data/information created, captured, copied and consumed from 2010 to 2020, with forecasts from 2021 to 2025 (in zettabytes). Figure from [43].

1.2 Video Question Answering

With the exponential increase in visual content, particularly in videos and images, it is becoming more critical to manage and interpret it. So, the need to interact with digital content, including querying and understanding visual data has become a necessity. Especially since our world works with dynamic multi-modal data, understanding videos is a crucial next step to developing intelligent machines and AI agents. Visual and Video Question Answering fields have emerged to address this need. Being an interesting intersection of computer vision and natural language processing, they allow the user to ask questions about an image or a video and receive accurate answers. So, given a video clip and a language query about it, Video Question Answering aims to accurately respond to that question, grounded on the multi-modal information available.

Before Video Question Answering, Image Question Answering has achieved many advancements, due to the development of deep neural networks. Mirroring real-world scenarios, such as helping the visually impaired, the task is to provide an accurate natural language processing answer based on an image and a natural language question about the image.[1]. The development of Video Question Answering systems began with the need to extract meaningful information from the rapidly growing volume of video content. This task was first introduced in 2015 with the publishing of MovieQA[41], the first Video Question Answering (Video QA) Dataset. The goal of this dataset is to understand complex narratives and information presented in videos and thus push the boundaries of machine learning in interpreting not just visual elements, but also the temporal domain. Video Question Answering involves a Machine Learning system recognizing and understanding various elements within a video, such as objects, actions, scenes, and even human interactions and emotions. As questions are potentially unconstrained, Video QA requires deep modeling capacity to encode and represent crucial visual properties.[26]



What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

Is this person expecting company?
What is just under the tree?

Does it appear to be rainy?
Does this person have 20/20 vision?

Figure 1.2: Examples of free-form, open-ended questions from Visual Question Answering (VQA) Dataset. Figure from [1].

In this short period of time, a variety of Video Question Answering Datatasets have emerged, each to test different aspects of video understanding. Datasets like MSVD-QA [51], MSRVTT-QA [51], ActivityNet-QA [54], and EgoVQA[12], using mostly web videos, focus on description-type questions, where the AI system needs to identify and explain visual content. MovieQA[42], TVQA [28], and TVQA+ [28] use movies and TV show clips, testing the system's capability to understand complex narratives using both the visual and auditory modalities.

Synthetic video datasets, including CLEVRER [53] push AI models logical understanding and spatiotemporal reasoning, while there are knowledge-based datasets, like KnowIT VQA [13] and NEWSKVQA[16] that consist mainly of TV shows and news videos and require the AI system to integrate external knowledge. Finally, there are Video QA Datasets, like TGIF-QA[24], AGQA[14], and NExT-QA[50] that emphasize causal relationships, event understanding, and action sequences.

Q: How does E.T. show his happiness that he is finally returning home?
A: His heart lights up

Q: Why do Joy and Jack get married that first night they meet in Las Vegas?
A: They are both vulnerable and totally drunk

Q: Why does Forrest undertake a three-year marathon?
A: Because he is upset that Jenny left him

Q: How does Patrick start winning Kat over?
A: By getting personal information about her likes and dislikes

Figure 1.3: Examples from the MovieQA dataset. For illustration we show a single frame, however, all these questions/answers are timestamped to a much longer clip in the movie. Figure from [41].

To address these datasets, Video Question Answering approaches have evolved from simple recognition to more complex reasoning. Initially, approaches utilized neural networks to recognize objects and actions within videos to answer simple "what is" questions. In the past years, the field has shifted into a more in-depth analysis of videos, trying to capture complex causal and temporal relationships between objects, actions, and events, focusing not only on the "what" but also on "why", "when", "who", "after what" etc.

Many research efforts have focused on cross-modal interaction, aiming to understand videos under the guidance of questions. General trends in Video QA are deep learning based, often utilizing Transformers [47] and other attention mechanisms cross-modal learning, and external knowledge integration. Deep learning techniques often use Convolutional Neural Networks (CNNs) [27] to encode the visual data in videos, effectively identifying visual elements. The processed visual information is often forwarded to a sequence model, like Recurrent Neural Networks (RNNs) [6] or Long Short-Term Memory (LSTM) [20] models, to capture the temporal dynamics. Transformer models, having revolutionized natural language processing, are also adapted for the Video Question Answering task, use their attention mechanisms to focus on specific parts of the video relevant to the question. Cross-modal algorithms integrate visual and textual data, building correlations between these domains. Knowledge integration models use external knowledge, linking with databases, or knowledge graphs to enrich the information available from the video.

1.2.1 Applications

Video Question Answering has a lot of diverse applications across multiple sectors. Firstly, in the field of education, Video Question Answering can allow students to interact with educational videos through queries, answering their questions and enabling a more personalized learning experience. The entertainment industry can benefit from Video Question Answering by providing users with the ability to learn about scenes or characters in movies and TV shows, or get more personalized and accurate recommendations.

In the security sector, Video Question Answering can help identify events and activities, while enhancing safety. Video Question Answering can provide driver assistance systems with information about road conditions or the surrounding environment while also analyzing experimental videos for research and development across multiple fields, including Physics, Biology, or Engineering.

Video QA systems can also help in dealing with large video databases, and organizing and retrieving videos more efficiently. Video Question Answering systems can also enable efficient extraction of information from long-form video content, like videos from Youtube, documentaries, thus performing information search not only across text sources, or image sources, but also video ones.

Another aspect of Video Question Answering applications lies in assisting visually impaired individuals. This application is particularly impactful in addressing the challenges faced by visually impaired individuals in interpreting and interacting with video content or even real-life data. Video QA can play a vital role in social inclusion by providing these individuals with better access to videos, enabling their independence, and helping them navigate and understand their environment. Finally, Video Question Answering Systems can lead to the development of augmented reality assistants, supporting humans, not just the visually impaired, in daily activities.

1.3 Challenges

The complexity of VQA lies in its multimodal nature, requiring the integration of diverse and very different in nature data forms, specifically video and text. This task is further complicated by the need to understand both spatial elements, like objects and scenes within the video, and temporal dynamics, which involve actions and events unfolding over time.

An accurate Video-Question Answering system should be able to reason on:

- **Action Recognition**
  A critical part of VQA is action recognition, where the goal is to identify and categorize actions within video sequences. This task faces challenges such as varying camera angles, occlusions, and the diversity in how actions are performed.

- **Temporal Localization**
  Temporal localization is another vital aspect, focusing on pinpointing the specific time frames in a video that are relevant to the question. This task becomes particularly challenging with lengthy videos or those featuring subtle or overlapping actions.

- **Natural Language Processing (NLP)**
  In VQA, Natural Language Processing (NLP) plays a crucial role in interpreting the questions and processing the language-based information.

- **Contextual Reasoning**
  Lastly, contextual reasoning is fundamental to VQA, as it involves drawing inferences and understanding the broader context of both the video content and the question posed. This requires an advanced level of AI that can not only recognize elements within a video but also understand their interrelations and the overall narrative or message of the video concerning the question asked.

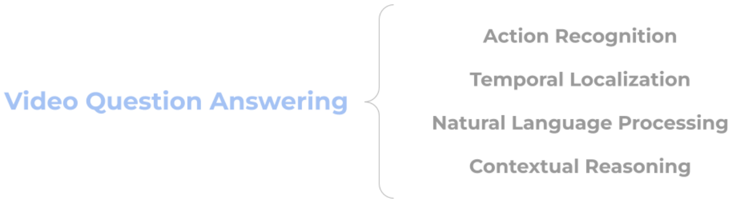


Figure 1.4: A Video Question Answering model should be able to reason about actions, their duration and localization, understand the linguistic cues of the question and perform contextual reasoning.

The field of Video Question Answering is filled with intricate challenges, centered around advancing from simple object recognition to understanding the complex relationships between visual elements in videos, as seen in Figure 1.4.

A significant challenge lies in the computational demand of Video Question Answering systems. The models are typically large and resource-intensive since they need to process extensive visual data, identify relevant elements, and decode the relations between them. They must handle both the visual and linguistic aspects, interpreting the video content while understanding and responding to language-based queries

At the same time, traditional video processing methods mainly operate at the pixel space and are thus inefficient for Video QA tasks. While these methods are effective for certain tasks, like object detection or basic action recognition, they can be limiting for Video Question Answering. Pixel-level analysis often lacks the capability to understand the broader context or story of a video. It focuses mostly on the visual details, having difficulty capturing the relationships between different elements in the scene. Apart from that, processing videos at the pixel level can be extremely resource-intensive. Videos consist of a sequence of frames, each containing a large number of pixels. Analyzing each pixel or small pixel groups across all these frames requires significant computational power and time, making it inefficient for real-time or large-scale applications.

Also, videos can vary widely in resolution, frame rate, and overall quality. High-quality videos present more clear visual data, but many real-world videos can be of lower quality with blurry frames, poor lighting, or artifacts from the compression. Apart from that, videos can come in a lot of different styles and genres, each with different characteristics. For example, a documentary has a different visual style than an animated movie, but the Video QA system should be able to understand visual elements and their interactions in both cases. Also, even in the same video styles, scenes can vary greatly, with very different topics, rapid scene changes, variant camera angles, and directing styles.

1.4 Graph Based Video Question Answering

Videos are the most direct and convenient media to record and reflect our physical world, while the sensory input we receive is multimodal. In the near future, AI agents will be able to assist humans in their everyday lives and daily activities by generating meaningful responses based on their understanding of our dynamic visual world.

Vision and language are two of the most fundamental activities of the human mind, allowing us to gain an understanding of our world, form intricate concepts, reason, and generate ideas. The combination of vision and language has emerged as a captivating research field, gaining more and more interest. Multimodality is critical for intelligence. To go beyond language models and build more aware, capable, and useful systems, the next crucial step will come from vision. So, our goal is to make more efficient multimodal models that can see and understand, show and explain, and eventually interact with our world.

The process of VQA requires transforming both images and questions into feature representations and embeddings, respectively. These features are then combined to generate accurate answers. This process requires a sophisticated interplay between image processing and natural language processing technologies. VideoQA extends VQA's principles to the dynamic and temporally rich domain of videos. The temporal aspect of videos adds a layer of complexity, necessitating an understanding of not just static scenes but also the progression and dynamics within the videos.

1.4.1 Motivation

Video Question Answering presents a set of challenges that stem primarily from the nature of video data. At the core of these challenges is the unstructured and non-salient pixel space of video frames, a characteristic that complicates the extraction of meaningful information. Videos, being highly data-intensive representations, consist of thousands of frames. This not only results in large file sizes but also poses significant difficulties in processing. Additionally, the variation in video resolutions can lead to inconsistencies such as distortions or the inclusion of unnecessary details, complicating the task of analysis and interpretation across different datasets. Another major hurdle is the difficulty of generalization, which is enhanced by domain-specific visual characteristics that videos often possess.

In response to these challenges, our thesis proposes a novel approach, using scene graphs to transform the way videos are processed and understood. Scene graphs offer a structured representation of video content and provide an insightful overview of visual elements and the relations between them. By transitioning from pixel-based analysis to a graph-based representation, we aim to achieve more efficient and equally semantically rich video processing.

We form our approach around the following research question:

> ”Can we decompose videos into structured graphs and perform video questions answering using these graphs instead of the video?”

1.5 Contributions

This study is a work towards answering this research question, focusing on graph-based video question answering. Our contributions include:

- We are the first to explore the use of explicit scene graphs as an intermediate representation for Video Question Answering. Scene graphs are a structured and informative representation of the video content, capturing essential information about the visual elements and the relations between them. By transitioning from pixel space to graph space, we have more efficient and semantically rich representations, reducing the computational load and enhancing the model's ability to understand complex visual scenes.

- We experiment with different types of architectures for the extraction of graph embeddings, providing insights into the effectiveness of different Graph Neural Networks. GNNs enable us to capture the relationships and attributes of the visual elements in a very efficient way. These embeddings provide a deeper understanding of the video content.

- We combine it with a temporal neural network. The graph embeddings are processed by a transformer architecture, specifically a variation of the Hierarchical Conditional Relation Network (HCRN), operating at two levels: clip level and video level. The clip-level processing allows the model to capture spatial relationships focusing on actions and interactions, whereas the video-level processing focuses on understanding the broader context. This hierarchical approach ensures that both the details and the broader context are considered

- We evaluate our method on the Action Genome Question Answering [14] dataset, a real-world dataset consisting of videos depicting humans in everyday activities. Our results demonstrate that our approach is among state-of-the-art methods, and even outperforms them in several question categories.

2 Background

2.1 Introduction

Machine Learning is a branch of Artificial Intelligence (AI), focusing on the concept of enabling computers to learn from experience, much like humans. The history of Machine Learning is marked by a lot of key developments over the last decades. It all started with the development of statistical methods in 1940s, leading to the use of simple statistical ML algorithms in 1950s. Then, Bayesian methods were introduced in 1960, but were followed by a period of pessimism, named the 'AI winter'. In the 1980s, the rediscovery of backpropagation awakened ML research, and in the 1990s a lot of data-driven approaches were born, including SVMs and RNNs. The 2010s focused on Support Vector Clustering and unsupervised methods, while Deep Learning was born in 2010s, allowing the emergence of a lot of ML applications. Recently, in the 2020s, generative AI has captivated research attention, and ML has gained its position in a lot of industrial and commercial fields.

2.2 Machine Learning

Machine Learning is a branch of Artificial Intelligence (AI) that focuses on the use of data to mimic the way humans learn. Its goal is for computers to learn by identifying patterns in data and thus making decisions with minimal to no human intervention. In Machine Learning, computers learn to program themselves, using statistical models and optimization algorithms. To do that, they need to observe a specific set of data, named the train set or training data, and the process in which they learn is called training. However, data can be interpreted in a lot of different ways, using different architectures and parameters, describing different machine learning models. Using different sets of parameters on the same architecture can lead to vastly different behaviors and performances, so adjusting parameters allows for the fine-tuning of models to specific tasks or datasets

To intelligently analyze these data and develop the corresponding smart and automated applications, the knowledge of artificial intelligence (AI), particularly machine learning (ML) is the key [36]. In the past decades, Machine Learning has evolved significantly, from rule-based systems to sophisticated data-driven approaches. This shift has enabled systems to perform complex tasks, like object detection, natural language processing, video understanding and generative tasks that find applications across various fields, like healthcare, finance, recommendation systems, etc [25].

2.2.1 Types of Data

In Machine Learning, a wide variety of data types can be used for training and model development, including numerical, categorical, text, audio, image, time series, or even video data. Numerical data, consisting of discrete or continuous numbers, are used for statistical analyses and quantitative modeling. Categorical data includes distinct categories and labels for classification tasks. Text data is used for Natural Language Processing, and to analyze written language. Image data is used in computer vision and encodes pixel-based information and features necessary for tasks like object detection. Audio data is used in tasks like speech recognition and consists of sound signals. Time-series data contains sequential and time-stamped data points that are used for forecasting and trend analysis. Video data combines visual and temporal elements, essential for understanding dynamic scenes and has application in fields like Action Recognition, Video Understanding and Video Question Answering. Finally, each data type comes with its own challenges and opportunities, leading to the development of such diverse fields in Machine Learning.

2.2.2 Types of Learning

ML approaches can be generally divided into three types: supervised, unsupervised, and reinforcement learning. [34]

Supervised Learning involves training models on labeled data. The model learns to predict outputs from inputs, where each data point is an input-output pair. Supervised learning is used in regression and classification tasks, for applications like image captioning, speech recognition, and visual relationship detection.

Unsupervised Learning, by contrast, deals with unlabeled data. This type of learning discovers hidden patterns and structures from data without any instructions on what to predict. Unsupervised Learning is most commonly used for clustering, dimensionality reduction, and association.

Finally, Reinforcement Learning, is about learning through interacting with an environment. The model makes decisions, gets feedback in the form of reward or punishment, and updates the decision-making policy, thus learning. This type of learning is used in gaming, navigation, and real-time decisions, where the model adjusts its strategies dynamically based on its experiences.

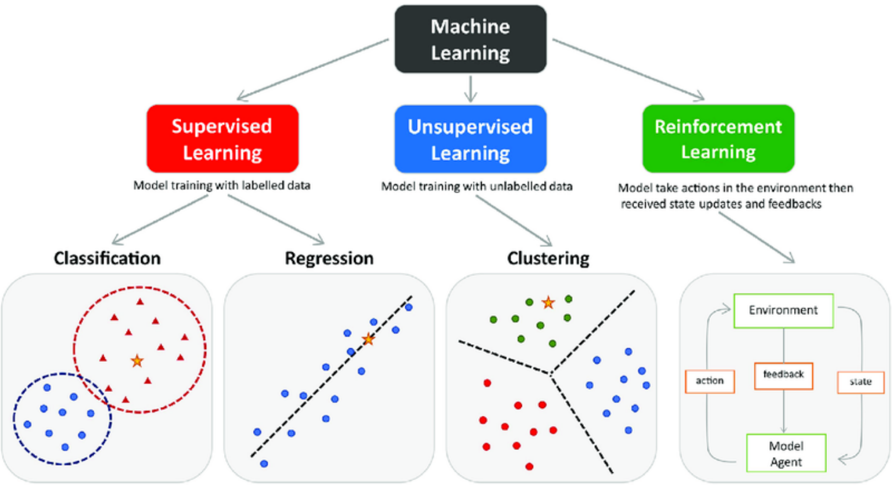


Figure 2.1: The main types of machine learning. Figure from [34].

2.2.3 Perceptrons

A perception is the simplest form of an artificial neural network. It is a single-layer binary classifier, used in supervised learning to categorize inputs into one class. Each perceptron consists of input nodes, weights, a bias, and an activation function, typically a step function. Each input feature, represented as x_i, is multiplied by a corresponding weight parameter w_i. The neuron then aggregates the weighted inputs, sums them and compares the result to a threshold, known as bias b to reach the final output, 0 or 1 [35]. The decision algorithm can also be seen below:

$$\text{Output} = \begin{cases} 1 & \text{if } \sum(w_i \cdot x_i) + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Since the decision is made, we can evaluate its output compared to the correct output, to update the values of w_i and b. The parameters s_i and b are updated through an iterative algorithm trying to map the most inputs to the correct output, going through every sample in the training set, until the perceptron converges.

Since the single neuron models the input to $w_i \cdot x_i + b$, it essentially models a linear function, drawing a straight line through the data. This line is used to separate the data into two classes, and the decision for each data point is based on which side of the line it is.

This model is limited to problems where classes are linearly separable. In many real-world scenarios, data can't be separated by a single line, since there are very complex relations between input variables and classes. To address this limitation, more layers of neurons are introduced, leading to the architecture known as multi-layer Perceptron (MLP), or simply, Neural Network [32].

2.2.4 Neural Networks

The Neural Network, or Multi-Layer Perceptron, is an advancement from the basic perceptron and can be considered its descendant. It addresses the limitations of the single perceptron by handling non-linearly separatable data. The structure of the Multi-Layer Perceptron consists of multiple layers of neurons, as opposed to the single neuron of perceptron.

The Multi-Layer Perceptron introduces hidden layers, each consisting of a set of neurons. Each neuron takes in as input the output of all previous layer neurons, processes it with a weighted sum and passes the result to the next layer. This allows the MLP to capture more complex relationships in the data and learn non-linear relationships.

Each hidden layer, necessarily has an activation function. Without a non-linear transformation, the stacking of two hidden layers would lead to a more complex, but still linear transformation, making it equivalent to a single layer. Some common activation functions used in MLPs are:

- **Sigmoid**: This function transforms the input values within the range of 0 and 1. It is more commonly used for binary classification tasks.
- **ReLU**: This function outputs the input directly if it is positive, otherwise it outputs zero. It's very computationally efficient and allows models to converge faster.
- **Tanh**: This function transforms the input values within the range of -1 and 1. It is similar to sigmoid, but with a broader output range.
- **Softmax** Used primarily in the output layer for multi-class classification tasks, this function converges the output scores from the neurons into probability distribution.

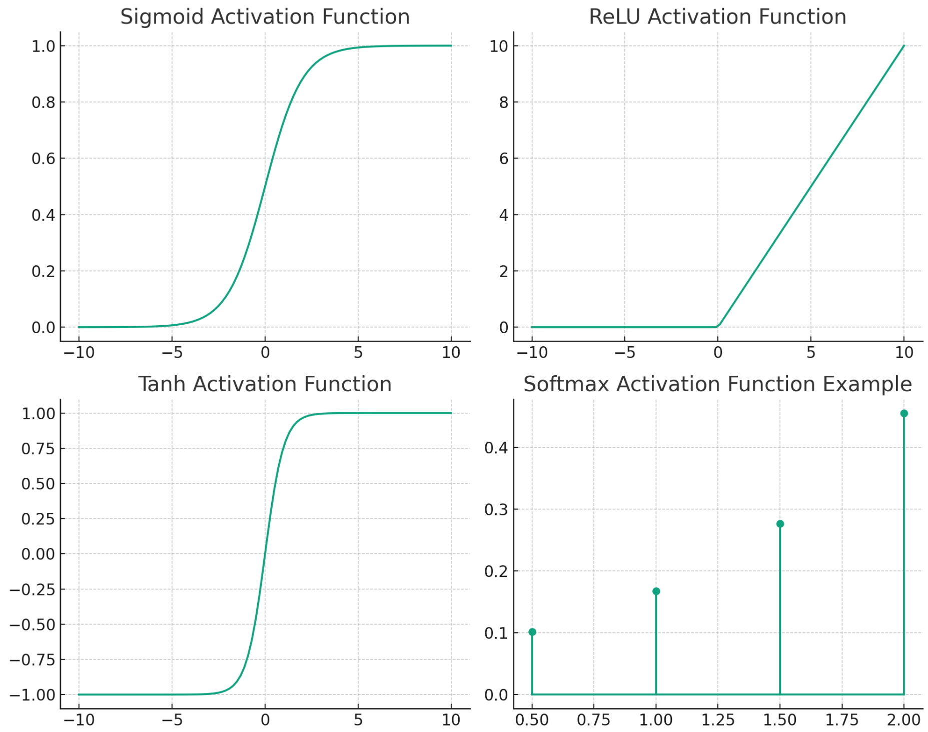


Figure 2.2: The most common activation functions used in MLPs.

2.3 Deep Learning

Deep learning is a machine learning concept based on artificial neural networks. For many applications, deep learning models outperform shallow machine learning models and traditional data analysis approaches [25].

2.3.1 Feed-Forward Neural Networks

Many Deep Learning approarches are based on a variety of neural network architectures. The basic, ”vanilla” neural network is often referred to as FeedForward Neural Network (FFNN), or a Fully Connected Neural Network (FCNN). This term is used to distinguish it from other types of neural networks, like Recurrent Neural Networks (RNNs). The key characteristic of an FFNN is that the information only moves in one direction - forward - from the input nodes, through the hidden nodes, to finally the output nodes. Each neuron in a layer is connected to all neurons in the subsequent layer, which is why it is called ”fully-connected”.

The interesting thing about feed-forward neural networks is the permutation invariance of the inputs in the neuron operations. This means that the relative position of the input features does not influence the output of the neuron operation. Due to this characteristic, FFNNs are suited for independent data, like separate measurement values.

