# Scene Graph Generation and Reasoning for Visual Question Answering

**Anirudh Kannan**
Language Technologies Institute
Carnegie Mellon University
akannan3@andrew.cmu.edu

## Abstract

Visual Question Answering (VQA) is a challenging task that aims to develop machine learning algorithms capable of answering questions about visual content. However, VQA presents significant difficulties, particularly in the need for advanced reasoning and the ability to handle complex questions. To overcome these challenges, we propose a novel approach that integrates scene graphs into LXMERT (Tan and Bansal, 2019), a state-of-the-art VQA Model. Our approach leverages the rich context provided by scene graphs, which captures the relationships between objects, to enhance the spatial reasoning capabilities of LXMERT. Our analysis of experimental results reveals that incorporating scene graphs as context into LXMERT yields a significant improvement in its performance on VQA tasks.

## 1 Introduction

The field of Visual Question Answering (VQA) is experiencing rapid growth and innovation, driven by the objective of creating machine learning algorithms that can accurately respond to questions related to visual information. VQA models can be used to reduce barriers for visually impaired individuals by allowing them to get information about images from the web and the real world. VQA requires question answering systems to effectively reason over two distinct modalities, text and images. This task involves processing an image and a natural language question as input and generating a natural language answer as output. VQA algorithms must possess both visual and language understanding to reason about the connections between the image and the question and generate accurate answers. Several VQA datasets, including VQA 2.0 (Antol et al., 2015) and GQA (Hudson and Manning, 2019), have been introduced, stimulating a surge of research in this field.

There are several state of the art models for VQA, including LXMERT (Tan and Bansal, 2019), VILBERT (Lu et al., 2019), and UNITER (Chen et al., 2020). These models have achieved remarkable performance on various VQA benchmarks and their success could largely be attributed to their ability to leverage pre-training on large-scale datasets. While considerable progress has been made in recent years, VQA still presents several challenges, including the need to handle complex questions that demand advanced spatial reasoning abilities.

One promising way to address these challenges is to incorporate scene graphs along with the visual content as context. Scene graphs are structured representations of an image, where entities and associated attributes are embedded as nodes and their relationships as edges (Zhu et al., 2022). Scene graphs have shown great potential in enhancing the ability of VQA models to understand the complex relationships between objects in an image, which in turn helps to answer questions that demand advanced spatial reasoning and counting skills (Damodaran et al., 2021). By utilizing scene graphs, VQA systems might be able disentangle various visual factors and reduce the impact of irrelevant information, leading to improved generalization across different settings and greater interpretability of the system

This work delves into a series of research questions concerning the integration of scene graphs into VQA models. Specifically, we investigate whether incorporating scene graphs can lead to improvements in VQA performance, whether VQA models can leverage question-specific scene graph information to achieve higher accuracy, and whether a pipelined approach involving scene graph generators can enhance the overall performance of VQA systems. By addressing these fundamental questions, we aim to contribute to a better understanding of the potential benefits and challenges associated with using scene graphs in VQA and pave the way for future advancements in the field.