2.3.2 Convolutional Neural Networks

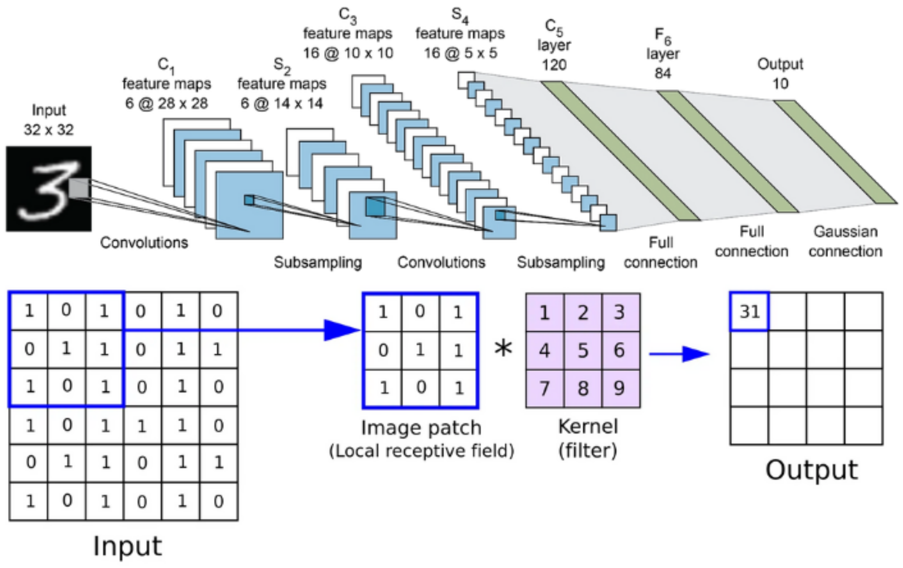
When dealing with images, FFNNs would not be a good fit, since they would require too many parameters and they also can't factor in pixel relative positions, so the spatial information of the pixels in an image would be lost. However, vision tasks also require positional invariance, meaning that the same object should be able to be recognized regardless on where it is placed in an image. The operation that manages to solve all of the above requirements, is convolution. The first paper to introduce Convolutional Neural Networks originated in 1998 [27].

The convolution operation involves sliding a filter (named kernel) K over the input image I and computing the dot product between the filter and the local region of the image to produce a feature map. The feature map represents a processed version of the input image where certain features have been highlighted. Mathematically, this operation for a pixel in position (i,j) is represented as:

$$(I * K)(i, j) = \sum_{m} \sum_{n} I(i + m, j + n) K(m, n)$$

The convolution process extracts local features from the input image by applying the filter to small patches of the image, step by step. This means that the CNN only considers a small portion of the input-image at a time.

CNNs have proven to be very effective for tasks like image recognition, where it is crucial to recognize local patterns like edges, textures and specific objects . We pass an image through multiple learnable filters - each of which extracts different kind of information - this information is much lower in dimension and higher in information density than the pure image pixels. To reduce computational complexity, CNNs use pooling layers to reduce the spatial dimensions of the feature maps. CNNs have become a fundamental tool in many computer vision tasks and have significantly advanced the field of image recognition and processing.



1	0	1	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	0	1	1	1	0
0	1	1	0	1	1
1	0	1	0	1	0

1	0	1
0	1	1
1	0	1

1	2	3
4	5	6
7	8	9

31			

Figure 2.3: Convolution Layer [7] and Convolutional Neural Networks [8]. Figure from [21].

ResNet

ResNet is a type of Convolutional Neural Network (CNN) that was introduced in 2015 and has significantly influenced the landscape of deep learning [19]. As networks grow deeper, they tend to suffer from vanishing gradients, making training less effective. To address this problem, ResNet introduced residual blocks with skip connections, allowing the network to learn identity mappings and ensuring that deeper layers can propagate signals back to earlier layers without loss.

2.3.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) stand out in the realm of neural networks for their distinctive ability to process sequential data, making them particularly adept at tasks involving text, speech, and time series data. Unlike their counterparts, RNNs possess an internal memory that captures information about previous inputs, allowing them to maintain context and make informed predictions based on the sequence of data they receive. This feature is crucial in scenarios where the sequence and context of data points are essential for accurate interpretation, such as language processing or stock market prediction. However, RNNs are not without challenges. They are notoriously known for issues related to vanishing and exploding gradients, which can hinder the network's ability to learn from data, especially when dealing with long sequences. The gradients used during training can become exceedingly small or large, making it difficult for the network to converge and learn the long-range dependencies within the data.

To address these issues, advanced variants like Long Short-Term Memory (LSTM) networks [20] and Gated Recurrent Units (GRUs) [6] were introduced. These architectures incorporate gating mechanisms to control the flow of information, effectively capturing long-term dependencies and mitigating the issues of vanishing and exploding gradients. Consequently, LSTMs and GRUs have become a staple in tasks requiring the understanding of complex, sequential patterns in data, such as machine translation, speech recognition, and text generation. In essence, RNNs, with their unique structure and internal memory, have been pivotal in advancing the field of sequential data analysis. Despite their challenges, the evolution of RNNs into more robust architectures like LSTMs and GRUs showcases the adaptability and potential of neural networks in handling the intricacies and nuances of sequential data.

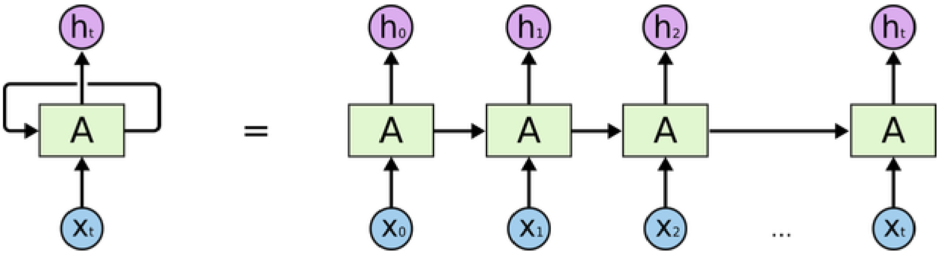


Figure 2.4: Recurrent Neural Network Architecture. Figure from [9].

2.3.4 Attention Mechanisms

Attention mechanisms [47] have emerged as a transformative force in the field of neural networks, particularly enhancing the performance of models dealing with sequential data like text and speech. The core idea behind attention is to allow models to focus on the most relevant parts of the input when performing a task, akin to how human attention works when we concentrate on specific aspects of our environment while ignoring others. In traditional neural network architectures, such as Recurrent Neural Networks (RNNs), each input or word in a sequence is processed in a fixed order, and each step depends on the previous one. While effective, this approach can struggle with long sequences, where distant elements in the input might be relevant to each other. Attention mechanisms address this limitation by enabling the model to weigh the significance of each part of the input data dynamically. This approach allows the model to create a context-sensitive representation of the input sequence, focusing on the most salient parts as needed for the task at hand.

The transformative impact of attention mechanisms became particularly evident with the introduction of the Transformer model, which relies entirely on attention mechanisms, dispensing with the sequential processing inherent to RNNs. Transformers use self-attention to weigh the importance of different words in a sentence, allowing for parallel processing of the sequence and significantly improving the efficiency and performance in tasks like language translation, text generation, and many others. Attention mechanisms have also facilitated the handling of multimodal data, enabling models to attend to different types of input, such as a combination of visual and textual information. This capacity has been instrumental in advancing fields like image captioning and visual question answering, where the interplay between visual elements and textual context is crucial.

In summary, attention mechanisms represent a significant leap forward in the design of neural networks, offering a more flexible and context-aware approach to processing sequential data. By mimicking the selective focus of human attention and allowing models to dynamically prioritize different parts of the input, attention mechanisms have unlocked new possibilities and set new standards in the field of artificial intelligence.

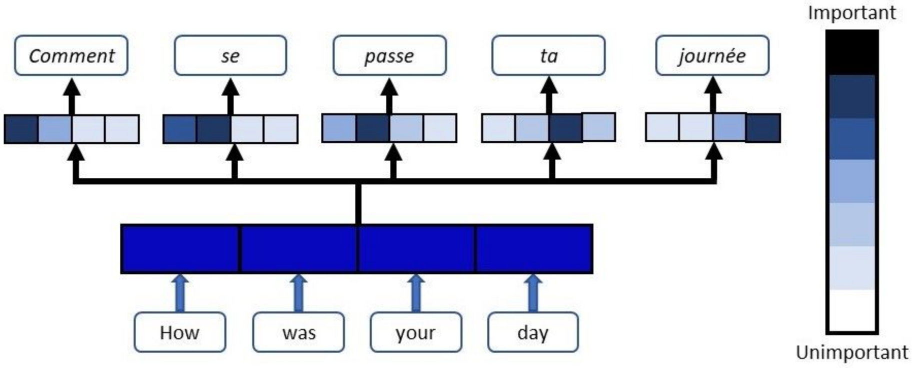


Figure 2.5: Attention Mechanism. Figure from [47].

2.3.5 Trasformers

Transformers have revolutionized the landscape of natural language processing and beyond, marking a significant departure from previous neural network architectures. Introduced in the paper ”Attention is All You Need” [47], the Transformer model stands out for its unique use of attention mechanisms, eschewing the sequential processing typical of Recurrent Neural Networks (RNNs) in favor of parallel processing of sequences. This shift has not only led to substantial improvements in computational efficiency but also set new benchmarks in a wide array of tasks.

At the heart of the Transformer is the self-attention mechanism, which allows the model to weigh the importance of each part of the input data, irrespective of their positions in the sequence. This means that for any given word in a sentence, the Transformer can directly attend to any other word, capturing their relationships and dependencies, regardless of their distance from each other. Such an architecture is particularly powerful in understanding the context and nuances of language, as it can process and relate all words in a sentence simultaneously.

The Transformer model is also inherently scalable and parallelizable, a feature that stands in stark contrast to the inherently sequential nature of RNNs. This characteristic has not only expedited training times but also paved the way for the development of much larger and more powerful models, such as GPT[52] and BERT [10]. These models, pre-trained on vast corpora of text, have demonstrated remarkable capabilities, from generating coherent and contextually relevant text to understanding and answering complex questions with a nuanced grasp of language.

Furthermore, the versatility of the Transformer architecture has transcended the realm of text, finding applications in other domains such as computer vision and multi-modal tasks, where the model's ability to handle sequences can be applied to pixels or combinations of different data types.

In essence, the Transformer model represents a paradigm shift in neural network design, offering a highly effective and flexible architecture that has not only advanced the state of the art in natural language processing but also opened new horizons across the broader field of artificial intelligence. Its influence continues to grow, shaping the future of how machines understand and generate human language.

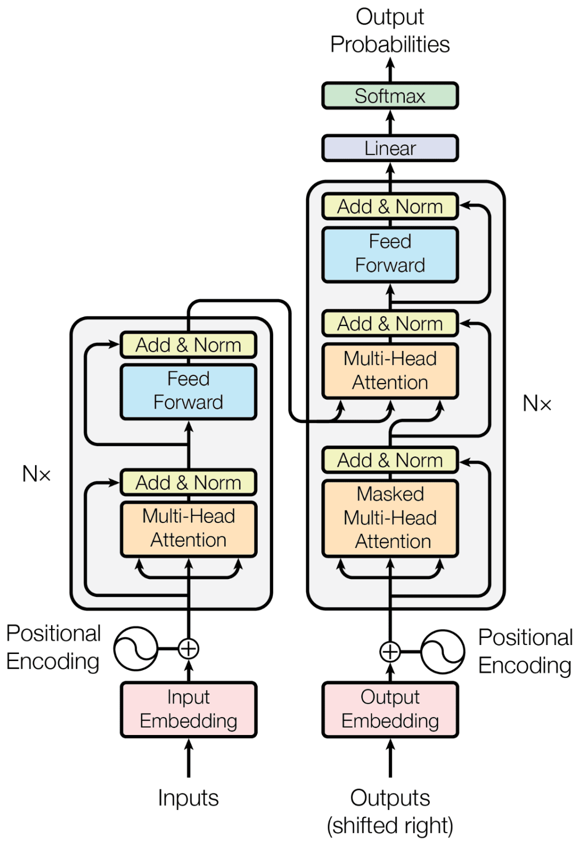


Figure 2.6: Transformer architecture featuring masked multi-head attention mechanisms. Figure from [47].

2.4 Multimodal Machine Learning

Human understanding and perception of the world are guided by diverse sensory experiences, including vision, hearing, smell, taste, and touch. Compared to human learning and perception, multimodal machine learning tries to mimic the way humans process and interpret information from their senses to understand and interact with the world. In machine learning, multimodal learning refers to the use of algorithms and models that can proces s and interpret information from multiple different data sources, such as text, images, audio, video etc. This approach is based on the assumption that combining information leads to more accurate, robust, and comprehensive understanding and reasoning.

Vision and Language

Vision and language are two of the most fundamental capabilities of the human mind, capturing a lot of sensory experience required for reasoning and understanding the world. The majority of everyday tasks we do revolves around vision and language, often requiring their interaction.

The interaction of vision and language has motivated researchers for the last decade, with great efforts to identify the relations between these two modalities, combine them, and reason about them. Language has proven to be an easier modality to deal with than vision since language is a more structured form of data, shaped by well-defined rules of grammar and syntax, making it easier to parse and analyze. Visual data also presents higher dimensionality compared to textual data since an image is composed of pixels, each with color values and spatial relationships, whereas language is represented as lower-dimensional word embeddings. Also, language tends to have less redundancy and noise since each word or phrase normally carries specific information, while in visual data, multiple pixels often depict the same feature or contain misleading visual information.

Vision and language tasks represent a fascinating intersection in the field of machine learning, where the goal is to develop models that can understand and interpret both visual and textual information. Some common vision-language tasks include visual question answering, captioning, image retrieval, text-to-image generation, and many more. Vision and language tasks can be categorized into three major areas. (A) Generation tasks, for example in image captioning text descriptions are generated for a given visual input, and in text-image generation visual output is generated from a textual input. (B) Classification tasks, for example in multiple-choice Visual Question Answering the correct answer to a question is chosen given a visual input, and in Visual Entailment statements regarding a visual input are classified as correct or incorrect. (C) Retrieval tasks, for example in image retrieval images are retrieved based on a textual description. These tasks challenge systems to understand a wide range of detailed semantics of an image, including objects, attributes, spatial relationships, actions, and intentions, and how all of these concepts are referred to and grounded in natural language.

2.5 Visual Question Answering

Visual Question Answering (VQA) is a typical vision-language task that requires the models to jointly reason about both the vision and text data. Given a question written in natural language and a visual object relating to the question, the goal of VQA is to give a correct answer based on the comprehension of the multimodal inputs. It is a very challenging tasks, since it requires a comprehensive understanding of both textual information and visual information independently, but also find the semantic connections between the two [15].

Visual Question Answering [1] was first introduced in 2015 inspiring many datasets focusing on different aspects of the task. Some of the initial steps in most VQA approaches involved extracting features using methods like Bag-of-Words (BOW) or Long-Short-Term Memory (LSTM) encoders for text inputs and Convolutional Neural Networks (CNNs) pre-trained on datasets like ImageNet for visual inputs.

The intriguing part of VQA lies in the way the features are extracted and then combined. An initial approach would be to concatenate the features and then process them through a linear classifier. However, more complex approaches, like attention-based mechanisms have recently emerged and have dominated the field due to their ability to focus on the most relevant segments of the input. For example, in questions about specific objects in an image, attention mechanisms enable the model to concentrate mostly on the relevant image regions and thus enhance the answer accuracy.

The use of pre-trained models like Inception V3 [38] has become a norm in the field. These models are selected for their refined image recognition capabilities, extracting more accurate features from images, which is pivotal for enhancing the performance of VQA systems. About the answer generation, VQA can be split into binary or multiple-choice questions and open-ended questions. For binary or multiple-choice questions, layers like sigmoid or softmax activation functions are used following fully connected layers. For open-ended questions, recurrent networks like LSTMs are employed to generate the answer word by word.

2.6 Video Question Answering

While Visual QA has advanced a lot and developed many impactful works, the questions that can be answered about static images are quite limited as they don't include the temporal dimension found in videos. Video Question Answering can be seen as a natural extension of image QA and consists of a list of temporal image sequences. It is a more challenging task compared to image QA due to the additional complexity of the temporal structure [15].

Visual events are a composition of temporal actions involving actors spatially interacting with objects[14].

Problem Formulation

VideoQA is a task to predict the correct answer a^* based on a question q and a video V. There are mainly two types of tasks in VideoQA: multi-choice QA and open-ended QA.

For multi-choice QA, the models are presented with several candidate answers

$A_{mc} = \{a_1, a_2, ..., a_n\}$ for each question and chose the correct answer $a^* = P_\theta(a|q, V, A_{mc})$ where θ are the model parameters.

For open-ended QA, the problem can be translated into classification, word-by-word generation, and regression (mainly used for counting tasks). The most popular approach is to treat the open-ended QA problem as a multi-class classification problem, where the model classifies a video-question pair into a pre-defined global answer set A_{oe}. The correct answer is picked as $a^* = F(a|q, V)$, where $a \, \epsilon \, A_{oe}$. Open-ended VideoQA can also be treated as a generation problem, where the answer is $a = (a_1, a_2, ..., a_M)$, of length M, given a pre-defined vocabulary set.

2.7 Metrics

For the evaluation of VideoQA models, the key metric used is accuracy. Accuracy is measured as the percentage of correct answers in the entire test set. For multi-choice QA and open-ended QA (treated as classification), accuracy is defined as:

$$acc = \frac{1}{|Q|} \sum_{q \epsilon Q} \boldsymbol{I}[a^* = a],$$

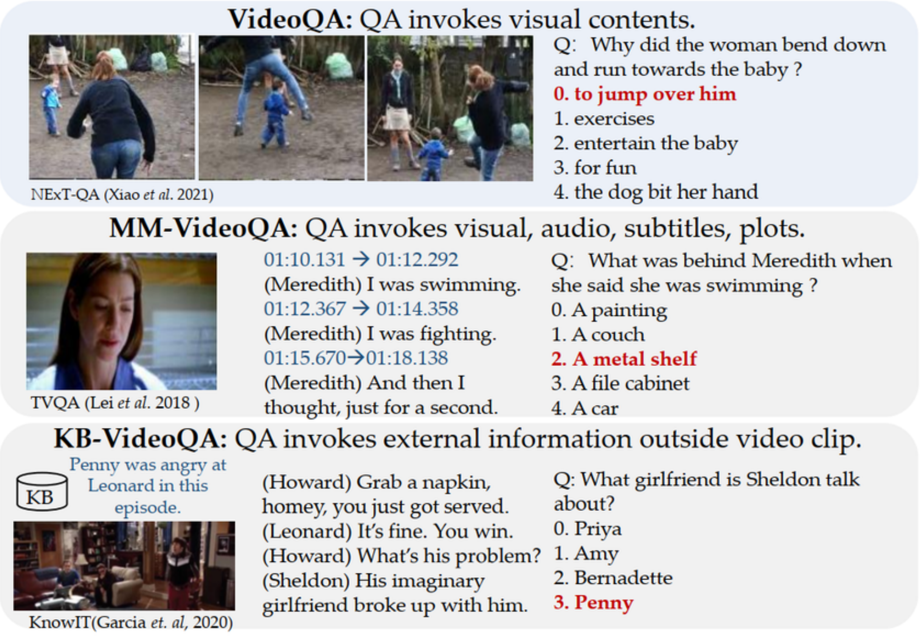
where Q represents the number of QA pairs and $\boldsymbol{I}$ is an indicator function (1 only if $a^* = a$ and 0 otherwise).
Similarly, for open-ended QA treated as word-by-word generation, accuracy is defined as:

$$acc = \frac{1}{|Q|} \sum_{q \epsilon Q} \frac{1}{M} \sum_{i=1}^{L} \boldsymbol{I}[a_i^* = a_i]$$

In this context, accuracy (acc) is calculated as the average number of correct words generated across all questions in the set Q. For each question q in the set Q, the equation iterates over the words in the generated answer sequence of length L, where L is the length of the ground truth answer for that question [58].

2.8 Datasets

Video Question Answering can include questions generated from very different perspectives since the goal is to gain a holistic understanding of videos, guided by specific questions [58].


VideoQA: QA invokes visual contents.

Q: Why did the woman bend down and run towards the baby ?

0. to jump over him
1. exercises
2. entertain the baby
3. for fun
4. the dog bit her hand

NExT-QA (Xiao *et al.* 2021)

MM-VideoQA: QA invokes visual, audio, subtitles, plots.

01:10.131 → 01:12.292
(Meredith) I was swimming.
01:12.367 → 01:14.358
(Meredith) I was fighting.
01:15.670→01:18.138
(Meredith) And then I thought, just for a second.

Q: What was behind Meredith when she said she was swimming ?

0. A painting
1. A couch
2. A metal shelf
3. A file cabinet
4. A car

TVQA (Lei *et al.* 2018)

KB-VideoQA: QA invokes external information outside video clip.

Penny was angry at Leonard in this episode.

(Howard) Grab a napkin, homey, you just got served.
(Leonard) It's fine. You win.
(Howard) What's his problem?
(Sheldon) His imaginary girlfriend broke up with him.

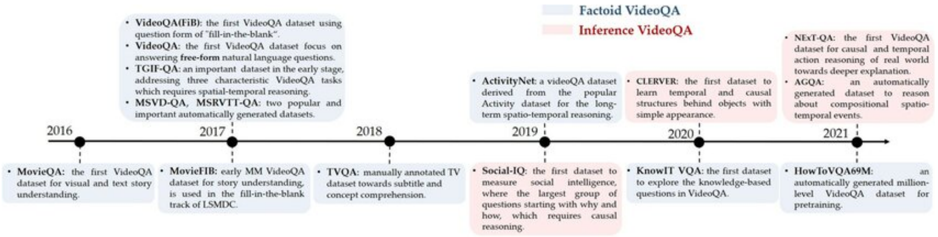
Q: What girlfriend is Sheldon talk about?

0. Priya
1. Amy
2. Bernadette
3. Penny

KnowIT(Garcia *et.* *al*, 2020)

Figure 2.7: Examples of normal VideoQA, Multimodal VideoQA (MM VideoQA) and Knowledge-based VideoQA (KB VideoQA). Figure from [58].

We can classify the datasets according to the data modalities invoked in the question and answers into normal VideoQA, multi-modal VideoQA (MM VideoQA) and knowledge VideoQA (KB VideoQA). Normal VideoQA only invokes visual resources to understand the question and derive the answer, MM VideoQA involves other resources, like subtitles - transcripts and text plots, while knowledge VideoQA demands external knowledge and commonsense reasoning, as seen in Figure 2.7.



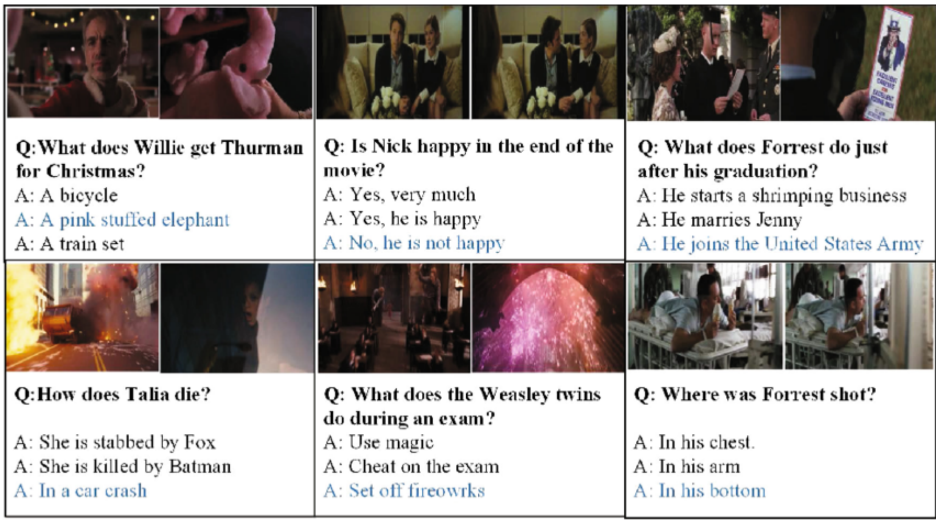
- **VideoQA(FiB)**: the first VideoQA dataset using question form of "fill-in-the-blank".
- **VideoQA**: the first VideoQA dataset focus on answering **free-form** natural language questions.
- **TGIF-QA**: an important dataset in the early stage, addressing three characteristic VideoQA tasks which requires spatial-temporal reasoning.
- **MSVD-QA, MSRVTT-QA**: two popular and important automatically generated datasets.
- **ActivityNet**: a videoQA dataset derived from the popular Activity dataset for the long-term spatio-temporal reasoning.
- **CLERVER**: the first dataset to learn temporal and causal structures behind objects with simple appearance.
- **NExT-QA**: the first VideoQA dataset for causal and temporal action reasoning of real world towards deeper explanation.
- **AGQA**: an automatically generated dataset to reason about compositional spatio-temporal events.
- **MovieQA:** the first VideoQA dataset for visual and text story understanding.
- **MovieFIB:** early MM VideoQA dataset for story understanding, is used in the fill-in-the-blank track of LSMDC.
- **TVQA**: manually annotated TV dataset towards subtitle and concept comprehension.
- **Social-IQ**: the first dataset to measure social intelligence, where the largest group of questions starting with why and how, which requires causal reasoning.
- **KnowIT VQA**: the first dataset to explore the knowledge-based questions in VideoQA.
- **HowToVQA69M**: an automatically generated million-level VideoQA dataset for pretraining.

Figure 2.8: Historical evolution of Video QA Datasets through the time. Blue and red colors represent datasets focused on Factoid VideoQA and Inference VideoQA. Figure from [58].

As we can see in Figure 2.8, several datasets have been developed to train and evaluate Video QA systems, each with unique characteristics and challenges.

2.8.1 MovieQA

MovieQA is the first Video QA dataset and it revolves around the domain of movies, leveraging rich, multi-modal content including videos, subtitles, scripts, and plot synopses. The core of the dataset is the set of question-answer pairs that focus on story-related queries, requiring an understanding of complex narrative elements, character motivations, and plot developments [41].



Q:What does Willie get Thurman for Christmas?
A: A bicycle
A: A pink stuffed elephant
A: A train set

Q: Is Nick happy in the end of the movie?
A: Yes, very much
A: Yes, he is happy
A: No, he is not happy

Q: What does Forrest do just after his graduation?
A: He starts a shrimping business
A: He marries Jenny
A: He joins the United States Army

Q:How does Talia die?
A: She is stabbed by Fox
A: She is killed by Batman
A: In a car crash

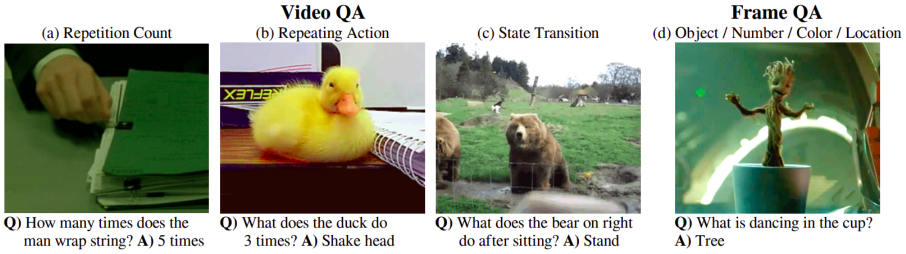
Q: What does the Weasley twins do during an exam?
A: Use magic
A: Cheat on the exam
A: Set off fireowrks

Q: Where was Forrest shot?
A: In his chest.
A: In his arm
A: In his bottom

Figure 2.9: Examples of multiple-choice QA from the MovieQA dataset. Each question has 5 multiple-choice answers. Figure from [41].

2.8.2 TGIF-QA

TGIF-QA [24] is the first Video QA Dataset that requires video-level spatial-temporal reasoning. It consists of a collection of GIFs, mostly sourced from Tumblr, paired with multiple-choice and open-ended questions. The questions are specifically crafted to test a model's ability to understand and interpret the dynamic and often subtle visual cues within these short, looped videos. Unlike traditional video QA datasets that may involve longer sequences and more complex scenes, TGIF-QA focuses on the comprehension of concise, repetitive actions and temporal dynamics within GIFs. The dataset challenges models to not only recognize visual patterns but also to understand the sequence of events, repetitive actions, and the transformation of objects over time.



Q) How many times does the man wrap string? **A)** 5 times

Q) What does the duck do 3 times? **A)** Shake head

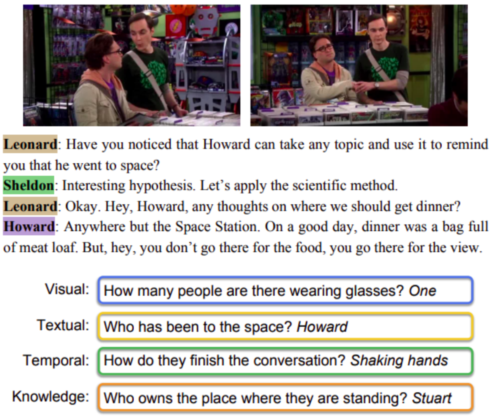
Q) What does the bear on right do after sitting? **A)** Stand

Q) What is dancing in the cup? **A)** Tree

Figure 2.10: Examples of multiple-choice QA from the TGIF-QA dataset. Figure from [24].

2.8.3 KnowIT VQA

KnowIT VQA [13] is a specialized dataset aimed at pushing the boundaries in the domain of video understanding, specifically focusing on the integration of visual content with external knowledge. It consists of video clips from popular TV shows paired with question-answer pairs that require not just an understanding of the visual content and dialogue but also the incorporation of external, common-sense knowledge to provide accurate answers. This unique aspect of KnowIT VQA sets it apart, as it demands a deeper level of reasoning and understanding from AI models. The questions are designed to be complex, often requiring the models to infer emotions, motives, and intentions of characters, or to predict consequences and outcomes based on the given context. KnowIT VQA serves as a critical benchmark for evaluating the ability of Video QA systems to perform high-level reasoning, understand narratives, and effectively integrate visual information with broader world knowledge.



Leonard: Have you noticed that Howard can take any topic and use it to remind you that he went to space?
Sheldon: Interesting hypothesis. Let's apply the scientific method.
Leonard: Okay. Hey, Howard, any thoughts on where we should get dinner?
Howard: Anywhere but the Space Station. On a good day, dinner was a bag full of meat loaf. But, hey, you don't go there for the food, you go there for the view.

Visual: How many people are there wearing glasses? *One*

Textual: Who has been to the space? *Howard*

Temporal: How do they finish the conversation? *Shaking hands*

Knowledge: Who owns the place where they are standing? *Stuart*

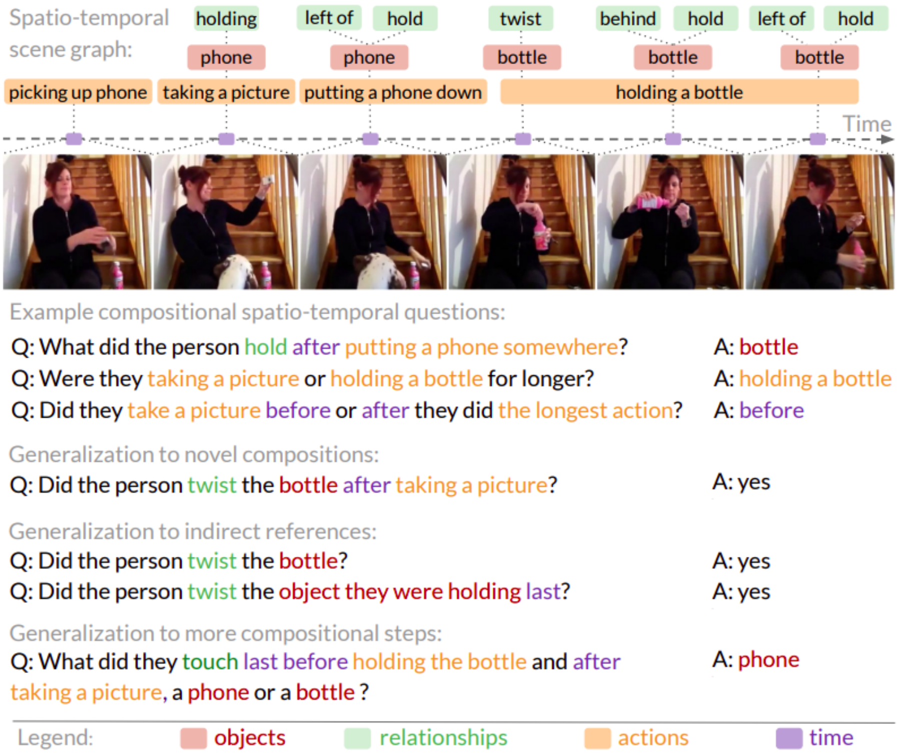
Figure 2.11: Examples from the KnowIT QA dataset. Figure from [13].

2.8.4 AGQA

Action Genome Question Answering [14] is the first large-scale Video QA dataset to include scene graphs to reason about compositional spatio-temporal events. Action Genome Question Answering focuses on understanding of complex actions and interactions within videos. AGQA is built upon the richly annotated Charades dataset and extends it by incorporating a diverse set of question-answer pairs that probe the understanding of sequential actions, the interaction between multiple actors, and the manipulation of objects in various scenes.

The distinctive feature of AGQA lies in its emphasis on the temporal and causal relationships of actions within videos. The Action Genome Question Answering (AGQA) dataset stands out for its incorporation of scene graphs, a feature that significantly enriches its complexity and utility. Scene graphs in AGQA are structured representations of the objects, attributes, and relationships within each frame of the video content. These graphs provide a detailed, structured semantic understanding of the visual elements, going beyond mere object detection to encapsulate the interactions and relations among different objects within the scenes.

The inclusion of scene graphs in AGQA allows for a deeper level of analysis and understanding. It enables models to not only recognize individual elements within the video but also to understand the intricate web of relationships and interactions that define the context and narrative of the scenes.



Example compositional spatio-temporal questions:

Q: What did the person hold after putting a phone somewhere? A: bottle

Q: Were they taking a picture or holding a bottle for longer? A: holding a bottle

Q: Did they take a picture before or after they did the longest action? A: before

Generalization to novel compositions:

Q: Did the person twist the bottle after taking a picture? A: yes

Generalization to indirect references:

Q: Did the person twist the bottle? A: yes

Q: Did the person twist the object they were holding last? A: yes

Generalization to more compositional steps:

Q: What did they touch last before holding the bottle and after taking a picture, a phone or a bottle ? A: phone

Figure 2.12: Examples from the AGQA dataset. Figure from [14].

3 Literature Review

Video Question Answering (Video QA) is an area of research that focuses on answering questions about the content within a video. Video Question Answering approaches showcase great diversity, while the main types of approaches include Memory Networks, Transformers and Graph Neural Networks.

3.1 Memory Networks

Memory Networks are a class of models designed to enhance the capability of neural networks by providing them with an explicit memory component. This memory component allows the networks to store and access information over long periods, making them particularly suitable for tasks that require understanding and reasoning over complex and sequential data, such as Video Question Answering (VideoQA). In the context of Video QA, Memory Networks help in storing information about different frames or segments of the video, enabling the model to refer back to this stored information when answering questions. This approach is beneficial for questions that require understanding sequences or events that happened at different times in the video. For instance, a Memory Network can help answer a question like "What happened to the man after he left the room?" by recalling the relevant video segments stored in its memory.

- **End-to-End Memory Networks**
  The first end-to-end trainable memory network, introduced in 2015, can read and write to an external memory matrix, allowing the network to store past states and later access them to make decisions. This approach is beneficial in VideoQA for storing features or representations of video frames and subtitles. For instance, an approach of a memory network for VideoQA to store video and subtitle features, enabling the model to refer back to earlier parts of the video or dialogue when answering questions about the content.

- **Co-Memory Attention Models**
  CoMem, proposed in 2018, is a two-stream framework, which deals with motion and appearance information separately but in a co-ordinated manner. The co-memory attention module in this framework introduces multi-level contextual information, enabling dynamic fact ensembles for diverse questions. This approach helps synchronize the attention mechanisms across different modalities, such as appearance and motion, leading to a more nuanced understanding of the video content.

- **Heterogeneous External Memory (HME) Models**
  To address the limitations of earlier memory networks that might generate

incorrect attentions by synchronizing appearance and motion features, the HME model was introduced in 2019. The HME model uses attentional read and write operations to integrate motion and appearance features, learning the spatio-temporal attention simultaneously. This model represents a significant advancement in the capacity of memory networks to handle the complexities of video data, particularly in capturing the spatial and temporal dimensions of videos.

- **Progressive Attention Memory Network**
  In 2019, a memory network that utilizes a progressive attention mechanism was introduced. This network progressively prunes out irrelevant temporal parts in the memory bank for each modality and adaptively integrates outputs of each memory. This approach is particularly useful in long video story understanding, such as movies or TV shows, where the model needs to focus on specific parts of the video that are relevant to the question, despite the presence of a large amount of visual information and a long narrative structure.

In summary, Memory Networks in VideoQA are designed to enhance the model's ability to store, access, and integrate information over long sequences, making them particularly adept at understanding complex video content and narratives. The versatility of Memory Networks, as demonstrated by these examples, underlines their significant role in advancing the field of VideoQA by enabling more nuanced and contextually aware models.

3.1.1 Heterogeneous Memory Enhanced Multimodal Attention Model

The most relevant Memory Network to our research is the Heterogeneous Memory Enhanced (HME) [5] architecture. The HME integrates various types of data, particularly video content and associated questions, through a series of well-orchestrated components and processes.

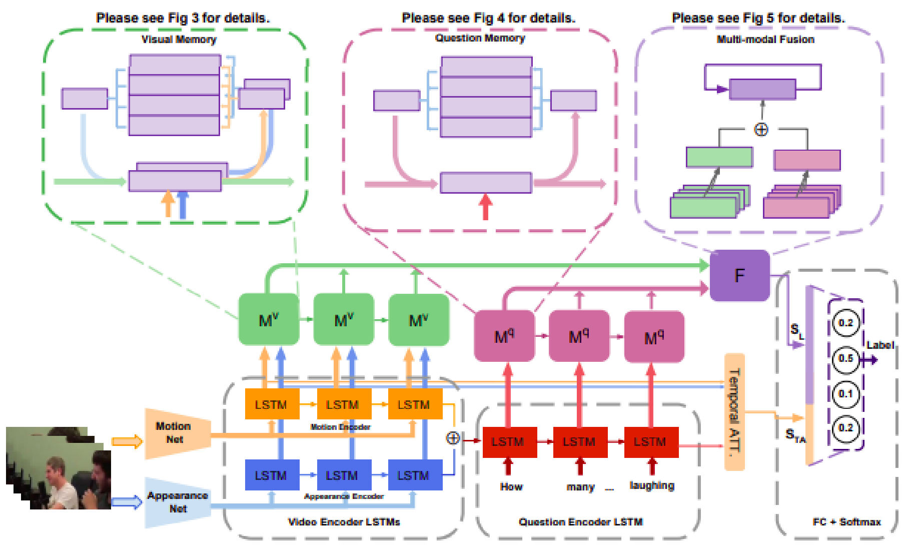


Figure 3.1: HME architecture. Figure from [5].

At its core, the HME architecture employs LSTM (Long Short-Term Memory) encoders for processing both video features and question embeddings. This includes the extraction of appearance features from video frames using pre-trained networks like ResNet [18] or VGG [37], and motion features using a C3D [45] network. What sets this architecture apart is its heterogeneous video memory, which is distinct from standard external memory. This component is designed to accept multiple inputs, including encoded motion and appearance features, and utilizes multiple write heads for determining the content written into memory slots. These memory slots comprise read and write heads, along with three hidden states, enhancing the model's capacity to handle complex video data.

A pivotal aspect of the HME architecture is its multimodal fusion layer. This layer is adept at attending simultaneously to visual and question hints, aligning relevant visual content with key question words. This simultaneous processing of visual and textual data is essential for answering intricate questions that necessitate an understanding of both the video's visual content and the semantics of the question.

The HME model distinguishes itself from existing frameworks by integrating a heterogeneous external memory module with attentional read and write operations, allowing for an efficient combination of motion and appearance features. Furthermore, it enables the interaction of visual and question features with memory contents to construct context-aware features globally. The model's multimodal fusion layer adeptly combines visual and question features with softly assigned attentional weights, facilitating multi-step reasoning.

The HME architecture features two-layer LSTMs for both video and question encoders, with a specified hidden size and memory slot dimension. The memory sizes for video and question components are carefully calibrated to align with the maximum lengths of the videos and questions. Overall, the HME architecture represents a sophisticated approach to processing and integrating diverse data types, particularly suited for applications that require deep interpretation and integration of visual and textual information, as is the case in VideoQA.

3.2 Transformer based Video QA

Transformers have revolutionized the field of natural language processing and extended their influence to Video Question Answering (VideoQA) by providing powerful mechanisms for modeling sequences and relationships within data. Their core mechanism, the self-attention, allows the model to weigh and focus on different parts of the input, making it particularly adept at handling long-range dependencies and varied input lengths. Some more detailed examples and applications of Transformers in the context of VideoQA include:

- **Positional Self-Attention for VideoQA**
  PSAC was introduced in 2019 as an architecture that employs the Transformer without pre-training specifically for VideoQA. This architecture replaces traditional LSTM units with two positional self-attention blocks, enabling the model to capture the intricate relationships within the video content and between the video and the question. A video-question co-attention block is also used to simultaneously attend to both visual and textual information, showcasing the Transformer's ability to handle multi-modal data effectively.

- **Incorporation of Pre-trained Language Models**
  Recognizing the power of pre-trained language models, the pre-trained language-based Transformer, BERT, was incorporated into the domain of VideoQA. These adaptations focus on understanding movies and stories, which require extensive language modeling, like processing subtitles and dialogues. By processing each input modality (video, subtitles) with the question and candidate answers, and then fusing several streams for the final answer, these models demonstrate the adaptability and effectiveness of Transformers in complex, multi-modal understanding tasks.

- **Cross-modal Pre-training and Fine-tuning**
  The potential of Transformers is further unlocked through cross-modal pre-training and fine-tuning. Some approaches applied image-text pre-trained Transformers for cross-modal pre-training, and then fine-tuned them for downstream video-text tasks like VideoQA. Similarly, other VideoQA models have been trained on a large-scale dataset using contrastive learning between a multi-modal video-question Transformer and an answer Transformer, demonstrating the benefits of task-specific pre-training for target

VideoQA tasks. The MERLOT and VIOLET models, which are cross-modal Transformer models trained in a self-supervised manner, further exemplify this approach by leveraging vast amounts of unlabeled data to understand and generate answers based on video content.

In summary, Transformers in VideoQA represent a significant advancement in the field, offering a flexible, powerful, and efficient framework for understanding and integrating information across modalities. The ability of Transformers to handle complex, sequential data and their adaptability to multi-modal tasks have made them a cornerstone in the ongoing evolution of models for VideoQA.

3.2.1 Positional Self-Attention with Co-Attention

One of the most relevant approaches to our work is the Positional Self-Attention with Co-Attention (PSAC) [30]. PSAC aims to overcome the limitations of recurrent neural networks (RNNs), particularly their inefficiency in handling long-range dependencies and sequential data processing.

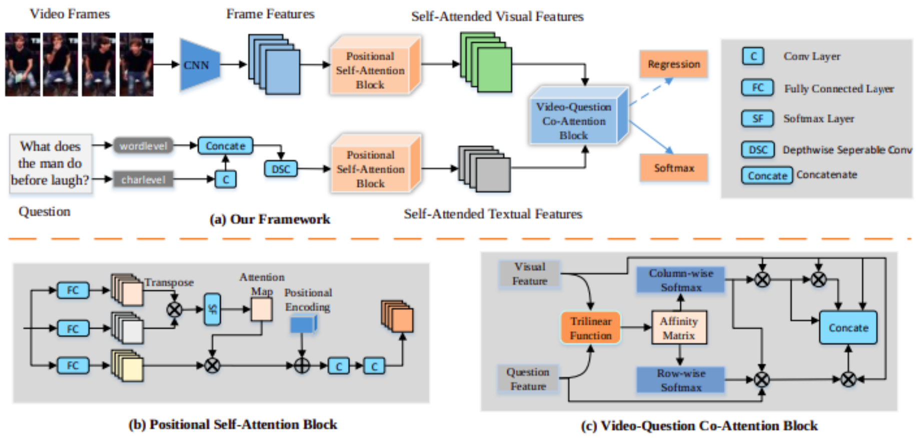


Figure 3.2: PSAC architecture. Figure from [30].

PSAC consists of two key components: Positional Self-Attention blocks (for both video and question processing) and a Video-Question Co-Attention block. The Positional Self-Attention blocks utilize a self-attention mechanism to process video and question data in parallel, capturing global dependencies without the need for RNNs. This is achieved by computing responses at each position in a sequence by attending to all positions within the sequence, along with representations of absolute positions.

The Video-Question Co-Attention block simultaneously models attention on both video and question features, helping to focus on relevant information for accurate answer prediction. This co-attention mechanism is crucial for filtering out irrelevant data and ensuring the generation of precise answers.

3.2.2 Hierarchical Conditional Relational Network

The second most relevant and motivational approach to our work is Hierarchical Conditional Relational Network (HCRN) [26]. HCRN is a hierarchical architecture that processes video data for question answering (VideoQA). This architecture is designed to encode and represent crucial video properties such as object permanence, motion profiles, prolonged actions, and varying-length temporal relations in a hierarchical manner.

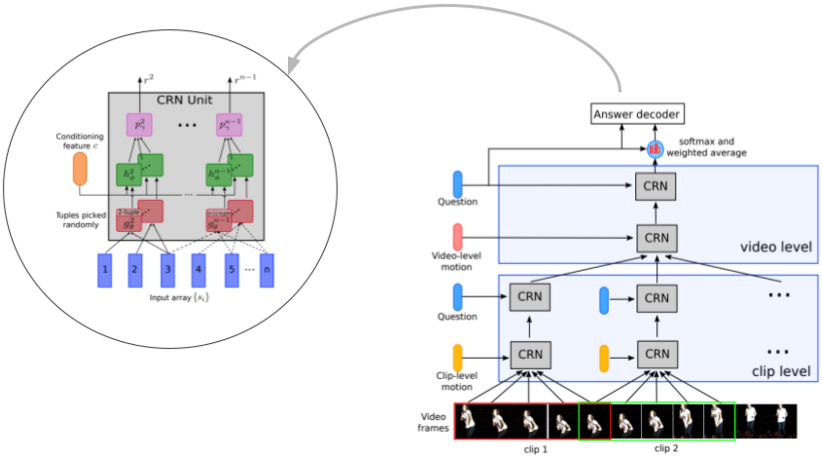


Figure 3.3: HCRN architecture and CRN unit architecture on the top left. Figure from [26].

The architecture, as seen in Figure 3.3, named Hierarchical Conditional Relation Networks (HCRN), is tailored to model videos for QA by integrating different subsystems each designed for specific purposes or data modalities. This hierarchical structure allows the CRNs to encode relations between frame appearances in a clip, integrate clip motion as context, and then progressively integrate linguistic context. This hierarchical stacking supports modeling of structures in video and relational reasoning, enabling multimodal fusion and multi-step reasoning.

The visual representation of a video is divided into equal length clips, each represented by frame-wise appearance feature vectors and clip-level motion feature vectors. For instance, in the case of a video V of L frames divided into N clips, each clip C_i of length $T = \lfloor \frac{L}{N} \rfloor$ is represented by frame-wise appearance features $v_{i,j}$ and motion feature vector f_i. Linguistic representation includes embedding vectors for words in questions and answers, processed through a bi-directional LSTM (biLSTM), forming a combined question representation.

The HCRN architecture operates by first computing frame-wise appearance feature vectors and clip-level motion feature vectors. These vectors are then used to form an input array at clip level, further conditioned on motion features and linguistic cues. The model's loss functions include cross-entropy for general tasks and Mean Squared Error (MSE) for repetition count tasks. For multi-choice question types, the model processes each answer candidate in a similar manner, using shared parameter HCRNs.

The results on various Video QA datasets demonstrate that the HCRN model achieves favorable accuracy across various VideoQA tasks, outperforming or competing well with state-of-the-art models. This underscores the significance of considering temporal relations, motion, and hierarchy in video modeling for question answering.

3.3 Graph Based Video QA

Graph Neural Networks (GNNs) have emerged as a powerful tool in the realm of Video Question Answering (VideoQA), particularly due to their ability to model complex relationships and interactions within data. By representing videos as graphs, GNNs can capture the intricate structure of scenes, including the relationships between different objects and the evolution of these relationships over time. Here are more detailed examples and applications of GNNs in the context of VideoQA:

3.3.1 Situation Hyper-Graph

SHG-VQA (Situation Hyper-Graph based Video Question Answering) [46], includes a situation hyper-graph decoder that identifies graph representations encapsulating actions and object/human-object relationships within video clips. The architecture employs cross-attention mechanisms between these predicted situation hyper-graphs and the question embeddings to accurately predict answers to video-related questions.

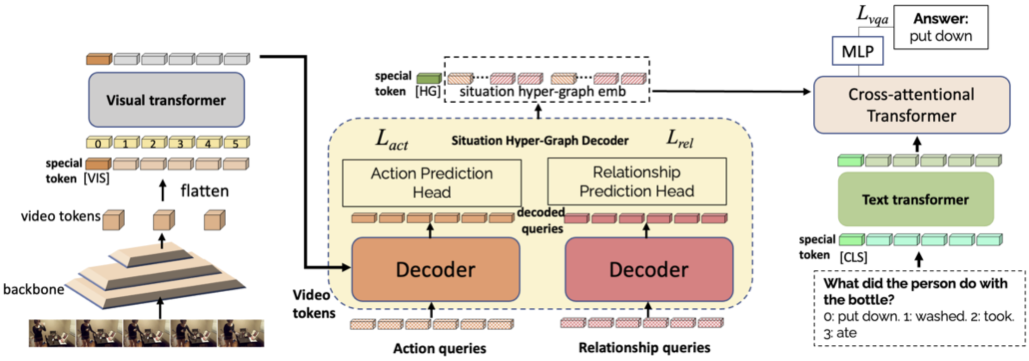


Figure 3.4: SHG-VQA architecture

The SHG-VQA method is trained end-to-end and optimized through a VQA loss function that uses the cross-entropy method, as well as a Hungarian matching loss for the situation graph prediction. This dual loss strategy ensures that the situation hyper-graphs are accurately predicted and aligned with the video content and the questions posed.

An essential aspect of the SHG-VQA architecture is that it focuses less on generating the most accurate scene graph and more on learning a representation of the scene that best facilitates the question answering process. This means that the architecture aims to capture the essence of the scenes and their transitions, optimizing not only for graph accuracy but also for VQA performance.

The SHG-VQA architecture was evaluated on two challenging benchmarks: the STAR dataset, which features various question types and is based on a subset of the Charades dataset, and the Action Genome QA (AGQA) dataset, which tests vision-focused reasoning skills. These datasets are particularly suitable for the SHG-VQA method as they provide dense ground truth hyper-graph information for each video, enabling the architecture to learn the embeddings necessary for answering questions effectively. The results demonstrated that the hyper-graph encoding significantly boosts VQA performance by allowing the system to infer correct answers from spatio-temporal graphs derived from the input video. Furthermore, ablation studies revealed that the quality of the graphs is crucial for VQA performance, underscoring the importance of the SHG-VQA architecture's ability to generate high-quality situation hyper-graphs.

4 Methodology

4.1 Overview of the approach

In this study, we treat Video QA as a classification problem. Specifically, given a video V, meaning a sequence of K frames $V = [f_0, f_1, f_2, ..., f_i, ...f_K]$ and a question q, our aim is to predict the correct answer a^* from the answer vocabulary set. The two typical QA formats are multi-choice QA, where each question is presented with several candidate answers, and open-ended QA, where no answers are provided. Action Genome Question Answering is an open-ended Video QA Dataset, so we follow previous works and set it as a multi-class classification problem. This means that we need to classify the video-question pair into a globally predefined answer set, containing all possible answers. So, we define given a dataset $\mathcal{X} = \{(u_i, q_i, a_i, r_i)\}_{i=1}^{N}$ consisting of N video clips, where $u_i \in \mathcal{V}$ represents the visual input from a sequence of frames, $q_i \in \mathcal{Q}$ is the corresponding question, and $a_i \in \mathcal{A}$ is the ground truth answer for each clip. The objective is to learn a mapping function $f : \mathcal{Q} \times \mathcal{V} \to \mathcal{A}$ that predicts a probability distribution $P(\mathcal{A})$ over the set of possible answers in $\mathcal{A}$.

In our approach, we include another modality, spatio-temporal scene graphs, in order to present our model with a more structured and condensed form of information, trying to achieve a higher-order understanding of the visual content. We represent the given video as a 'hypergraph', describing relationships between objects across the length of the video clip. For each time step in the video, we represent the corresponding frame as a graph, g_t, that captures the entities (objects, actors) and the relationships present in it. The hypergraph for one video is represented by the set of graphs $G = \{g_1, ..., g_T\}$. For each hypergraph, we construct graph embeddings in order to be used with question embeddings for video question answering.

As seen in Figure 4.1, we first sample several frames, extract the scene graph g_i for each frame f_i. These graphs are then processed through a Graph Neural Network (GNN) to capture their topology and structure and produce graph embeddings, e_g . The graph embeddings are then passed through a model to infer answers. We have experimented with several model architectures, leading to our final approach, a hierarchical conditional scene graph model.

Based on the CRN unit [26], we introduce a query-conditioned attention unit for graph embeddings, designed to direct focus within the scene graph embeddings based on the specific queries. We use these blocks to build a deep network architecture to support reasoning guided by linguistic questions on the hierarchy of video structure. We set up two different granularities, one on clip level and one on the entire video level. At each hierarchy level, we use one adjusted CRN unit, conditioned on linguistic cues. The input array at the clip level consists of the video hypergraph, while at the video level is the output of the clip level. To our knowledge, this is the first hierarchical approach using scene graphs for more accurate Video Question Answering.

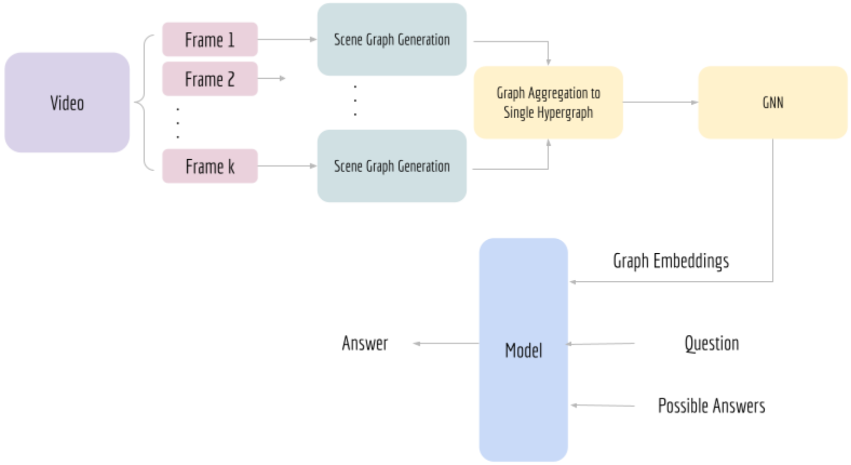


Figure 4.1: Hierarchical conditional approach for Video Question Answering with the use of Scene Graphs architecture. The adjusted CRN units are stacked in hierarchy, processing the hypergraph in different granularities conditioned on linguistic cues. The final output is joined with the question and fed into an output classifier for prediction.

So, our approach consists mainly of 5 steps:

1. Sample frames from the clip
2. Extract scene graph for each sampled frame
3. Aggregate graphs
4. Pass graphs through GNN to extract video-level graph embeddings
5. Classify the answer based on the question and graph embeddings

We will further analyze each of the above steps in this chapter.

4.2 Data Processing and Feature Extraction

In Video Question Answering, each sample consists of a video and a question.

4.2.1 Video

We extract a video at p frames per second and then partition it into K clips of length N. For each clip C, we maintain a dense stream of N frames to obtain the clip-level motion feature and a sparse stream of γN frames $(\gamma \epsilon (0, 1))$ to obtain the region and frame appearance features. In our baselines implementation, the motion features and frame appearance features are extracted from pre-trained CNNs, specifically ResNeXt-101 [17] for motion and ResNet-101 [18] for frame appearance.

Appearance Features

For ResNet-101 [18], an image is passed through a deep network consisting of 101 layers, each consisting of residual blocks. Each block has convolutional layers, batch normalization, and ReLU activations, but the key element is the shortcut connection that skips layers. This design ensures that the signal can propagate effectively through the network without the vanishing gradient problem, allowing the model to learn even from very deep layers. As the data progresses through these layers, ResNet-101 efficiently extracts appearance features from the image, identifying and learning from complex patterns and textures. This deep and intricate processing enables the model to recognize and classify images with high accuracy. We can describe the process of obtaining the appearance features as $Af = g_q(f_i)$, where Af is the appearance feature vector, f_i is the ith frame and g is the ResNet-101 model.

The resulting appearance features are high-level representations of visual content captured by ResNet-101. These features encode rich information about the appearance of objects, colors, textures and spatial arrangements within images.

Appearance Features enhance the model's understanding by providing detailed information about the visual content of the video. These features are crucial for recognizing objects, actions and general visual patterns in the video frames.

Motion Features

ResNeXt-101 [17] takes a slightly different approach, focusing on handling multiple feature representations within its structure. When data enters a ResNeXt-101 model, it is subjected to group convolutions within the residual blocks, where the input is divided into smaller subsets, each processed in parallel paths. This methodology allows the model to capture a diverse range of features simultaneously, making it particularly adept at extracting motion features from sequences of images, like frames in a video. By analyzing these frames collectively, ResNeXt-101 can detect and interpret subtle changes and movements, effectively understanding the temporal dynamics of the visual data. Thus, while ResNet-101 excels in extracting detailed appearance features from static images, ResNeXt-101 is more attuned to capturing and analyzing motion features in dynamic, sequential data. We can describe the process of obtaining the motion features as $Mf = g_m(C)$ where Mf is the motion feature vector, $C = [f_i, f_j, .., f_N]$ is the clip of N sparsely sampled frames and g_m is the ResNeXt-101 model.

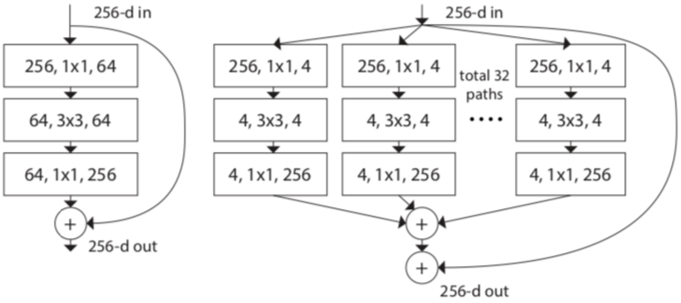


Figure 4.2: Left: ResNet block[18] and Right: ResNeXt architecture [17]. Figures from [18] and [17] respectively.

Motion features are essential for capturing the temporal dynamics and changes in the video content. These features enhance the model's ability to detect and interpret human activities, object interactions, and other dynamic scene interactions.

4.2.2 Question

To obtain a well-contextualized word representation, we extract the token-wise sentence embeddings from the penultimate layer of a BERT model [10]. In BERT, the data journey begins with tokenizing the input text, including a special CLS token at the start. This tokenization is essential for capturing the sentence's overall context. As the tokens pass through BERT's layers, they are analyzed in a bidirectional context, allowing the model to understand each word in relation to the entire sentence. The penultimate layer is crucial for extracting embeddings, particularly the CLS token embedding. This layer provides a rich, balanced representation of the sentence, encapsulating the contextual nuances and relationships between words. The CLS token embedding from this layer offers a comprehensive view of the sentence, crucial for tasks like classification or sentiment analysis, highlighting BERT's capability to deliver deep, context-aware word representations. We can describe the process of obtaining the question features as $Qf = g_q(q)[CLS]$ where Qf is the question feature vector, q is the question, g_q is the BERT model and [CLS] represents the selection of only the CLS token embedding from BERT's output.

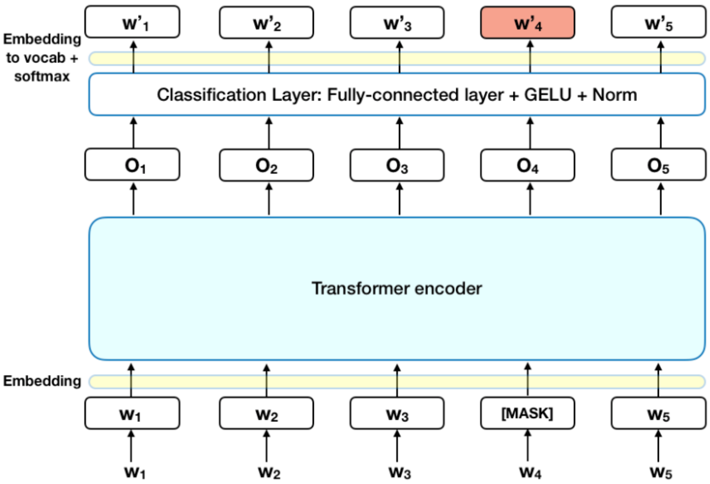

W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 4.3: Overview of BERT architecture demonstrating the process of extracting CLS question embeddings. Figure from [10].

4.3 Scene Graph Generation

After dividing the video into clips C and sampling frames from them, we generate their scene graphs using a pre-trained SGG model. This process can be summed up in the Figure 4.4.

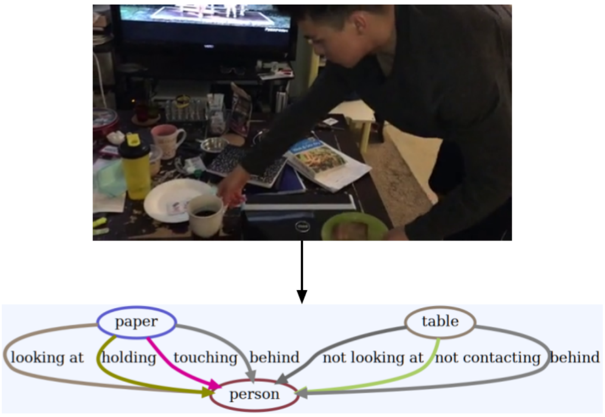


Figure 4.4: The ground truth scene graph for this frame including major interactions of the person with objects in the room.

Scene graphs are generated from an image using a process called Scene Graph Generation (SGG). SGG involves detecting objects and their relationships in an image. To generate scene graphs:

1. **Input Image (f_i)**: The process begins with an input image f_i containing various objects and their visual features. The image is fed into a pre-trained and frozen Faster R-CNN model, which outputs a set of bounding boxes and a feature map.

2. **Object Detection**: The bounding boxes obtained from the Faster R-CNN model represent the detected objects in the image. Each object is assigned a unique identifier and its corresponding visual features.

3. **Relationship Detection**: Once the objects are detected, the next step is to determine their relationships. This involves identifying the interactions between different objects in the image. The triplets describing the objects and their interrelations make up the scene graph of the input image, g_i. Various SGG methods, such as VTransE [57], MOTIFS [56], and VC-Tree [40], can be applied to infer these relationships.

In summary, scene graphs are generated from an image by first detecting objects and then referring the relationships between objects pairs in the form of triplets. Triplets provide a structured and consise representation of relationships between entities in a scene. They also encompass linguistic aspects, comprising subject - predicate - object relationships. This format facilitates efficient storage, retrieval and processing of information about the scene.

In our study, we use MOTIFS [56] to extract scene graphs for our video frames, but any other scene graph generation model can be used in its place.

MOTIFS

MOTIFS [56] stands for “Multimodal Online Temporal Fusion for Image-to-Sentence Matching.” It is a method used for the task of image-to-sentence matching, specifically in the context of multimodal retrieval, where the goal is to find relevant sentences given an input image.

The key idea behind MOTIFS is to temporally fuse multiple modalities (such as images and sentences) in an online manner. It takes into account the sequential nature of multimodal inputs and learns to align the modalities at different time steps. This fusion process allows the model to capture fine-grained temporal relationships between the modalities, leading to improved matching performance.

MOTIFS utilizes a recurrent neural network (RNN) to model the temporal dynamics of sentences and employs a convolutional neural network (CNN) to capture visual features from images. These two networks are jointly trained to learn the cross-modal alignments and capture the semantic relationships between images and sentences.

Overall, MOTIFS is a method that effectively combines image and sentence modalities, taking into account their temporal relationships, to enhance the task of image-to-sentence matching in multimodal retrieval scenarios.

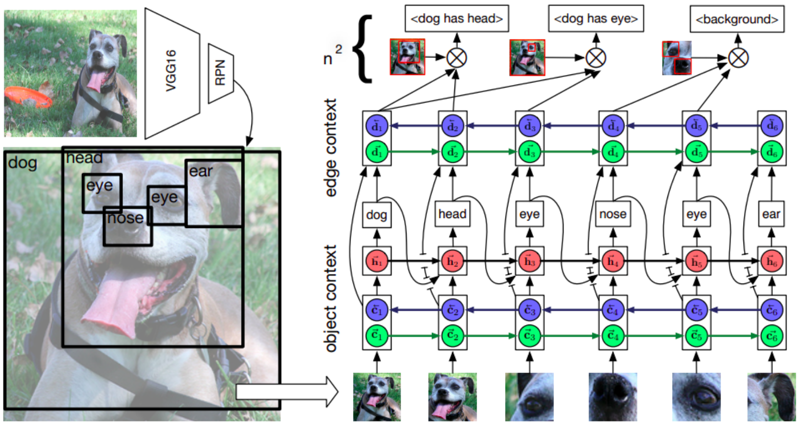


Figure 4.5: A diagram of MOTIFS architecture. Figure from [56].

Motifs is wrapped in 'Scene Graph Benchmark' Github Repository [39], so we use it to extract the scene graphs. It processes the input image f_i and returns hundreds of triplets, along with their confidence score and each object's bounding box coordinates and confidence. We implement a post-processing filtering, keeping only the most confident objects and relationships, resulting in less than 50 triplets per frame. We thhen use these triplets to form the graph g_i for each frame f_i.

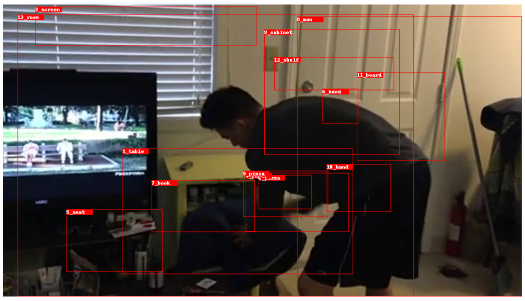


Figure 4.6: An example of object detection used in MOTIFS, filtered by confidence score more than 10%.

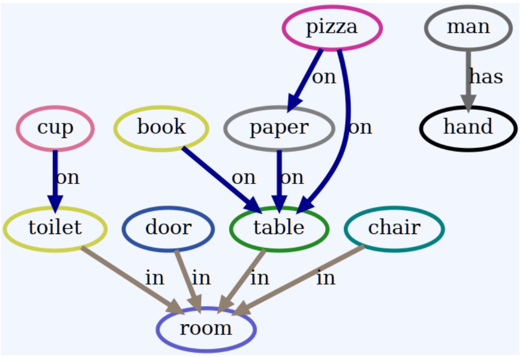


Figure 4.7: An example of the extracted scene graph used in MOTIFS, after the post-processing for the frame of Figure 4.6.

4.4 Graph Neural Networks (GNNs)

The next step in our process is to generate graph embeddings, $g_{e,i}$, for each sampled frame's scene graph, g_i.

Scene graphs are by definition graphs, where objects - nodes are connected with the relationships between them as edges. Graphs are fundamental structures in mathematics and computer science, used to model a wide array of complex systems. Formally, a graph is defined as a set of nodes (or vertices) and a set of edges connecting these nodes. Mathematically, this can be represented as $G = (V, E)$, where G stands for a graph, V is the set of vertices, and E is the set of edges.

Graphs can be classified into various types, such as directed and undirected graphs, depending on the nature of the relationships between the nodes. In a directed graph, each edge has a direction, indicating a one-way relationship, while in an undirected graph, the edges represent a two-way, reciprocal relationship. Graphs are also characterized by their vertices (or nodes) and edges. The edges can represent various types of relationships or interactions between the vertices. An important concept in graph theory is the adjacency of vertices, which can be represented mathematically through an adjacency matrix A. In an adjacency matrix, each element A_{ij} indicates whether there is an edge from vertex i to vertex j. Another key aspect is the degree of a vertex, which is the number of edges connected to it. In directed graphs, the degree is often split into the in-degree and out-degree, representing incoming and outgoing edges, respectively.

Scene graphs represent a specialized application of graph theory in the field of computer vision. A scene graph for an image is a graph where nodes correspond to objects within the image, and edges represent the relationships or interactions between these objects. Generating a graph from a scene graph is modeled very naturally. Each object identified becomes a node in the graph. The relationships between these objects are then analyzed and represented as edges, forming a comprehensive graph that encapsulates the dynamics of the scene.

4.4.1 Graph Formulation

Scene graphs, a structured representation of the elements within an image and their relationships, can be by definition modeled into graphs for Graph Neural Networks (GNNs) to process. In a scene graph, nodes typically represent objects within the image, and edges represent the relationships or interactions between these objects. For example, in an image depicting a park, the nodes could represent entities like "tree," "bench," or "person," while the edges could describe relationships such as "next to" or "sitting on."

To integrate scene graphs into GNNs, each node in the scene graph is encoded with features that describe the basic property of the corresponding object, meaning its type. The edges are also encoded with features that describe the type of relationship between the nodes they connect. Both features, node and edge ones, are formulated as 1-hot vectors of their category.

The GNN processes this information by aggregating features from neighboring nodes and edges, learning to identify patterns and interactions within the graph structure. This enables the GNN to produce informative graph features that enable complex reasoning about the frame by understanding the relationships and context provided by the scene graph.

4.4.2 GNN Architectures

The goal of a Graph Neural Network can be to encode nodes and edges as low-dimensional vectors that summarize their graph position and the structure of their local graph neighborhood, as well as the features they may have. The defining feature of a GNN is that it uses a form of neural message passing in which vector messages are exchanged between nodes and updated using neural networks.

For a node u, its updated state $h_u^{(k+1)}$ at iteration $k+1$ is calculated as follows:

$$m^{(k)}(u) = AGGREGATE(\{h_v^{(k)} : v \in N(u)\}) \tag{1}$$

where $N(u)$ represents the set of neighbors of node u, and $m^{(k)}(u)$ is the aggregated message for node u at iteration k.

The update rule for the node's state is then given by:

$$h_u^{(k+1)} = UPDATE(h_u^{(k)}, m^{(k)}(u)) \tag{2}$$

In this, $UPDATE$ is a function that combines the node's current state and the aggregated message to produce the new state.

A Graph Neural Network (GNN) can produce a graph representation by updating the features of each node based on the features of its neighboring nodes. This is achieved through the message-passing mechanism, where nodes exchange information with their neighbors. The process involves aggregating these messages and updating each node's state iteratively. Over multiple iterations, each node's features become a representation that reflects not only its own attributes but also the collective information of its local graph neighborhood. This results in a graph where the representation of each node encapsulates both individual and contextual information from its surroundings.

For example, if we had a scene graph comprising of the objects "person", "tree", and "bench", with relationships

person - sitting on - bench,
person - next to - tree,
tree - next to - bench

the GNN would pass messages between the nodes and their neighboring nodes to update node embeddings based on the local graph structure. At each iteration the nodes would aggregate information from their neighbors, updating their own embeddings to reflect the relationships and context. For example, the person node would receive information about the tree node, being next to the bench. After multiple message-passing iterations, each node would have a final representation that captures both the attributes of the node itself and the relationships with the other nodes.

In order to get graph representations for each frame scene-graph, we used different GNN architectures from the broader literature, used in various applications. The selected architecture should be able to capture not only the nodes position and graph structure, but also the node and edge features, that represented the object and relationship class respectively.

4.4.3 Graph Attention Network

GAT [48] (Graph Attention Networks) operates by learning to assign different weights or importance to the nodes and edges in a given graph. GAT can be used for various tasks including node classification, link prediction, and graph classification. The core idea behind GAT is the attention mechanism, which enables the model to focus on different parts of the graph when computing embeddings.

Instead of treating all nodes equally, GAT assigns attention coefficients to each node based on its neighbor nodes and the features associated with them.
The process of producing graph embeddings using GAT involves the following steps:

1. **Input Graph**: The initial input to the GAT model is a graph represented by nodes and edges, along with associated features or attributes for each node.

2. **Node Embeddings**: GAT starts by transforming the initial node features using a shared linear transformation, producing node embeddings. These embeddings capture the information about individual nodes.

3. **Attention Mechanism**: GAT employs the attention mechanism to compute the edge weights or attention coefficients for each node and its neighbors. The attention coefficients are learned during training and represent the importance or relevance of each neighbor node for a given node.

4. **Aggregation and Weights**: Once the attention coefficients are computed, GAT performs a weighted aggregation of the neighbor node embeddings based on these coefficients. This aggregation step takes into account the importance of each neighbor when calculating the representation of a node.

5. **Non-Linearity and Output**: After aggregation, a non-linear activation function (e.g., ReLU) is applied to the aggregated features, enhancing the expressive power of the model. Finally, the output is generated, which may be in the form of node labels, graph properties, or embeddings.

The Graph Attention Layer can be described as:

$$h_i^k = \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k W^k h_j^{k-1}\right)$$

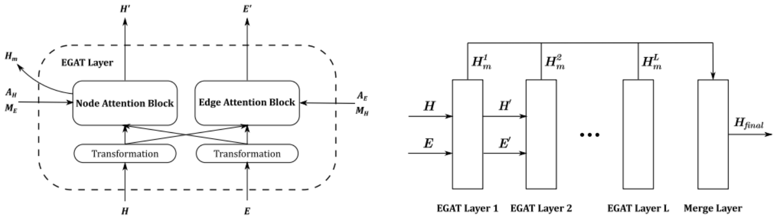
where h_i^k represents the output feature representation of node i in the k-th attention head, σ denotes the activation function, N_i represents the neighborhood of node i, α_{ij}^k are the attention coefficients computed by the k-th attention mechanism, W^k is the weight matrix, and h_j^{k-1} represents the input feature representation of node j in the k-th attention head.

In summary, the data flow in GAT involves passing information between connected nodes in the graph through the attention mechanism. The attention coefficients determine the importance of each node's neighbors, allowing the model to selectively focus on relevant nodes during computation. This attention-based information flow helps GAT capture the structural and relational dependencies present in the graph, resulting in rich and meaningful graph embeddings.

Edge-Featured Graph Attention Network

Graph Attention Network takes into consideration node features, but ignores edge features, that in our case play a similarly important role. So, Edge-featured Graph Attention Networks [4] (EGATs) were born, as an extension to Graph Attention Networks (GATs).

A single EGAT layer contains two different blocks: node attention block and edge attention block. Each EGAT layer is designed in a symmetrical scheme; thus, the node and edge features can update themselves in a parallel and equivalent way.



(a) structure of EGAT layer

(b) EGAT architecture

Figure 4.8: (a) the structure of one EGAT layer. Produces 2 mapping matrices, one for nodes and one for edges respectively. (b) the architecture of EGAT, constructed of several EGAT layers and a merge layer. Both figures from [4].

Each EGAT layer accepts a set of node features, $\mathbf{H} = \{\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \ldots, \tilde{\mathbf{h}}_N\}$, with $\tilde{\mathbf{h}}_i \in \mathbb{R}^{F_H}$, as well as a set of edge features, $\mathbf{E} = \{\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \ldots, \tilde{\mathbf{e}}_M\}$, with $\tilde{\mathbf{e}}_p \in \mathbb{R}^{F_E}$ as inputs. Here, N and M represent the number of nodes and edges, while F_H and F_E symbolize the number of their respective features. After processing, the layer will produce high-level outputs, which include a new set of node features, $\mathbf{H}' = \{\tilde{\mathbf{h}}'_1, \tilde{\mathbf{h}}'_2, \ldots, \tilde{\mathbf{h}}'_N\}$, with $\tilde{\mathbf{h}}'_i \in \mathbb{R}^{F'_H}$, and a new set of edge features, $\mathbf{E}' = \{\tilde{\mathbf{e}}'_1, \tilde{\mathbf{e}}'_2, \ldots, \tilde{\mathbf{e}}'_M\}$, with $\tilde{\mathbf{e}}'_p \in \mathbb{R}^{F'_E}$.

The Node Attention Block processes node features H and edge features E, outputting a new set of node features H'. Edge features in E, organized in a specific order, do not directly show their connections to adjacent nodes. A mapping transformation converts E into E^*, where each element $\tilde{e}_{ij}$ is related to nodes i and j. This transformation employs an edge mapping matrix M_E, a $N \times N \times M$ tensor, reshaped to $N^2 \times M$ for matrix multiplication with E, and then back to $N \times N \times F'_E$, transforming E into an adjacency-like structure. M_E is unique for each graph and is determined in preprocessing.

In this model, thanks to its adjacency-based structure, identifying edges between specific nodes is efficient. The edge-integrated attention mechanism focuses on each node, considering not only the features of neighboring nodes but also the connecting edges. For a given node i, attention weights w_{ij} are calculated for all nodes j in N_i, the set of first-order neighbors of i, including i itself. Features are concatenated and processed through a LeakyReLU activation function, parameterized by a weight vector $\tilde{a}$. Normalization of these weights is performed using a softmax function across all j in N_i. This process, including the node's own features and those of its neighbors, is mathematically represented as:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\tilde{a}^T[\tilde{h}_i^k \tilde{h}_j^k \tilde{e}_{ij}]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\tilde{a}^T[\tilde{h}_i^k \tilde{h}_k^k \tilde{e}_{ik}]))}$$

This approach allows the model to effectively integrate edge features, enhancing the representation of each node's context.

In EGAT, node features are periodically updated in node attention blocks to acquire high-level features. However, reusing the original low-level edge features for weight computation is not optimal. To address this and maintain a balance between nodes and edges, edge attention blocks are introduced. These blocks take node features H and edge features E, and output updated edge features E'. The update process involves aggregating adjacent edges' features. In undirected graphs, adjacency is defined by sharing a common vertex. The method involves transforming the graph, swapping the roles of nodes and edges. This concept, also applied in directed graphs for community detection, involves creating a new graph where the original graph's nodes and edges are interchanged. The attention mechanism is then easily applied on this new graph, using a node mapping matrix M_H. The normalized attention weight for an edge p in relation to edge q is given by:

$$\beta_{pq} = \frac{\exp(\text{LeakyReLU}(\tilde{b}^T[\tilde{e}_{pk} \tilde{e}_{qk} \tilde{h}_{pq}]))}{\sum_{k \in N_p} \exp(\text{LeakyReLU}(\tilde{b}^T[\tilde{e}_{pk} \tilde{e}_{kk} \tilde{h}_{pk}]))}$$

Here, N_p is the first-order neighbor set of edge p and $\tilde{b}$ is a weight vector.

4.4.4 Graph Isomorphism Network

The Graph Isomorphism Network (GIN) is a type of Graph Neural Network architecture designed for graph representation learning. It operates on a neighborhood aggregation scheme to compute each node's representation vector in a graph. GIN is known for its expressive power, making it a highly capable GNN architecture. The node embeddings in GIN are updated iteratively based on neighborhood information, following the update equation:

$$h^{(k+1)}(v) = \mathrm{MLP}_k \left((1 + \epsilon^{(k)}) \cdot h^{(k)}(v) + \sum_{u \in N(v)} h^{(k)}(u) \right)$$

In this equation, $h^{(k)}(v)$ is the representation vector of node v at iteration k, $N(v)$ is the set of neighboring nodes of v, and MLP_k is a multi-layer perceptron. The term $(1+\epsilon^{(k)})$ scales the node's own representation, enhancing the model's ability to capture complex graph structures.

Graph Isomorphism Network with Edge Features

GINE was introduced in 2021, extending the GIN architecture with some minor modifications to include edge features, as well as center node information in the protein ego-networks. GINE was used for molecular property detection. In molecular property prediction, node and edge features are initially 2-dimensional categorical vectors. Unique categories are used for masked nodes/edges and self-loops. For input to GNNs, these vectors undergo embedding:

$$h_v^{(0)} = \mathrm{EmbNode1}(i_{v,1}) + \mathrm{EmbNode2}(i_{v,2})$$
$$h_e^{(k)} = \mathrm{EmbEdge}_1^{(k)}(j_{e,1}) + \mathrm{EmbEdge}_2^{(k)}(j_{e,2}) \text{ for } k = 0, 1, \ldots, K-1$$

In GINE, embedding operations convert integer indices to d-dimensional vectors. Node representations are updated at the k-th GNN layer:

$$h_v^{(k)} = \mathrm{ReLU}\left(\mathrm{MLP}^{(k)} \left(\sum_{u \in N(v) \cup \{v\}} h_u^{(k-1)} + \sum_{e=(v,u): u \in N(v) \cup \{v\}} h_e^{(k-1)} \right) \right)$$

Graph-level representation h_G is the average of node embeddings at the final layer, used for label prediction:

$$h_G = \mathrm{MEAN}(\{h_v^{(K)} | v \in G\})$$

This approach allows for comprehensive feature integration and graph-level inference.

4.4.5 SCENE

SCENE (SCene Encoding NEtwork), is introduced in 2023 [31] and is an innovative methodology for encoding and reasoning about traffic scenes. This approach is centered around the use of a heterogeneous graph that effectively models various aspects of traffic scenes, including different node types and relation types. The methodology combines a generic Graph Neural Network (GNN) architecture, which employs cascaded layers of graph convolution, with a task-specific decoder to predict relevant information about the scene.

In the GNN architecture of SCENE, the encoder plays a crucial role. It aggregates information from the traffic scene into node embeddings by utilizing multiple layers of graph convolution. This process is based on a modified version of the Graph Attention Network (GAT) operator, which is adapted to incorporate edge features. A significant aspect of this architecture is the HetEdgeGAT, a combination of EdgeGAT and the aggregation of embeddings across multiple relation types, which enhances the model's ability to process diverse information within the graph.

On the decoding side, SCENE uses a task-specific decoder based on a Multilayer Perceptron (MLP). This decoder is applied to the encodings of agent nodes for binary node classification tasks, making it highly effective in interpreting the complex data embedded in the traffic scene graphs.

In SCENE, the input to the system includes features representing dynamic agents over a duration of three seconds, along with an abstract representation of static infrastructure, like HD maps. These inputs are then encoded into a heterogeneous scene graph. The architecture is designed to avoid over-smoothing, a common issue in multilayer GNNs, by incorporating concatenated residual connections. For training, the model employs a binary cross-entropy loss function and utilizes the Adam optimizer.

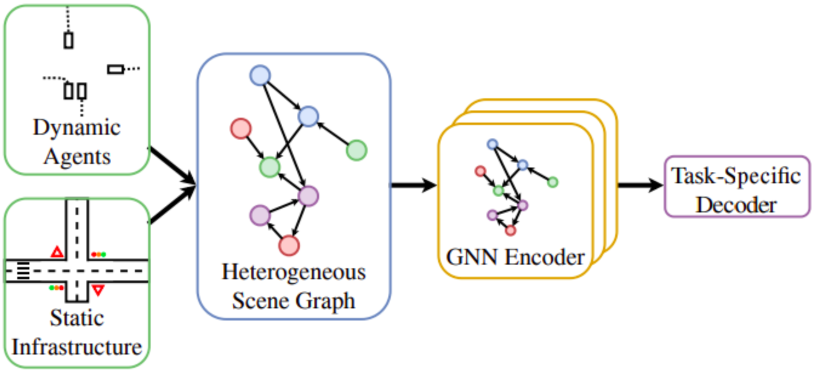


Figure 4.9: Overview of SCENE. Figure from [31].

In our study, we represent the frame-scene as an undirected heterogeneous graph with objects as nodes and relationships as edges. Let $G = (V, E, T, R, \varphi)$ represent the graph structure, where:

- V is the set of nodes, with each node $v_i \in V$ having a feature vector $\mathbf{v}_i$.
- E is the set of edges, where an edge $e_{j,r,i} = (v_j, r, v_i) \in E$ connects the source node v_j to the destination node v_i with relation type $r \in R$, and has a feature vector $\mathbf{e}_{j,r,i}$.
- T is the set of allowed node types.
- The type operator $\varphi : V \to T$ defines the type of each node v.

4.5 Hierarchical Conditional Neural Network & Answer Decoder

The final step in our approach is a hierarchical architecture conditioning on the previously extracted graph embeddings, to answer the question. Drawing inspiration from HCRN's architecture, seen in Figure 3.3, we adapt the model to integrate scene graphs, thus benefiting from the hierarchical and contextual processing of the CRN units.

Motivated by the Hierarchical Conditional Relation Network(HCRN), we propose a novel architecture that integrates scene graph generation (SGG) and a Graph Neural Network (GNN) to process and reason over video data. After the scene graph generation and post-processing, GNNs extract frame-level graph embeddings incorporating the relationships and interactions between objects in the scene graphs. These features are then fed into our Hierarchical Architecture.

The core of our architecture consists of multiple CRN units arranged hierarchically. The CRN units at the lower level process data at the clip level, handling more granular information, while CRN units at the higher level operate at video level, gathering information from multiple clips. The hierarchical design enables the model to consider information in different contexts. The top level CRN unit outputs a video graph embedding, used to classify the answer. This video-graph embedding is then aggregated with the question embeddings and the final feature is processed by an answer decoder that generates the final output.

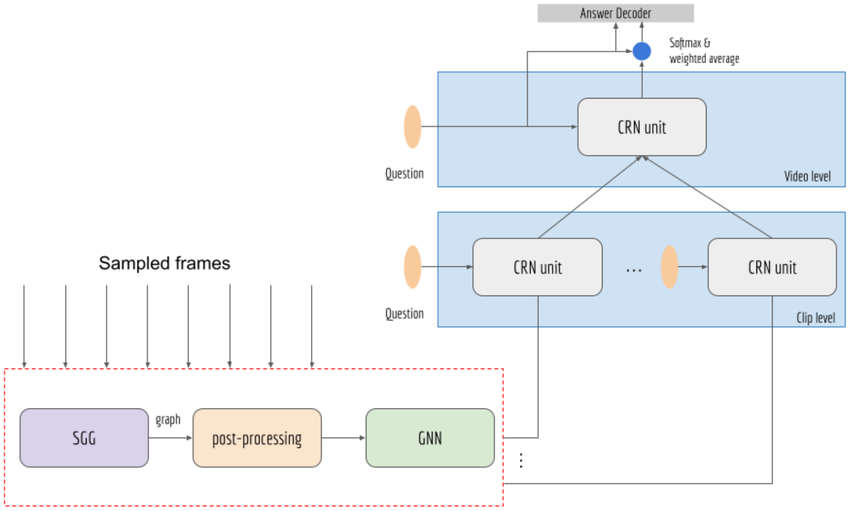


Figure 4.10: Temporal approach model with predicted scene graphs architecture.

By integrating HCRN with scene graphs, the model can exploit this contextual information to better understand the semantics and spatial arrangements of objects within scenes, leading to more accurate scene understanding and interpretation.

4.6 Training Process

Our architecture is trained end-to-end with pre-generated scene graphs. During training, out model sequentially processes batches of data, each comprising of graph representations of videos, class labels, and encoded answers among other elements. The GNN component first processes the graph data to produce embeddings, which are then utilized by the HCRN model along with the questions to predict answers.

Optimization is performed using CrossEntropyLoss and the AdamW optimizer, with accuracy and loss metrics for training sessions logged via Weights & Biases (WandB) [2], facilitating real-time monitoring and analysis as seen in Figure 4.11.

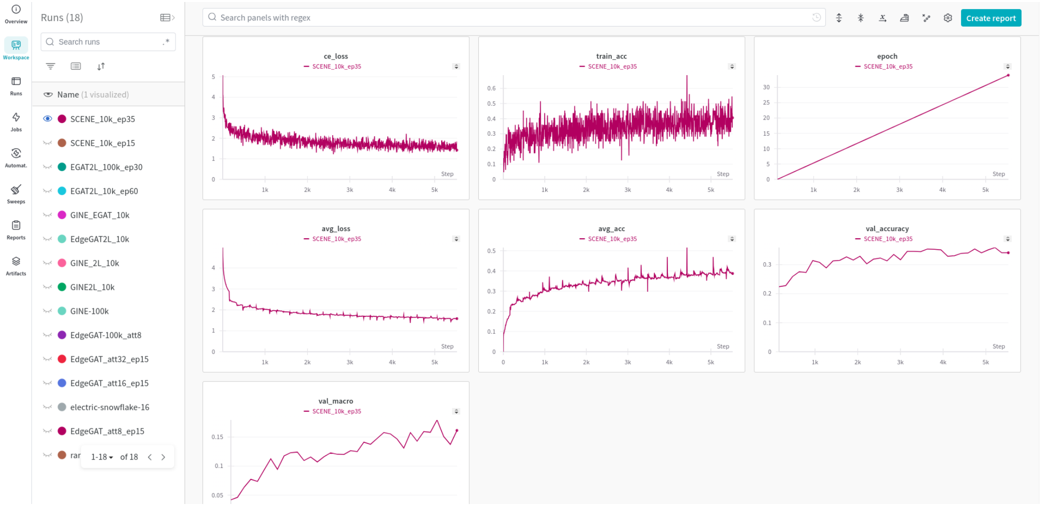


Figure 4.11: Weights & Biases Experiment Dashboard.

Validation phases in between training epochs enable the evaluation of the model's performance on unseen data. Saving checkpoints and the best model based on validation accuracy ensures that progress is retained and that the most effective model configuration is saved.

4.7 Challenges and Limitations

4.7.1 Scene Graphs Accuracy

The scene graphs generated from the video frames are essential to its overall performance. This part of the pipeline is constrained due to the use of MOTIFS, which is an older pre-trained model. The use of MOTIFS can lead to less efficient and accurate scene graphs extraction.

Additionally, the scene graph extraction's effectiveness is further limited because of the lack of fine-tuning on Action Genome Question Answering. Without this step, the model's parameters remain optimized for the generic data distribution of the original training set of another dataset. This way, the model may not be subjective to the unique characteristics of AGQA potentially resulting in less precise scene graphs;

In summary, both the architecture of the pre-trained model and the absence of fine-tuning present limitations to our method. However, we can easily address these issues by using a state-of-the-art methodology instead of MOTIFS with little to no change in our methodology and also implementing a fine-tuning phase tailored to AGQA.

4.7.2 Dataset size

Comprising nearly 3 million question-answer pairs, with around 2 million allocated for the training set, the dataset presents a considerable obstacle in terms of computational demands for model training. To tackle the impracticality of training a model on the entire dataset, we employ a strategy of training on smaller subsamples. These subsamples are carefully curated to maintain the original dataset distributions, ensuring that the model is exposed to a representative mix of data during training. So, this computational bottleneck imposes a significant constraint on the iterative process of model development and evaluation.

4.7.3 Generalization

The proficiency of the model in generalizing beyond the AGQA dataset is indeed an area of concern given the dataset's specialized nature. The AGQA dataset is meticulously constructed to represent daily activities, with its videos curated to encapsulate a spectrum of commonplace scenarios. The questions within this dataset are systematically generated from the scene graphs and actions using predefined scripts.

When considering the generalization of this model to other datasets, one must acknowledge that the divergence in question generation algorithms can pose a significant challenge. If the algorithm used to create question-answer pairs in a new dataset diverges from the one used in AGQA, the model might struggle to perform with the same level of accuracy. The reason lies in the model's potential overfitting to the patterns and distributions present in AGQA's questions, which are inherently tied to the scripts used for their creation.

4.8 Method overview

In conclusion, our architecture involves the following steps, as also seen in Figure 4.1:

- **Frame Sampling**: selecting clips and frames from video
- **Scene Graph Generation (SGG)**: extracting scene graphs from frames, depicting objects and their relationships
- **Scene Graphs Post-Processing**: filtering objects and triplets.

- **Clip-Level Features**: CRN neurons process sequences of frames to understand temporal and relational dynamics

- **Video-level Features**: CRN units process clip-level features to understand the whole video.

- **Answer Classification**: Combines the video-level understanding with the language query to predict the answer

5 Experiments, Results & Discussion

In this chapter, we detail the experiments conducted on AGQA, a large-scale Video Question Answering dataset, and analyze the results. Our investigation begins with an overview of the AGQA dataset, including its scale, diversity, and specific challenges it poses for Video QA tasks. This section provides context for understanding the dataset's complexity and the rationale behind our experimental design. Following the dataset introduction, we outline the implementation details of our approach. This includes the computational framework, hardware specifications, and considerations made to address the dataset's challenges. We then describe the baseline models developed to benchmark our methodology. These baselines are structured to incrementally introduce complexity and assess the impact of different data modalities and structures on performance:

- **Language Bias Baseline**: Focuses on the dataset's linguistic aspects, ignoring visual information to evaluate performance based solely on textual input.
- **Language-Vision Baseline**:Integrates visual data with textual queries to examine the improvement over language-only models, still without temporal analysis.
- **Language and Scene Graphs Baselines**: This includes two models, one utilizing ground truth scene graphs and another with predicted scene graphs, to explore the benefits of structured semantic information on performance, without temporal modeling.

Building on these baselines, we introduce our main contribution: a hierarchical conditional architecture that incorporates temporal modeling to better capture the dynamics of scene graph sequences in relation to the questions asked. This approach is designed to overcome the limitations identified in the baseline models and improve accuracy by exploiting the temporal dimension. Our analysis includes a quantitative comparison of our model against the baselines and state-of-the-art methodologies, using standard metrics to position our results within the context of existing research. The chapter concludes with a summary of our experimental insights, emphasizing the significance of temporal modeling in video QA tasks and suggesting directions for future research.

5.1 Action Genome Question Answering (AGQA)

We evaluate the proposed approach on a challenging video question answering benchmark, Action Genome Question Answering (AGQA). Action Genome Question Answering is a benchmark of 3.9M balanced and 192M unbalanced question-answer pairs associated with 9.6K videos, each 30 seconds in length.

The Action Genome Question Answering Dataset is an extension of Action Genome, an action recognition benchmark built upon Charades[55]. Charades is composed of videos of daily indoor activities, collected through Amazon Mechanical Turk, which includes 267 different users. Action Genome is built on top of Charades and offers annotated frames, uniformly sampled across the activities of each clip. We can see an example of an AGQA video in Figure 5.6. Action Genome decomposes actions into spatio-temporal scene-graphs, capturing the objects and how their relationships evolve as the actions progress.

Scene graphs are a formal representation of image information in the form of a graph. Each scene graph encodes objects as nodes, connected together by pairwise relationships as edges. Each action in Action Genome is represented as changes to objects and their pairwise interactions with the person performing the action. The representation can be viewed as a temporally changing version of Visual Genome scene-graphs, but instead of densely representing the objects in the scene, it aims to decompose actions by annotating only those segments of videos that involves an activity that can be decomposed.

Action Genome Question Answering provides frame-level scene graph labels for the components of each action. Overall, there are more than 234k frames annotated, with more than 476k bounding boxes, 35 object classes, and more than 1.7M instances of 25 relationship classes, as seen in Table 5.1 and Table 5.2. Even if some objects and relationships occur more frequently than others, almost all objects have at least 10k instances, and every relationship has at least 1k instances.

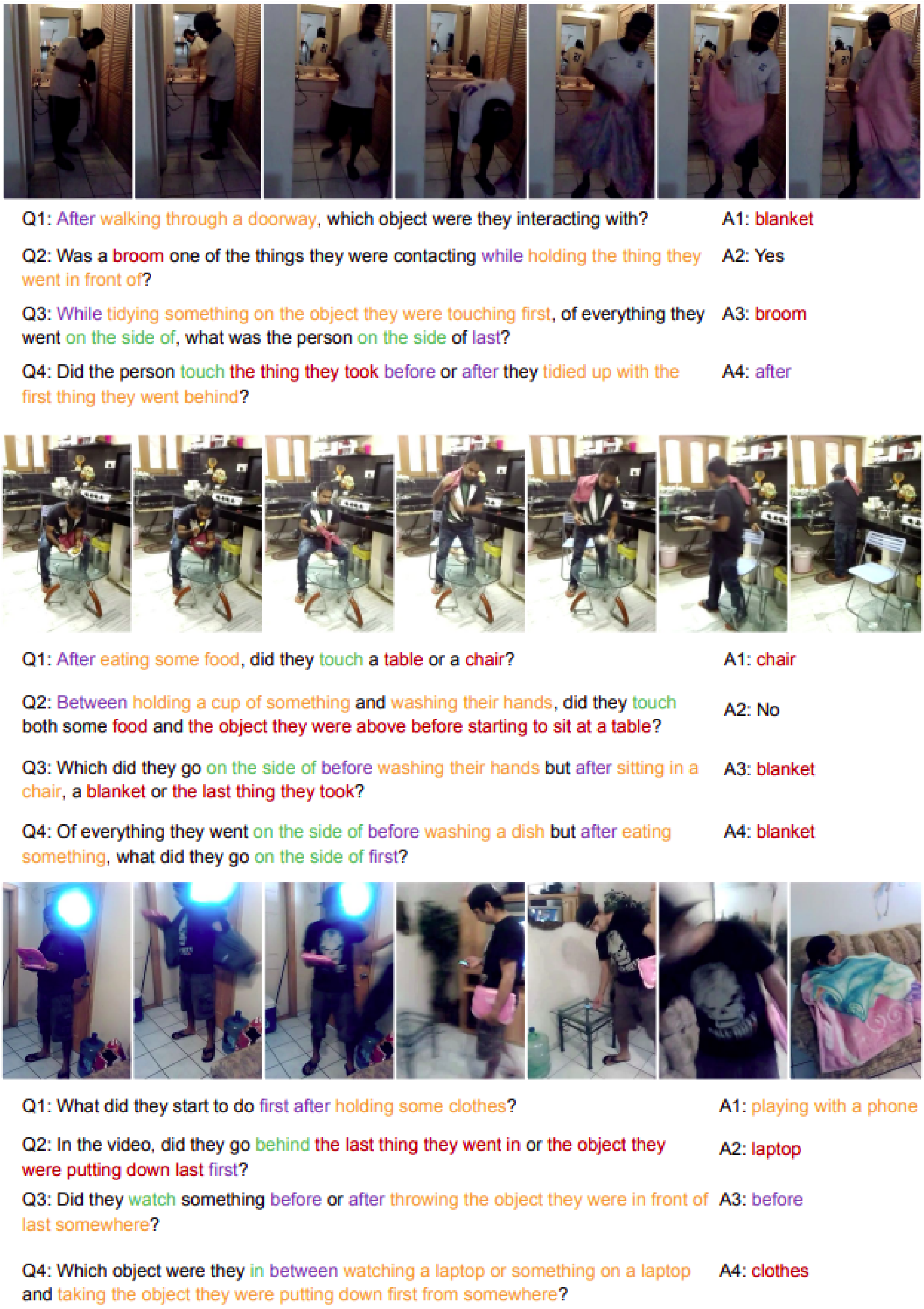
Window	Bag	Bed	Blanket	Book	Box
Broom	Chair	Closet/Cabinet	Clothes	Cup/Glass/Bottle	Dish
Door	Doorknob	Doorway	Floor	Food	Groceries
Laptop	Light	Medicine	Mirror	Paper/Notebook	Phone/Camera
Picture	Pillow	Refrigerator	Sandwich	Shelf	Shoe
Sofa/Couch	Table	Television	Towel	Vacuum	

Table 5.1: AGQA Objects Types.

Attention	Spatial	Contact
looking at	in front of	carrying
not looking at	behind	covered by
unsure	on the side of	drinking from
	above	eating
	leaning on	have it on the back
	lying on	holding
	beneath	
	not contacting	
	sitting on	
	in	
	standing on	
	touching	
	twisting	
	wearing	
	wiping	
	writing on	

Table 5.2: AGQA Relationships Types.

The questions in Action Genome Question Answering are generated using templates and scene graph information to create a diverse set of questions. Questions are categorized as reasoning primarily about an object, relationship, or action. In Figure 5.1 we can see some examples of Action Genome Question Answering question-answer pairs, along with some sampled frames.



Q1: After walking through a doorway, which object were they interacting with? A1: blanket

Q2: Was a broom one of the things they were contacting while holding the thing they went in front of? A2: Yes

Q3: While tidying something on the object they were touching first, of everything they went on the side of, what was the person on the side of last? A3: broom

Q4: Did the person touch the thing they took before or after they tidied up with the first thing they went behind? A4: after

Q1: After eating some food, did they touch a table or a chair? A1: chair

Q2: Between holding a cup of something and washing their hands, did they touch both some food and the object they were above before starting to sit at a table? A2: No

Q3: Which did they go on the side of before washing their hands but after sitting in a chair, a blanket or the last thing they took? A3: blanket

Q4: Of everything they went on the side of before washing a dish but after eating something, what did they go on the side of first? A4: blanket

Q1: What did they start to do first after holding some clothes? A1: playing with a phone

Q2: In the video, did they go behind the last thing they went in or the object they were putting down last first? A2: laptop

Q3: Did they watch something before or after throwing the object they were in front of last somewhere? A3: before

Q4: Which object were they in between watching a laptop or something on a laptop and taking the object they were putting down first from somewhere? A4: clothes

Figure 5.1: Examples of AGQA Questions. Figure from [14].

An interesting experiment was the human study, where humans were used to validate the correctness of AGQA's questions. Two tasks were run, one where humans had to verify given answers and one where they had to select the answer from a drop-down list, as seen in Table 5.3. This represents an upper bound of accuracy for the Video Question Answering task on AGQA.

Question Types			Verification (%)	Dropdown (%)
Reasoning	obj-rel	B	78.95	68.42
		O	90.90	63.64
		All	80.65	67.74
	rel-action	B	90.20	78.43
	obj-act	B	93.75	83.33
	superlative	B	81.81	72.73
		O	80.77	55.77
		All	81.25	63.54
	sequencing	B	94.73	78.94
		O	85.18	59.26
		All	90.77	70.77
	exists	B	79.80	74.03
	duration	B	91.89	70.27
		O	92.31	69.23
		All	92.00	70.00
	activity recognition	O	78.00	54.00
Semantic	object	B	87.39	74.19
		O	90.90	60.52
		All	87.97	72.93
	relationship	B	83.58	75.37
	action	B	90.21	73.91
		O	80.95	57.14
		All	86.45	67.10
Structure	query	O	83.53	58.82
	compare	B	92.53	78.16
	choose	B	83.02	66.04
	logic	B	70.69	70.69
	verify	B	88.26	76.93
Overall		B	86.65	73.85
		O	83.53	57.93
		All	86.02	71.56

Table 5.3: This table presents the human performance on two tasks per question category. On the first one they had to verify given answers (Verification) and on the second they had to select the correct answer from a dropdown list. For each question type, see can see their performance on binary questions, B, open-ended questions, O and all.

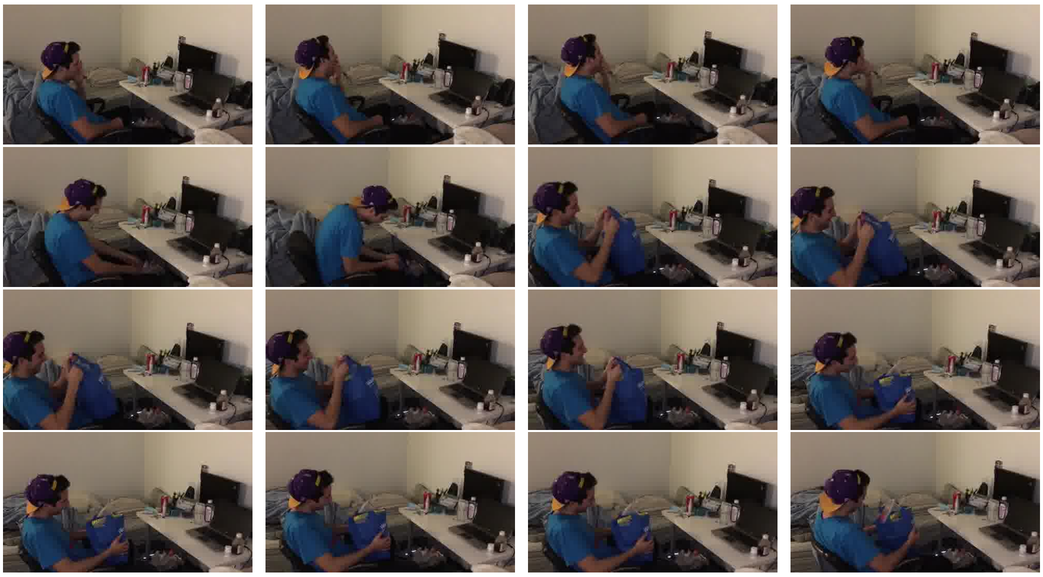


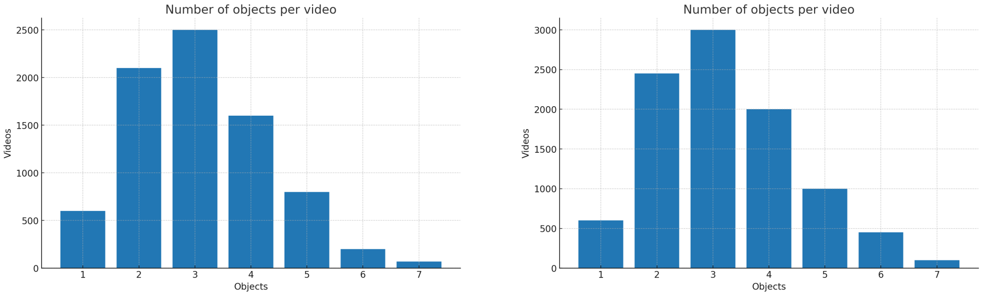
Figure 5.2: Example of an AGQA video, depicting a young man doing daily activities, like a man sitting at his desk and picking up a bag.

5.1.1 Dataset Statistics

The dataset comprises approximately 9.6 thousand videos, each with a duration of 30 seconds and recorded at a frame rate of 30 frames per second (fps). This translates to an average of around 900 frames per video. However, a notable aspect of AGQA is the selective annotation process applied to these videos. Despite the large number of frames available per video, on average, only 35 frames in each video are annotated. This disparity highlights the focused nature of the annotations, where only a fraction of the total frames are chosen for detailed labeling. This approach underscores the dataset's emphasis on specific, salient moments within the videos, rather than a comprehensive frame-by-frame annotation.

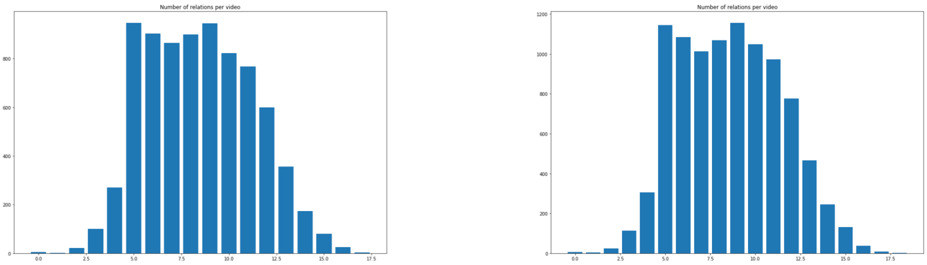
We can measure the average annotated relations per video, observing a bell-shaped distribution with an average of around 7.5 to 10 relations per video in both the training and test sets, as seen in Figure 5.4. This indicates a common complexity level within the dataset where most videos contain a similar range of relations. There's a notable decrease in frequency as the number of relations increases, suggesting that fewer videos have a very high complexity in terms of relations depicted.

About the average annotated objects per video, as seen in Figure 5.4 the distribution is somewhat left-skewed, with a peak at 4 objects per video in both training and test datasets. It shows that a majority of the videos feature a modest number of distinct objects, with the number of videos rapidly declining as the number of



(a) Train set **(b)** Test set

Figure 5.3: Average objects annotated per video.



(a) Train set **(b)** Test set

Figure 5.4: Average relations annotated per video.

objects increases. The consistency of this pattern across training and test sets suggests that the dataset is well-balanced in terms of object variety per video.

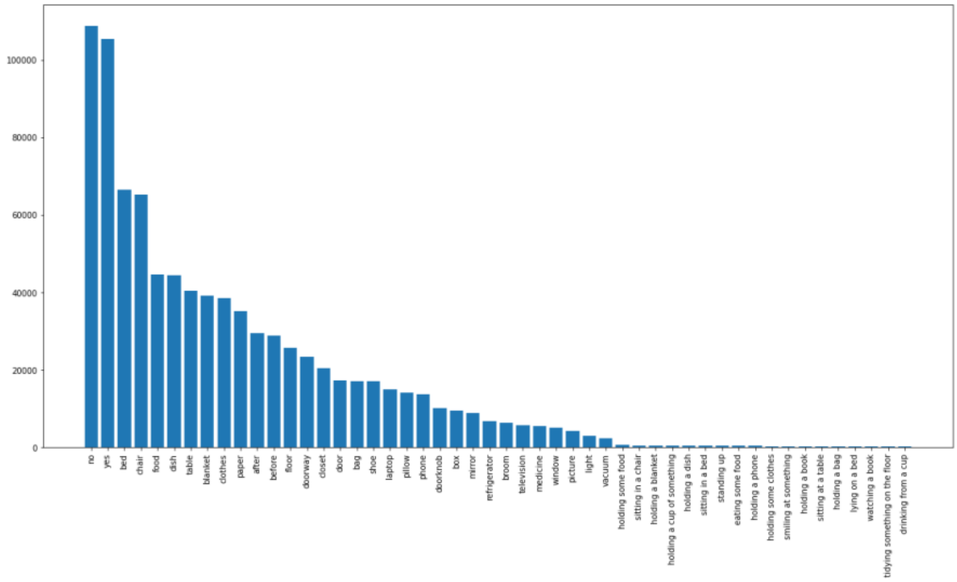


Figure 5.5: AGQA Answer Distribution.

As we can see in Figure 5.5, the distribution follows a steep descending order, with the most frequent answer being significantly more common than the rest. This first category towers over the others, indicating a heavily skewed distribution where one or a few answers dominate the dataset. The two most common answers with almost equal shares are ”yes” and ”no”.

As we move from left to right along the x-axis, there is a rapid decline in the frequency of each successive answer, demonstrating a long-tail effect. This suggests that while there are a handful of very common answers, there is also a wide variety of less common ones. This type of distribution is typical in natural language datasets, where a small number of words or phrases are extremely common (following Zipf's law), and there is a long tail of rare words or phrases.

This distribution could imply that any model trained on this dataset might become biased towards the most common answers, and thus, special consideration might need to be given to ensure that the model does not simply learn to always predict the most frequent categories. Techniques such as re-sampling, re-weighting, or using sophisticated loss functions might be necessary to counteract this imbalance and encourage the model to learn a more generalizable understanding of the data.

AGQA authors run three state of the art models on their benchmark (HCRN, HME, and PSAC), and find that the models struggle on the benchmark. If the model only chose the most likely answer (“No”) it would achieve a 10.35% accuracy. The highest scoring model, HME, achieved a 47.74% accuracy, which at first glance appears to be a big improvement. However, further investigation found that much of the gain in accuracy comes from just exploiting linguistic biases instead of from visual reasoning. Although HCRN achieved 47.42% accuracy overall, it still achieved a 47% accuracy without seeing the videos. The fact that the model is so dependent on linguistic biases instead of visual reasoning reduces the ability of our other test splits to effectively measure visual reasoning for these particular models.

5.2 Implementation Details

5.2.1 Framework

In our experimental setup, we utilized PyTorch [33] as the main framework for all model training and development. For graph-related tasks, especially in Graph Neural Networks (GNNs), we employed the Deep Graph Library (DGL)[49], which is compatible with PyTorch and provides optimized graph data structures and operations.

5.2.2 Operating Environment

Our computational resources were divided between two machines, each equipped with four GPUs. The first machine contained four GeForce GTX 1080 Ti GPUs and was used for preprocessing tasks, including motion and appearance feature extraction, CLS feature generation, and scene graph extraction. The second machine, equipped with four NVIDIA GeForce RTX 3090 GPUs, was dedicated to running the experiments. This setup allowed for efficient processing and task distribution, leveraging the strengths of each machine and GPU type for their specific roles in the experimental pipeline.

5.2.3 Train-Test Sets split

We train and test our model on the large-scale AGQA dataset, which comprises a total of 3 million questions, segmented into 1.8 million for training and 1.2 million for testing. To accommodate various computational capacities and to facilitate detailed analysis, we designed two distinct experimental frameworks: a tiny setup and a small setup. The tiny setup includes a subset of 10,000 training samples and 2,000 test samples, while the small setup expands this to 100,000 training samples and 20,000 test samples. These subsets were carefully curated from the original training set, ensuring that the questions in the train and test sets correspond to entirely different videos to avoid any potential data leakage and to closely mimic real-world application scenarios. Additionally, we employed random sampling techniques to maintain the original dataset's distribution, thereby ensuring that our experimental setups accurately reflect the diversity and complexity of the AGQA dataset.

Train - Test Splits

Split	Train Size	Test Size
tiny	10.000	2.000
small	100.000	20.000

Table 5.4: Train - Test Splits on AGQA Benchmark.

5.2.4 Frame Extraction

A crucial part of our preprocessing involved the extraction of individual frames from video data, a process we accomplished using a modified script based on the Action Genome framework. This script, written in Python, automates the frame dumping process from video files, tailored to our specific dataset and annotation requirements.

Using FFMPEG [44], we create a mapping of video files to the corresponding frames to organize the grame extraction process. It's important to note that the frames are extracted at their original video frames per second (FPS), which may not always be a standard rate like 24 FPS. This means that the frame indices may differ from other datasets, such as the Charades dataset. After extracting the frames, the script optionally deletes frames not listed, keeping only the annotated frames. This script handles large datasets efficiently and is flexible to different project requirements, making it a vital tool in our preprocessing pipeline. The use of ffmpeg ensures high compatibility with various video formats and efficient processing. Moreover, the script's ability to selectively extract frames based on annotations significantly reduces unnecessary data storage and computational overhead.

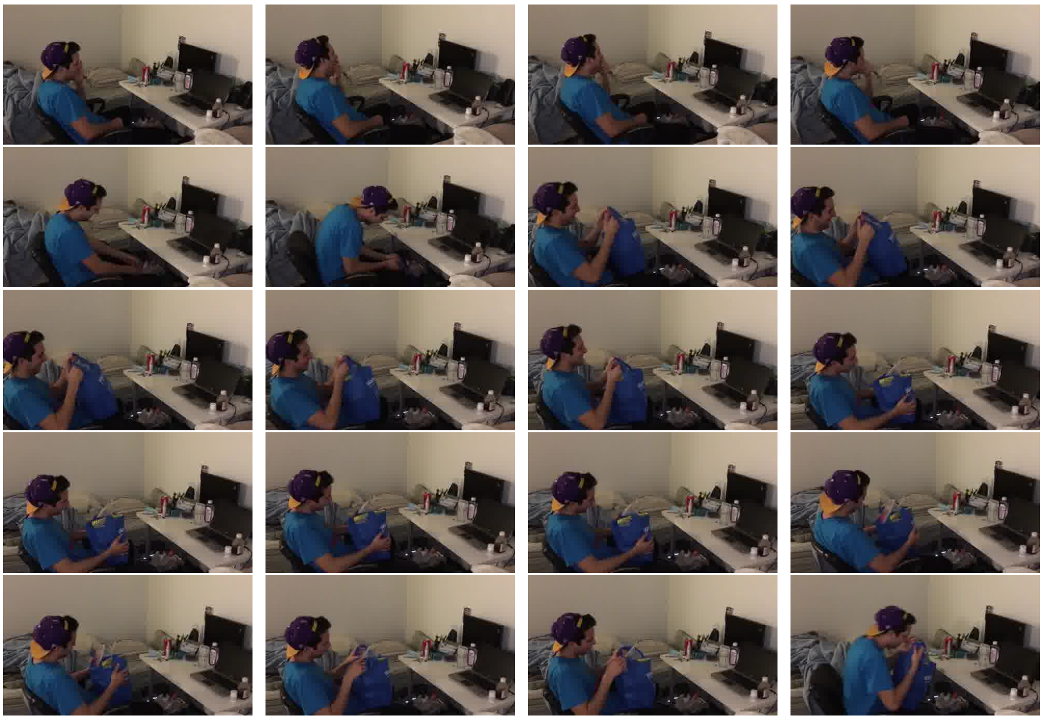


Figure 5.6: Example of an AGQA video after the annotated frames extraction. The frames are annotated as following: For each annotated action, 5 frames are selected uniformly across the action and are annotated.

In AGQA, questions are engineered using specific templates, resulting in some 'Good' and some 'Tricky' questions. Good questions can be better phrased, more direct, and easier to understand, while tricky questions can have ambiguous answers, can be very difficult to interpret, or have a vague meaning. Below we can see an example of questions for the video in Figure 5.6:

'Tricky' questions:
- What did they take while sitting in the thing they went above? **bag**
- In the video, what was the person on the side of? **chair**
- What was the person above while standing up? **chair**
- In the video, was a chair the last thing they held? **no**
- Which object were they in front of? **chair**
- Did they touch a bag but not the thing they tidied after taking the thing they held from somewhere?

'Good' questions:
- Which object did the person take while sitting at a table? **bag**
- In the video, which object were they taking? **bag**

- Before standing up but after taking a bag from somewhere, which object were they above? **chair**
- Did they hold a bag or sit in a chair for longer? **sit in a chair**
- Which object were they above? **chair**
- After taking the object they were holding from somewhere, did they interact with a chair? **yes**
- Did the person hold a bag for a shorter amount of time than they spent laughing at something? **yes**
- Was the person holding a bag or laughing at something for a shorter amount of time? **holding a bag**

5.3 Non-Temporal Baseline Models

In this section we will discuss the baselines we built on our way to our final methodology and discuss their performance. We experiment with a language only experiment, a non temporal baseline with vision and two non-temporal baselines with scene graphs, one with ground truth scene graphs and the other one with predicted ones.

5.3.1 Language Bias Experiment

In our first baseline experiment, we focused on the simplest approach for answer classification using only the question embeddings, trying to measure the language bias. We utilized [CLS] token embeddings derived from BERT, a former state-of-the-art language model, as the primary feature representation for our textual input. The choice of BERT was motivated by its proven capability in capturing deep contextual relationships within text, making it ideal for understanding the nuances in the questions and answers.

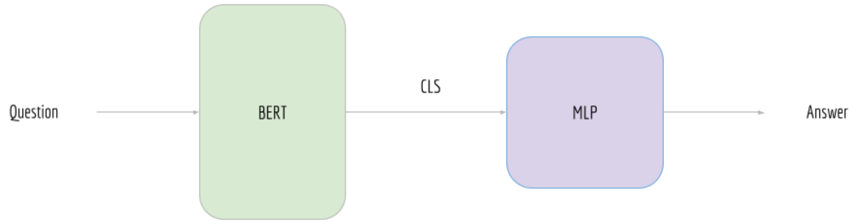


Figure 5.7: Language bias baseline architecture for question classification.

The process began with feeding the textual data into the BERT model to obtain the CLS embeddings. In BERT's architecture, the CLS token is a special token added at the beginning of each input sequence, and its corresponding embedding in the output layer is designed to capture the overall context of the sequence. This makes the CLS embedding a comprehensive representation of the entire input text, encapsulating its semantic essence.

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-

Table 5.5: Modalities & Capabilities of Models.

Once we obtained the CLS embeddings, we used them as input to a Multi-Layer Perceptron (MLP). The MLP in our experiment was a simple feedforward neural network with fully connected layers. It was responsible for mapping the high-dimensional CLS embeddings to the space of our target classes. The MLP's architecture was kept relatively simple, with a few hidden layers, to provide a baseline understanding of how well the CLS embeddings from BERT could perform in a classification task with minimal additional complexity.

The objective of this baseline experiment was to classify the answers into predefined categories. The combination of BERT's sophisticated language understanding and the MLP's classification capability aimed to set a foundational performance benchmark. This would then serve as a point of comparison for more complex models or approaches explored in subsequent experiments.

In this baseline experiment, we sought to assess the language bias on AGQA using uniformly sampled datasets of 10k and 100k samples, meaning question-

Experiments	Accuracy
10k samples	21.5%
100k samples	34.1%

Table 5.6: Language Bias Results on AGQA.

answer pairs. Uniform sampling ensures each data point from the dataset has an identical probability of selection, thus eliminating sampling bias and reflecting the true distribution of the dataset.

According to the dataset authors, if the model were to naively predict the most frequent category, "No", it would stand at an accuracy of 10.35%. When analyzing the results under this light, the Language Bias model achieved 21.5% accuracy with 10k samples, which is more than double accuracy. This indicates that the model has learned to identify linguistic patterns between questions and answers that go beyond mere guesswork.

Upon expanding the dataset to 100k samples, the baseline model's accuracy improved to 34.1%. This is a substantial increase not only over the baseline but also over the smaller sample size, demonstrating the model's enhanced ability to generalize from a larger dataset. With more data, the model can discern between different types of questions, rather than leaning on a one-size-fits-all approach. It's noteworthy that the increased accuracy is also comparable to accuracy that would result from always predicting the most common answer. For object-related questions, the accuracy for predicting the most common answer would result with 9.38% accuraacy, the relationship-based questions with 50%, and the action-related questions with 32.91% .

That being said, through this experiment, we have assessed the existing semantic connections between the question and answer pairs. As also demonstrated in the AGQA paper, there is a strong linguistic bias that state-of-the-art models can exploit to score up to 47% accuracy.

5.3.2 Non-Temporal Video-Language Baseline

As a next step from the previous baseline, we wanted to examine how well a non-temporal model could use the combination of language and video data.

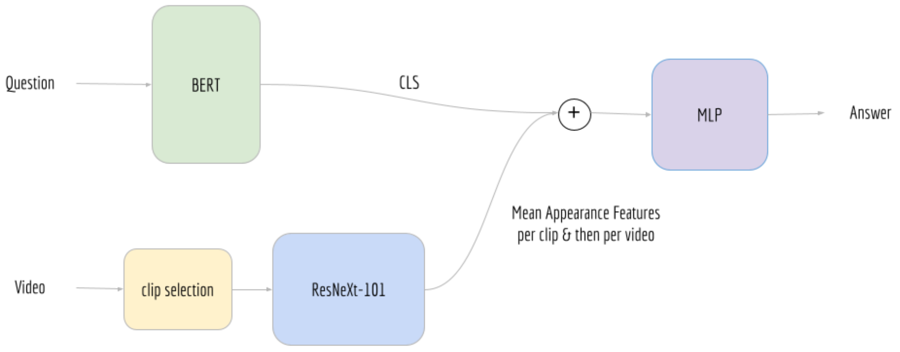


Figure 5.8: Non-temporal language & video architecture.

Specifically, the experiment investigated how the integration of appearance embeddings, extracted from video data, would enhance the model's performance compared to a language-only baseline. For this purpose, two distinct sets of embeddings were used:

- **Question BERT CLS Embeddings**: These embeddings were generated by using BERT (Bidirectional Encoder Representations from Transformers), a pre-trained transformer-based model well-known for its effectiveness in encoding a wide range of language representations. The [CLS] token embeddings, which are designed to capture the context of the entire input sequence, were extracted to represent the questions.
- **Mean Appearance Features of 8 Clips**: In addition to the language embeddings, appearance features from the video were included. These features were obtained by uniformly sampling 8 clips across the length of the video, extracting their appearance features, and then taking the mean. This process involved using a convolutional neural network pre-trained on image data to capture visual features from each frame. The procedure for clip sampling, feature extraction, and processing was consistent with the methods used in HCRN

A Multilayer Perceptron (MLP), a type of feedforward artificial neural network, was then trained to classify the answers using these combined embeddings. The MLP took as input the concatenated question and video embeddings and learned to map this high-dimensional input to the correct answers.

The results from the experiment showed an improvement over the language-only model. With 10k samples, the language-only model (referred to as 'lang')

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-
Vid_Lang_MLP	✓	✓	-	-

Table 5.7: Modalities & Capabilities of Models.

Experiments	lang	lang+vid
10k samples	21.5%	26.1%
100k samples	24.1%	25.8%

Table 5.8: Accuracy of MLP model with BERT CLS embeddings and mean appearance features.

achieved 21.5% accuracy, while the model combining language and video (referred to as 'lang+vid') achieved 26.1% accuracy. Similarly, with 100k samples, the 'lang' model had an accuracy of 24.1%, while the 'lang+vid' model reached 25.8% accuracy. These results suggest that the additional context provided by the appearance features from the video data contributes positively to the model's ability to answer AGQA questions more accurately.

question-type	accuracy
exists	32.8%
obj-rel	18.5%
obj-act	24%
sequencing	24.62%
duration-comparison	12%
rel-act	22%
action-recognition	0%

Table 5.9: Non-temporal language & video results per question type.

We can measure the qualitative performance of the baseline model in the bar graph of Table 5.9. Taking a quick look at the table, we can easily infer that the categories the baseline model falls short of are the ones with a temporal aspect, like 'action_recognition'. The MLP used to classify the answers cannot process temporal sequences, crucial to action recognition, thus failing at some relevant categories. However, the MLP can sometimes recognize static features from the video frames appearance features, identifying objects and answering the 'exists' and 'obj' related questions. So, even if the baseline model can answer some questions based on static image feature extraction, it is not designed to process time-dependent information, so it lacks a lot in temporal understanding. Another weakness of this baseline model lies in the 'obj-rel' category, which also requires higher reasoning and comprehension levels.

5.3.3 Upper bound - Non-temporal baseline model

Drawing motivation from the results of the previous baseline model, our approach focuses on providing better scene comprehension and video understanding through scene graphs. As seen in Figure 4.1, after sampling clips and frames, we extract scene graphs, process them with a Graph Neural Network to get graph embeddings for each frame, and use those embeddings to classify the answer for each question. Our hypothesis is that using a richer representation of the video content through scene graphs can lead to improved accuracy in Video Question Answering.

Building upon the insights of the initial baseline model, our proposed method aims to enhance scene understanding by incorporating scene graphs. Our proof of concept involved the extraction of ground truth scene graphs from individual frames within a video. These scene graphs are structured representations that encapsulate the objects and relationships within a scene. By using these graphs, we want to prove that we can capture a snapshot of the scene's composition that provides more contextual clues than the raw pixel data alone. So, our goal is to see if the ground truth scene graphs can leed to better accuracy than using the visual appearance features.

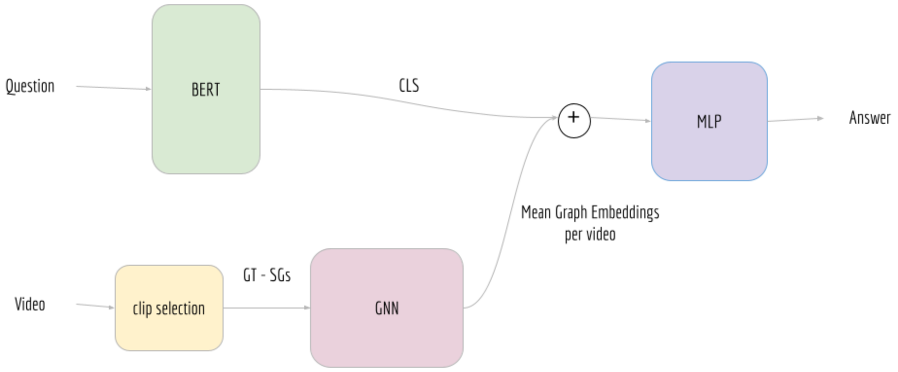


Figure 5.9: Non-temporal proof of concept architecture.

As seen in Figure 5.9, our process consists of several steps. We first extract contextual embeddings of the question from a pre-trained large language model, BERT. The question embeddings are the 'CLS' token embeddings, that are typically used in classification tasks.

We then proceed to clips and frame sampling. The HCRN clip sampling strategy wouldn't be possible since it samples 8 clips of 16 sequential frames each. The AGQA annotation process included annotating 5 frames per action in videos, uniformly spread across the action duration. Since the majority of videos had more than 15 annotated frames and 35 on average, we chose 5 clips of 3 frames each. For the clip separation, we split the video into equal parts and then sampled as uniformly as we could across the annotated frames. We hope that this sampling strategy will enable our graph embeddings to capture the video's diversity.

After selecting the frames for each video, we form the ground truth graphs, using the dataset annotations. We then use a GNN to extract features from the scene graphs. We use a simple Graph Attention Model with edge weights to capture the relationships and interactions between objects within a scene. This model updates the nodes features of each node by aggregating the neighboring nodes and edges features. In this way, each frame is encoded into a graph embedding that encapsulates its content in a dense, informative vector.

Model	**Language**	**Vision**	**Scene Graphs**	**Temporal Processing**
Lang_MLP	✓	-	-	-
Vid_Lang_MLP	✓	✓	-	-
PoC	✓	✓	✓	-

Table 5.10: Modalities & Capabilities of Models.

To classify the answers we use a simple feedforward MLP with few hidden layers. An MLP, by its nature, is not capable of handling temporal information, a significant aspect of video content. To tackle this problem, we use a video-graph embedding strategy.We take the mean of the graph embeddings from all the frames within a video clip to form a clip-level embedding. This process aggregates the information from multiple frames, providing a temporal aspect to the otherwise static embeddings. We then further condense this information by taking the mean of all such clip-level embeddings to produce a whole video graph embedding.This final embedding represents the entire video's content and serves as input to the MLP for the final classification of the answer.

Experiments	**Lang_MLP**	**Vid_Lang_MLP**	**PoC**
10k samples	21.5%	26.1%	49.1%

Table 5.11: Experimental results showing the effect of adding ground truth scene graphs to language features.

To test this architecture, we conduct an experiment with 10k samples. We can detect a significant improvement compared to the baseline models, reporting accuracy of 49.1%, as also seen in Table 5.11. From that we can infer that scene graphs convert visual data into a structured format that can be more effectively utilized by an MLP. Also, while MLPs inherently lack the ability to process temporal information, the mean of graph embeddings from clips can offer an approximation of temporal dynamics by capturing changes in the scene graph over time. The structured nature of scene graphs means that they can represent complex scenes with relatively little data compared to raw pixels, leading to a significant accuracy boost.

Question type	Accuracy
superlative	22.6&
obj - rel	27.72%
exists	32.8%
obj-act	36%
sequencing	30.8%
duration-comparison	6%
rel-act	0.3%
action-recognition	0%

Table 5.12: Non-temporal proof of concept architecture results per answer category.

In Table 5.12, we can see significant improvement in almost all question categories. The most impressive one is the object-relation, object-action and exists category, where the accuracy has doubled due to the use of scene graphs, as expected.

5.4 Our final approach

As we have already discussed in Figure 4.1, our approach proposes the use of scene graphs to get more structured and contextual video features and lead to improved Video Question Answering performance.

5.4.1 Graph Extraction and Filtering

To compute the scene graphs, we use the Scene Graphs Benchmark and the pre-trained MOTIFS model. The process begins with the detection of objects within an image. These detected objects are then used to predict the relationships between each pair of objects. The MOTIFS model uses visual features from the objects and the spatial information between them to predict these relationships.

The model works by first detecting objects in the image using a pre-trained object detector. Once the objects are detected, the model computes features for each object and the pairwise spatial features. These features are then fed into the MOTIFS model, which includes a message-passing mechanism to refine the object features based on their context within the image.

Finally, the refined features are used to predict relationships by considering the types of objects detected and their visual features. The output is a graph structure that represents the objects as nodes and the predicted relationships as edges, forming the scene graph.

5.4.2 Scene Graphs post-processing

As a result of the above process, there are multiple objects detected and triplets generated. In order to keep useful triplets, we implemented a filtering process. Firstly, due to the large volume of detected objects, whereas only a handful of them are accurate, we only keep objects with confidence larger than 10%. We also remove duplicate object instances by calculating the intersection over union of the objects bounding boxes. Then, after resulting in several filtered detected objects, we select the triplets that only involve those. Then, we also filter the less confident ones, gathering the triplets for our final scene graph.

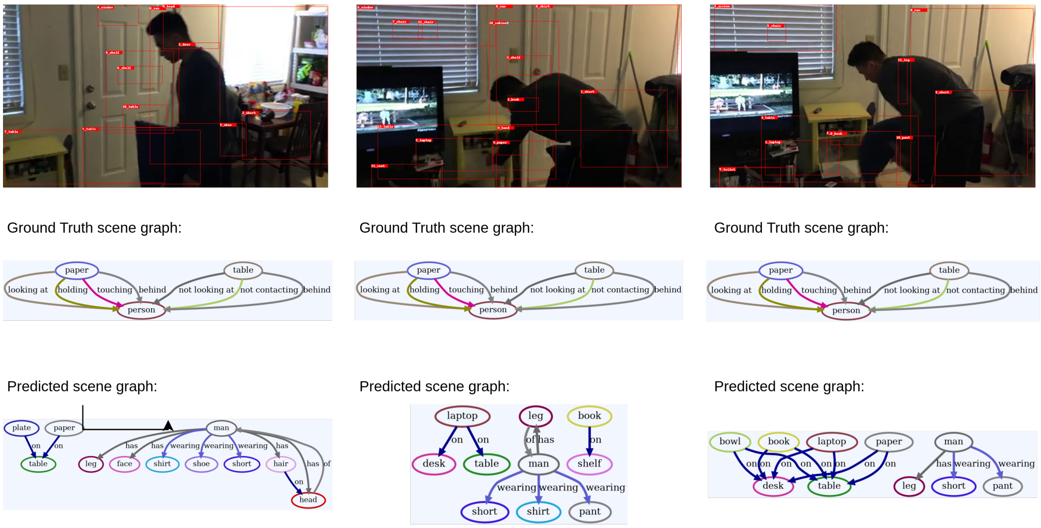


Figure 5.10: Example of scene graph extracted from video frames compared to the ground truth scene graphs.

5.4.3 Non-temporal Graph Model

Our first approach was to examine the proof-of-concept model's performance with predicted scene graphs instead of the ground truth ones.

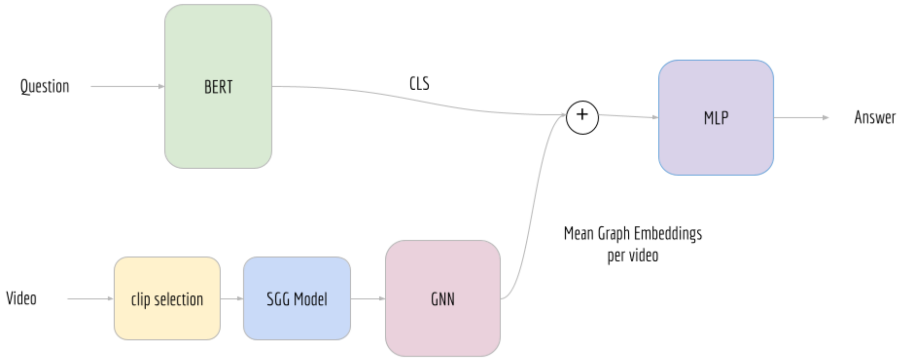


Figure 5.11: Non-temporal approach model with predicted scene graphs architecture.

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-
Vid_Lang_MLP	✓	✓	-	-
PoC	✓	✓	✓	-
SG-MLP	✓	✓	✓	-

Table 5.13: Modalities & Capabilities of Models.

Experiments	Lang_MLP	Vid_Lang_MLP	PoC	SG_MLP
10k samples	21.5%	26.1%	49.1%	31.6%

Table 5.14: Experimental results comparing the non-temporal scene graphs approach to baselines.

As seen in Figure 5.11, the architecture is very similar to the proof-of-concept one, except for the origin of the scene graphs. In summary, we sample clips and frames, extract scene graphs, process them with a Graph Neural Network to get graph embeddings for each frame, and use those embeddings to classify the answer to each question.

Question type	Accuracy
obj-rel	42.47%
exists	31.04%
obj-act	36%
sequencing	24.66%
superlative	19.5%
duration-comparison	12%
rel-act	24%
action-recognition	0%

Table 5.15: Performance per question type

As we can see in Table 5.15, there is a significant boost in accuracy in almost every category compared to the baselines. However, if we compare this model to the proof-of-concept model with the ground truth graphs, the latter is over 30% more accurate in the object-relation category. This was expected since MOTIFS is an older model, so the scene graphs may not capture the most accurate objects and their interrelations. Also, the AGQA questions have been engineered from the ground truth scene graphs, actions and captions, so there may be a much higher correlation between the ground truth graphs and the answer, so the proof-of-concept model performs a lot better.

5.4.4 Temporal Graph Model

Building on top of the previous models, we take motivation from HCRN and propose the architecture as sen in Figure 4.10. Hierarchical architecture aligns well with the hierarchical structure of videos. It allows the model to reason about objects and their interactions at different levels, from frame level to video level, while also making use of CRN units, able to capture contextual information between scene graphs.

Model	Language	Vision	Scene Graphs	Temporal Processing
Lang_MLP	✓	-	-	-
Vid_Lang_MLP	✓	✓	-	-
PoC	✓	✓	✓	-
SG-MLP	✓	✓	✓	-
SG-HCRNx	✓	✓	✓	✓

Table 5.16: Modalities & Capabilities of Models

In the training phase, we noticed that this model needed more epochs to converge compared to the previous models. This could be indicative of the complexity of this architecture and the higher number of parameters needed to capture patterns in the data.

Experiments	Lang_MLP	Vid_Lang_MLP	PoC	SG_MLP	SG_HCRNx
10k samples	21.5%	26.1%	49.1%	31.6%	42.5%

Table 5.17: Experimental results comparing our approach to baselines

After our experiments, we can see that the SG_HCRNx model outperforms the baseline models, demonstrating the value of integrating structured visual information with language features. We can also notice the performance jump from Vid_Lang_MLP to SG_HCRNx underscoring the effectiveness of utilizing scene graphs for enhancing model understanding. However, there is still a gap to fill between PoC and SG_HCRNx, due to the accuracy of the scene graph generation.

obj-rel	exists	superlative	rel-act	sequencing	obj-act	duration-comparison	action-recognition
49.8%	53.7%	55.1%	40.9%	50.4%	56.3%	25.5%	7.4%

Table 5.18: Accuracy of our final approach per question category

As we can see in Table 5.18, our approach presents strengths in understanding certain types of questions better than others. The category with the highest accuracy is 'obj-act' (object-action), that suggests that our model can recognize and understand objects and their interactions in the video, supporting our hypothesis.

As we can see in Table 5.19, our approach places second in overall score among the state-of-the-art methods. Our approach presents comparable results in almost all question categories and even outperforms in some of them. First of all, our approach achieves the highest accuracy in 'obj-rel' category, meaning our model can efficiently understand relationships between objects within the scene. We also expect this score to increase with the use of a more contemporary SGG model that generates more informative scene graphs. Our model also performs best in 'exists' and 'superlative' categories which means it can accurately identify the occurrence of concepts and objects and their order. In the rest of the categories -except for duration-, our model is ranked second, whereas in the duration category it is the least accurate method. This could indicate that our clip sampling strategy is providing too little and too sparse frames to our method so as to understand actions and their durations accurately. Our method has limitations in capturing temporal sequences and understanding temporal events. Finally, we have to mention that our scores are indicative, but not directly comparable to the other methods, since we are testing on a subset of the test set, maintaining the original set's distributions.

Method	obj-rel	rel-action	obj-action	superlative	sequencing	exists	duration	activity	Overall
PSAC [30]	37.84	49.95	50.00	33.20	49.78	49.94	45.21	4.14	40.18
HME [5]	37.42	49.90	49.97	33.21	49.77	49.96	47.03	5.43	39.89
HCRN [26]	40.33	49.86	49.85	33.55	49.70	50.01	43.84	5.52	42.11
SHG-VQA [46]	46.42	**60.67**	**64.63**	38.83	**62.17**	56.06	**48.15**	**10.12**	**49.20**
SG_HCRNx(ours)	**49.8**	53.7	55.1	**40.9**	50.4	**56.3**	25.5	7.4	42.5

Table 5.19: Comparison to sota approaches

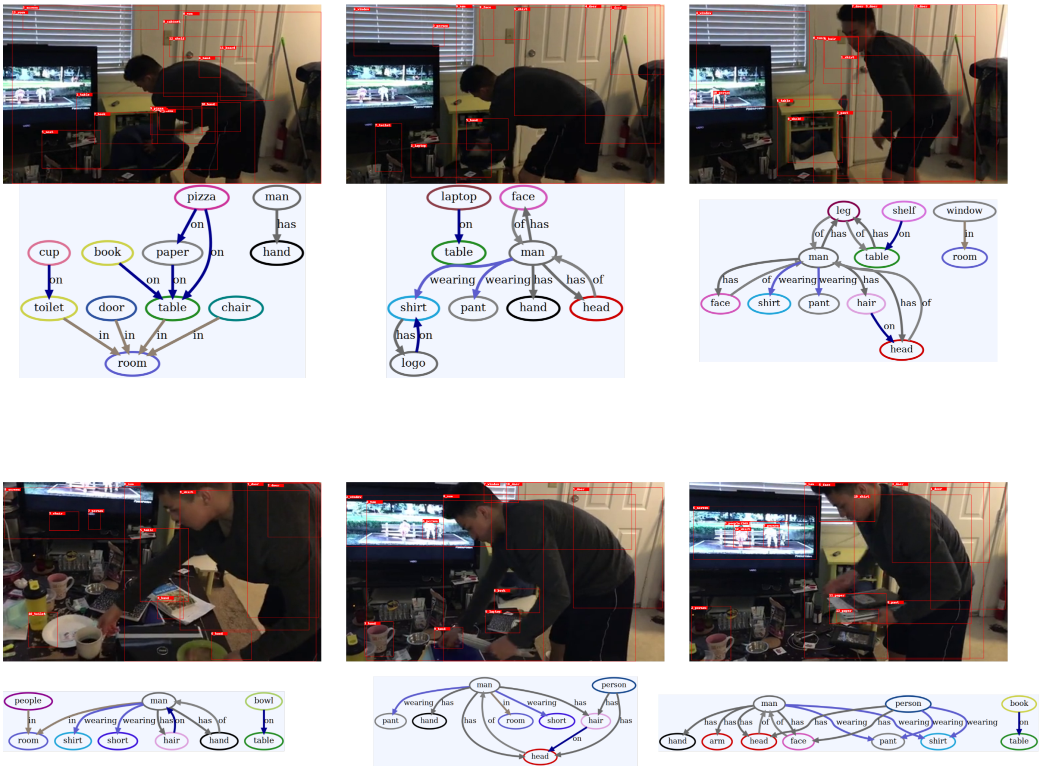


Figure 5.12: Example 1. Video sample with corresponding scene graphs. We only demonstrate 6 out of 15 sampled frames for space efficiency.

Examples of our model's performance on the video sample shown in Figure 5.13.

Question: Between putting a dish somewhere and putting their paper somewhere, what was the person tidying?
Predicted answer: table
Ground truth answer: **table**

Question: Before putting a dish somewhere, was a table the last thing they tidied?
Predicted answer: yes
Ground truth answer: **yes**

Question: Were they interacting with a closet before or after they put their paper somewhere
Predicted answer: after
Ground truth answer: **before**

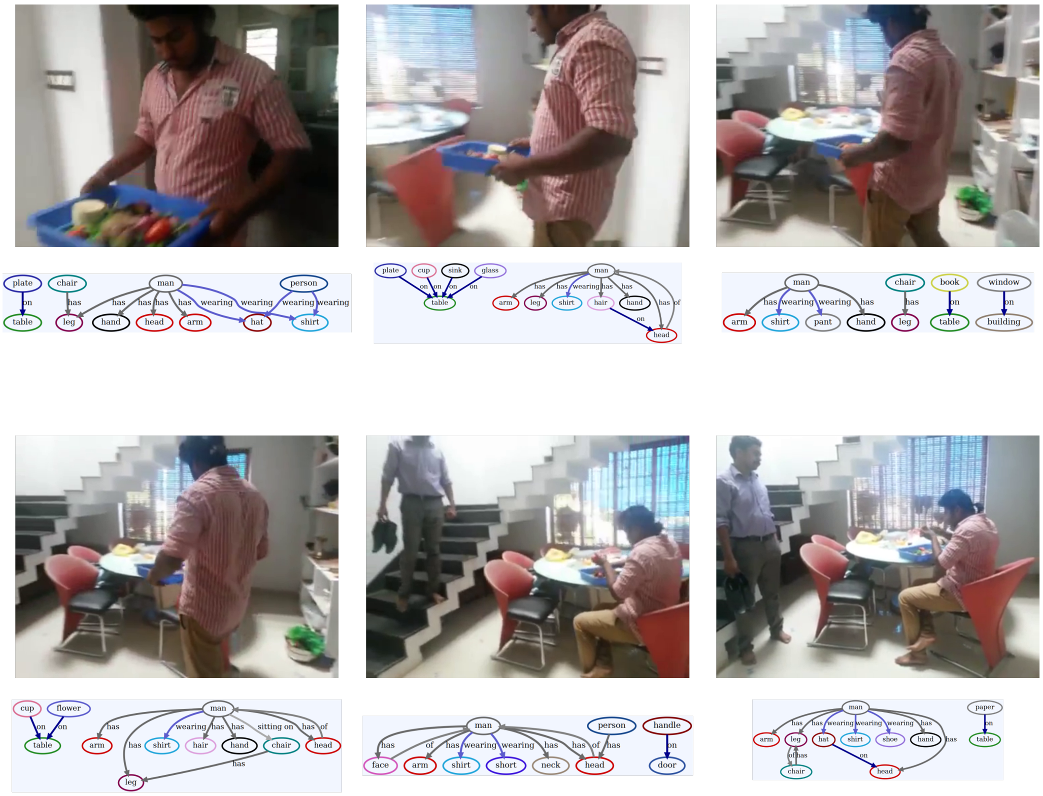


Figure 5.13: Example 2. Video sample with corresponding scene graphs. We only demonstrate 6 out of 15 sampled frames for space efficiency.

Question: Which object did the person go above while putting something on a table?
Predicted answer: chair
Ground truth answer: **chair**

Were they interacting with the object they were in front of before or after taking the object they were putting down from somewhere?
Predicted answer: before
Ground truth answer: **after**

Question: Which did they go on the side of after putting the object they were

taking somewhere, a table or a shoe?
Predicted answer: table
Ground truth answer: **shoe**

5.5 Ablation Studies

5.5.1 Hierarchical Conditional Relational Network

At first, in order to explore the importance of each modality in a hierarchical conditional network, we adjusted the code released by the dataset's authors and reproduced HCRN experiments and ablation studies.

Model	Language	Appearance	Motion	Accuracy
HCRN	✓	✓	✓	38.2%
HCRN(no motion)	✓	✓	-	37.7%
HCRN(no appearance)	✓	-	✓	37.4%
HCRN(blind)	✓	-	-	39.3%

Table 5.20: Ablation study of performance of different HCRN components

Ablation experiments evaluated the impact of excluding motion features altogether, excluding short-term motion features (clip level), and excluding long-term motion features (video level). The findings highlight that motion features are critical for detecting actions and computing action counts, with long-term motion being particularly crucial for tasks requiring a global temporal context. This demonstrates the significance of motion features in understanding the dynamics of video content.

The model was also tested without any linguistic conditioning. The results indicate that linguistic cues are essential for selecting relevant visual content, thereby improving the model's performance across different tasks. This emphasizes the role of linguistic features in providing a contextual basis for interpreting video content.

5.5.2 Graph Neural Networks

In our study, we conducted an investigation into the application and optimization of Graph Neural Networks (GNNs) . Our exploration spanned a variety of GNN architectures, examining both the efficacy of stacking multiple GNN layers and the impact of integrating different attention mechanisms. A key focus was on the inclusion of edge features within these networks, assessing how they contribute to the model's ability to capture complex relationships and interactions within the data. Additionally, we examined isomorphic networks with the ability to process edge features, because of their higher understanding level of structural properties of graphs. Beyond these configurations, our experimentation extended to more sophisticated models characterized by an increased number of layers, like SCENE.

This endeavor aimed to determine the optimal balance between model complexity and performance, exploring how deeper network architectures influence learning capacity, generalization, and computational efficiency.

Model	**Accuracy (%)**
1-layer GINE	32.8
1-layer EdgeGAT	34.6
2-layer GINE	30.5
2-layer EdgeGAT	33.1
GINE_EdgeGAT	34.6
SCENE	33.9

Table 5.21: Accuracy comparison of our non-temporal baseline approach for different GNN architectures

We can make several observations based on Table 5.21. Firstly, the single-layer configurations (1-layer GINE and 1-layer EdgeGAT) outperform their two-layer counterparts (2-layer GINE and 2-layer EdgeGAT). This suggests that adding more layers does not necessarily lead to better performance for this specific task. Also, the GINE_EdgeGAT model, matches the highest accuracy among the individual models. This architecture combines the strengths of both GINE and EdgeGAT models, getting a higher understanding of the graph's topology and structure due to the isomorphism and also comprehension of the nodes and edges because of the attention mechanism on both. This suggests that leveraging the features and mechanisms of both models can effectively capture both node and edge representations, resulting in optimal performance.

5.5.3 Temporal Graph Model

We also explored the importance of contextual integration, both at the hierarchical levels within the model and the features used as context. We first examined the importance of varying stages of contextual information and hierarchical structuring on the model's performance. Specifically, we added a question-context stage within each hierarchical level, as seen in Figure 5.14 to give our model the capacity to understand more complex questions.

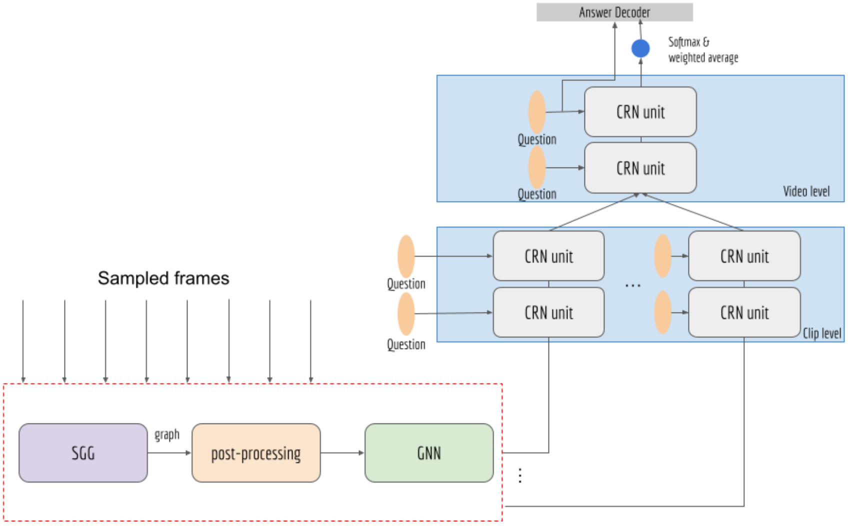


Figure 5.14: 2-stage context on question per level ablation for SG_HCRNx.

We trained and tested this architecture on the tiny setting, consisting of 10k train samples and 2k test samples. After studying the results, seen in Table 5.22, we came to the conclusion that adding an extra level of linguistic cues context doesn't enhance the model's ability to reason on complex questions, but rather performs on the same level as with one question context level per hierarchy level.

Experiments	Lang_MLP	Vid_Lang_MLP	PoC	SG_MLP	SG_HCRNx	2-stage SG_HCRNx
10k samples	21.5%	26.1%	49.1%	31.6%	42.5%	41.3%

Table 5.22: Experimental results showing the effect of adding a CRN stage inside the hierarchical levels

5.6 Experimental Conclusions

In this study, we introduce a new approach to Video Question Answering (VQA) through the development of a hierarchical architecture that enhances the model's ability to comprehend complex video content using scene graphs. This architecture employs hierarchical conditional relational networks alongside scene graphs. The utilization of scene graphs offers a structured representation of video scenes, which, as demonstrated in our experiments, plays a crucial role in boosting the model's interpretative capabilities. The effectiveness of scene graphs is particularly evident in the substantial accuracy gains observed when they are incorporated, underscoring their value in providing semantic structure to video data. This is further highlighted by the success of the upper bound baseline model that leverages ground truth scene graphs, achieving remarkable improvements in accuracy across diverse question types.

The experiments conducted reveal that our approach is good at handling questions related to object actions and relations, showcasing its capability to discern and interpret interactions and relationships among objects within videos. Notably, the model exhibits exceptional performance in answering 'exists' and 'superlative' question categories, which points to its effectiveness in identifying objects. When compared with current state-of-the-art methods, the SG HCRNx model demonstrates competitive performance, securing a second-place ranking in overall effectiveness. This achievement reflects the model's strength in understanding scene semantics and recognizing concepts and objects.

We also recognize opportunities for future work, such as refining the clip sampling strategy or incorporating more advanced scene graph generation models, which could further enhance the model's performance. Additionally, exploring more intricate temporal modeling techniques and the integration of multimodal data present promising avenues for overcoming the identified limitations.

In summary, we propose an Video Question Answering architecture that leverages scene graphs and hierarchical conditional relational networks to advance the understanding of complex video content. This work achieves remarkable improvements in model accuracy but also demonstrates the potential of structured semantic information in improving the quality of video understanding.

6 Conclusion

6.1 Conclusions

In this thesis, we have worked on scene-graphs guided Video Question Answering and have explored the integration of scene graphs to transform video content into structured representations. More specifically:

- Our research explores the integration of scene graphs into a hierarchical architecture for Video Question Answering (VQA), particularly focusing on real-world visual relations and human activities as depicted in the Action Genome Question Answering dataset.

- We hypothesize that scene graphs contain critical information for answering questions about videos, especially those involving human activities. Actions can be decomposed into spatio-temporal scene graphs that capture the relationships between objects and their attributes

- We propose a 2-stage framework: the first stage involves Scene Graph Generation (SGG) and graph formulation; the second stage focuses on training the VQA model using a Graph Neural Network (GNN) alongside a Hierarchical Architecture.

- After scene graph extraction, GNNs are utilized to derive graph embeddings that provide deeper insights into the video content by efficiently capturing the relationships and attributes of visual elements.

- A key component of our methodology is a query-conditioned graph attention unit, designed to focus on relevant parts of the scene graph embeddings based on specific queries. This unit is reusable and stackable, enhancing the model's flexibility and scalability.

- Experimental validation on a randomly sampled subset of AGQA demonstrates that our approach ranks among the top state-of-the-art methods, showing superior performance in certain question types. This indicates that transforming the video content from the pixel space to a structured sequence of scene graphs can enable better video understanding. The effectiveness of our approach is confirmed, though it is noted to be sensitive to the quality of the scene graphs.

- Ablation studies reveal that the SCENE GNN architecture, which incorporates attention features for heterogeneous graphs, node and edge features, yields the best results. Our second ablation study suggests that incorporating an additional level of linguistic cues into the model does not significantly enhance its reasoning capabilities for complex questions.

- Our research contributes to the field of Video QA by demonstrating the value of scene graphs and GNNs in enhancing the model's understanding of complex video content. Future work includes exploring the transferability of our approach to other datasets and domains, as well as investigating more advanced scene graph generation models to further improve the model's performance.

6.2 Limitations

While the adoption of scene-graph-driven approaches in Video Question Answering (Video QA) represents a significant advancement in understanding complex video content, it is imperative to acknowledge the limitations of this methodology. Our approach, which leverages structured representations through scene graphs to facilitate reasoning over video content, faces some challenges.

A principal challenge in scene-graph-driven VQA is its heavy reliance on the quality and comprehensiveness of the generated scene graphs. The system's ability to accurately answer questions is directly tied to how well the scene graph represents the video's content, including objects, attributes, and their interrelations. An incomplete or inaccurate scene graph undermines the system's performance, as it may leave out crucial details required to answer the questions. Moreover, the specificity of the graph types plays a critical role. For instance, a simple graph focusing solely on objects and their relationships might lack the necessary detail to answer questions regarding attributes, such as ”What color is the book?” This limitation points to the need for generating more detailed scene graphs that not only capture object relationships but also include visual attribute information to accommodate a wider array of question types.

Generalizing scene-graph-driven VQA systems across different domains and types of video content is hindered by the variability in objects, attributes, and relationships characteristic of each domain. Additionally, the computational intensity of generating and reasoning over scene graphs, particularly for lengthy and complex videos, limits the applicability of these systems in scenarios requiring real-time responses or where computational resources are limited.

6.3 Future Steps

6.3.1 Fine-tuned SGG

As far as quality improvements are concerned, there is much room for improvement, including the cases of inaccurate, not salient and uninformative scene graphs. This problem is very common and is caused by training in a different dataset and targeting a different application context. This can be approached through fine-tuning in the scene graph model used specifically in Action Genome Question Answering Dataset. Through this fine-tuning we will recalibrate the model's parameters to capture the specificities of the dataset. This will help us generate more accurate and relevant scene graphs boosting the whole system's performace.

6.3.2 End-to-end

An idea motivated by the analysis of the qualitative results would be to train the whole approach, including the sgg model, end-to-end, learning better graph representations and more meaningful graph embeddings at the same time.

As for the graph embeddings, we would like to explore a soft neurosymbolic approach using graphs such as the Neural State Machine [23], instead of GNNs. The Neural State Machine performs an iterative computation of a differentiable state machine over a semantic graph, so it would possibly lead to more informative graph embeddings.

Another idea for an alternative end-to-end approach is to fine-tune recent promising large-scale video and language transformer models, such as CLIP-BERT [29], which offer affordable end-to-end learning for video-and-language tasks through methods like sparse sampling and hierarchical transformers designed and tailored for the temporal dimension

6.3.3 Graph Augmentation

A future step would be to augment the scene graphs with visual information and properties of the objects, such as their bounding box coordinates etc. That way our model would be able to answer more accurately vision-related questions with information that is not currently included in the graph. For example, in a question like 'What did the woman do before grabbing the red book and after grabbing the green book?', it would be impossible for our approach to differentiate between the two books and correctly understand the video.

6.3.4 Adding Modalities

Another future step would be to add more modalities to our model. For example, we could add audio information or even text information from the video's captions. Both the sound and the captions could provide very insightful and salient information about the scene, and we could combine them with the scene graphs embeddings using multimodal fusion techniques. This would allow our model to have a more comprehensive understanding of the video content and thus be able to answer more complex questions.

References

[1] Stanislaw Antol et al. "VQA: Visual Question Answering". In: *International Conference on Computer Vision (ICCV)*. 2015.

[2] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/.

[3] Longbing Cao. "Data Science: A Comprehensive Overview". In: 50.3 (2017). ISSN: 0360-0300. DOI: 10.1145/3076253. URL: https://doi.org/10.1145/3076253.

[4] Jun Chen and Haopeng Chen. "Edge-Featured Graph Attention Network". In: *CoRR* abs/2101.07671 (2021). arXiv: 2101.07671. URL: https://arxiv.org/abs/2101.07671.

[5] Fan Chenyou et al. "Heterogeneous Memory Enhanced Multimodal Attention Model for Video Question Answering". In: *CVPR*. 2019.

[6] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[7] *Convolution operation*. https://iq.opengenus.org/content/images/2023/01/2023_01_20_0te_Kleki-min.png. [Accessed 18-01-2024].

[8] *Convolutional Neural Networks*. https://drek4537l1klr.cloudfront.net/elgendy/v-3/Figures/05_01.png. [Accessed 18-01-2024].

[9] *Day 01 Basics of Sequential Modelling , NLP and Large Language Models(LLM) — linkedin.com*. https://www.linkedin.com/pulse/day-01-basics-sequential-modelling-nlp-large-language-varshney/.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[11] Fabio Duarte. *Amount of Data Created Daily (2024)*. https://explodingtopics.com/blog/data-generated-per-day. Accessed: 2024-01-12.

[12] Chenyou Fan. "EgoVQA - An Egocentric Video Question Answering Benchmark Dataset". In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 4359–4366. DOI: 10.1109/ICCVW.2019.00536.

[13] Noa Garcia, Mayu Otani, Chenhui Chu, and Yuta Nakashima. *KnowIT VQA: Answering Knowledge-Based Questions about Videos*. 2019. arXiv: 1910.10706 [cs.CV].

[14] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. "AGQA: A Benchmark for Compositional Spatio-Temporal Reasoning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021.

[15] Mao Gu, Zhou Zhao, Weike Jin, Richang Hong, and Fei Wu. "Graph-Based Multi-Interaction Network for Video Question Answering". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 2758–2770. DOI: 10.1109/TIP.2021.3051756.

[16] Pranay Gupta and Manish Gupta. *NEWSKVQA: Knowledge-Aware News Video Question Answering.* 2022. arXiv: 2202.04015 [cs.CV].

[17] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2018.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "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.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: (2016).

[20] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997). DOI: 10.1162/neco.1997.9.8.1735.

[21] *How Convolutional Neural Networks (ConvNets or CNNs) works? — linkedin.com.* https://www.linkedin.com/pulse/how-convolutional-neural-networks-convnets-cnns-works-karel-becerra-ddzbf/?trk=article-ssr-frontend-pulse_more-articles_related-content-card.

[22] Weihua Hu et al. *Strategies for Pre-training Graph Neural Networks.* 2020. eprint: 1905.12265.

[23] Drew Hudson and Christopher D Manning. "Learning by Abstraction: The Neural State Machine". In: *Advances in Neural Information Processing Systems.* Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/c20a7ce2a627ba838cfbff082db35197-Paper.pdf.

[24] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. *TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering*. 2017. eprint: 1704.04497.

[25] Christian Janiesch, Patrick Zschech, and Kai Heinrich. “Machine learning and deep learning”. In: *Electronic Markets* 31.3 (Apr. 2021), pp. 685–695. ISSN: 1422-8890. DOI: 10.1007/s12525-021-00475-2. URL: http://dx.doi.org/10.1007/s12525-021-00475-2.

[26] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. “Hierarchical Conditional Relation Networks for Video Question Answering”. In: (2020).

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[28] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. *TVQA: Localized, Compositional Video Question Answering*. 2018. eprint: 1809.01696.

[29] Jie Lei et al. *Less is More: ClipBERT for Video-and-Language Learning via Sparse Sampling*. 2021. eprint: 2102.06183.

[30] Xiangpeng Li et al. “Beyond RNNs: Positional Self-Attention with Co-Attention for Video Question Answering”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 8658–8665. ISSN: 2159-5399. DOI: 10.1609/aaai.v33i01.33018658. URL: http://dx.doi.org/10.1609/aaai.v33i01.33018658.

[31] Thomas Monninger et al. “SCENE: Reasoning About Traffic Scenes Using Heterogeneous Graph Neural Networks”. In: *IEEE Robotics and Automation Letters* 8.3 (Mar. 2023), pp. 1531–1538. ISSN: 2377-3774. DOI: 10.1109/lra.2023.3234771. URL: http://dx.doi.org/10.1109/LRA.2023.3234771.

[32] Fionn Murtagh. “Multilayer perceptrons for classification and regression”. In: *Neurocomputing* 2 (1991).

[33] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[34] Junjie Peng, Elizabeth Jury, Pierre Dönnes, and Coziana Ciurtin. "Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges". In: *Frontiers in Pharmacology* 12 (Sept. 2021). DOI: 10.3389/fphar.2021.720694.

[35] F. Rosenblatt. *The perceptron - A perceiving and recognizing automaton*. Tech. rep. 85-460-1. Ithaca, New York: Cornell Aeronautical Laboratory, Jan. 1957.

[36] Iqbal H. Sarker. "Machine learning: Algorithms, real-world applications and Research Directions". In: *SN Computer Science* 2.3 (2021). DOI: 10.1007/s42979-021-00592-x.

[37] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision*. 2016. eprint: 1512.00567.

[39] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. *Scene Graphs Benchmark.pytorch - Unbiased Scene Graph Generation from Biased Training*. https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch. 2013.

[40] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. *Learning to Compose Dynamic Tree Structures for Visual Contexts*. 2019. eprint: 1812.01880.

[41] Makarand Tapaswi et al. "MovieQA: Understanding Stories in Movies through Question-Answering". In: (2016).

[42] Makarand Tapaswi et al. *MovieQA: Understanding Stories in Movies through Question-Answering*. 2016. eprint: 1512.02902.

[43] Petroc Taylor. *Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025*. https://www.statista.com/statistics/871513/worldwide-data-created/. Accessed: 2024-01-12.

[44] Suramya Tomar. "Converting video formats with FFmpeg". In: *Linux Journal* 2006.146 (2006), p. 10.

[45] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "C3D: Generic Features for Video Analysis". In: *CoRR* abs/1412.0767 (2014). arXiv: 1412.0767. URL: http://arxiv.org/abs/1412.0767.

[46] Aisha Urooj et al. “Learning Situation Hyper-Graphs for Video Question Answering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 14879–14889.

[47] Ashish Vaswani 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: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[48] Petar Veličković et al. *Graph Attention Networks*. 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[49] Minjie Wang et al. “Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs”. In: *CoRR* abs/1909.01315 (2019).

[50] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. *NExT-QA:Next Phase of Question-Answering to Explaining Temporal Actions*. 2021. eprint: 2105.08276.

[51] Dejing Xu et al. “Video Question Answering via Gradually Refined Attention over Appearance and Motion”. In: *ACM Multimedia*. 2017.

[52] Gokul Yenduri et al. *Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions*. 2023. arXiv: 2305.10435 [cs.CL].

[53] Kexin Yi* et al. “CLEVRER: Collision Events for Video Representation and Reasoning”. In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HkxYzANYDB.

[54] Zhou Yu et al. “ActivityNet-QA: A Dataset for Understanding Complex Web Videos via Question Answering”. In: *AAAI*. 2019, pp. 9127–9134. eprint: 1906.02467.

[55] Yuan Yuan, Xiaodan Liang, Xiaolong Wang, Dit-Yan Yeung, and Abhinav Gupta. “Temporal Dynamic Graph LSTM for Action-driven Video Object Detection”. In: *ICCV*. 2017.

[56] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. “Neural Motifs: Scene Graph Parsing with Global Context”. In: *CoRR* abs/1711.06640 (2017). arXiv: 1711.06640. URL: http://arxiv.org/abs/1711.06640.

[57] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. *Visual Translation Embedding Network for Visual Relation Detection*. 2017. eprint: 1702.08319.

[58] Yaoyao Zhong et al. *Video Question Answering: Datasets, Algorithms and Challenges*. 2022.