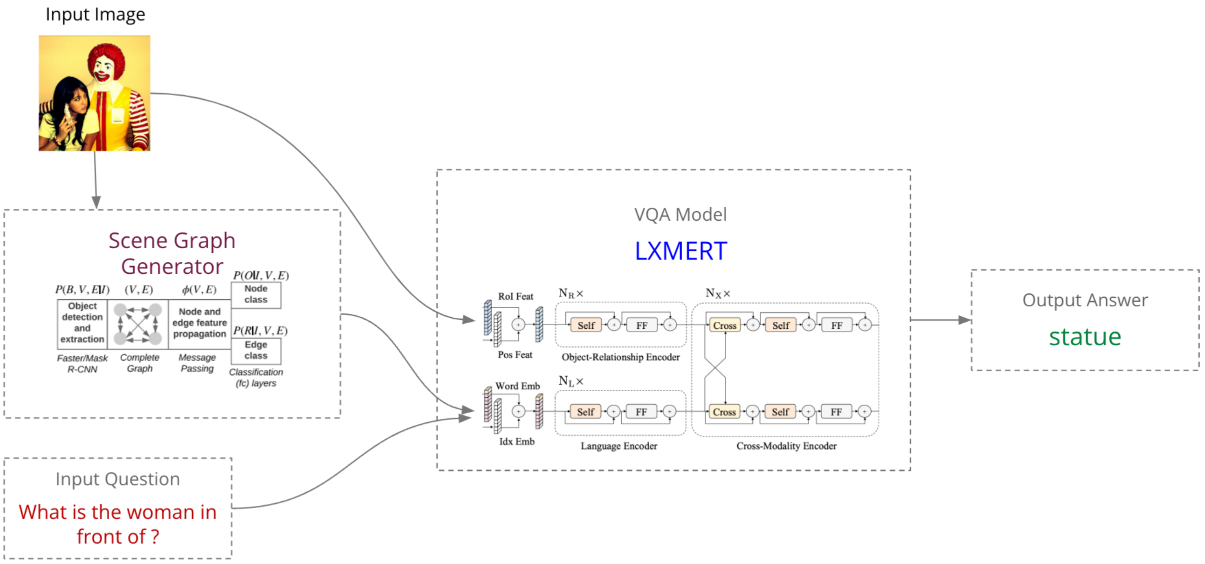


Figure 1: Pipelined Architecture

## 2 Related Works

(Hudson and Manning, 2019) proposed the GQA dataset which is a large-scale VQA dataset that leverages scene graphs from Visual Genome (Krishna et al., 2016) to create a diverse set of questions (22 million) with over 113,000 images. The use of scene graphs in this dataset has shown promise in addressing the challenges of developing complex and challenging questions that require advanced reasoning abilities.

(Lee et al., 2019) was one of the first works to explore the use of scene graphs for the VQA task. The authors encoded a scene graph using graph neural networks and then fed the encoded graph and question as input to the Memory, Attention, and Composition (MAC) model (Hudson and Manning, 2018). Their study found that augmenting the question answering model with representations learned from the auxiliary scene graph processing task, where graph neural networks were utilized to learn condensed embeddings, led to significant improvement in performance over existing baselines.

Along similar lines, (Zhang et al., 2019) performed an empirical study on the use of only scene graphs as the context for question answering and concluded that "scene graphs, even automatically generated by machines, can definitively benefit Visual QA if paired with appropriate models like GNs." (Zhang et al., 2019). Instead of combining image and language features, this paper proposes using scene graphs derived from images for Visual QA. They use graph neural networks (GNNs) to encode the scene graph and perform structured reasoning based on input questions. Their empirical studies show that scene graphs can capture essential image information and graph neural networks have potential to outperform current Visual QA methods with a simpler architecture.

In a study by (Damodaran et al., 2021), an extensive analysis was conducted on the use of scene graphs for visual question answering (VQA). The researchers employed a late fusion approach between GNNs and the MAC model (Hudson and Manning, 2018), using scene graph inputs alongside features from UNITER. Their findings highlight the potential benefits of utilizing scene graphs in VQA tasks. Moreover, the study investigated the impact of using ground truth versus generated scene graphs on the model's performance. The results revealed a notable difference in performance between the two, with ground truth scene graphs outperforming the generated ones. This observation bears significance as it aligns with similar patterns observed in this work.

Recent successes of GraphVQA (Liang et al., 2021), which utilizes graph neural networks to process natural language questions through multiple message-passing iterations, have highlighted the potential benefits of incorporating scene graphs into the VQA task. With this in mind, it is reasonable to hypothesize that leveraging scene graphs can provide substantial enhancements to VQA models.

# 3 Baselines

## 3.1 VQA Model

LXMERT (Tan and Bansal, 2019), used as the baseline VQA model in this study, stands out for its ability to seamlessly integrate both visual and linguistic information, making it highly proficient at answering questions about visual content. It's transformer architecture consists of three specialized encoders - an object relationship encoder, a language encoder, and a cross-modality encoder. The model is pre-trained on large-scale datasets of image-and-sentence pairs through various tasks, such as masked cross-modality language modeling, masked object prediction via RoI-feature regression, masked object prediction via detected-label classification, cross-modality matching, and image question answering. Being a state-of-the-art model, it has achieved impressive scores in the GQA task, making it an ideal choice for this study.

## 3.2 Scene Graph Generator

The goal of Scene Graph Generation (SGG) is to synthesize a graphical representation of a given input image (Zhu et al., 2022). This involves identifying objects within the image (nodes) and determining relationships between these objects (predicates). This process enables a more comprehensive understanding of visual content and has numerous applications, including image retrieval and visual reasoning. Typically, the process of generating a scene graph from an image involves two distinct phases. In the first phase, an object detector such as Faster-RCNN (Ren et al., 2016) is trained to identify objects present in the image. In the second phase, the features extracted from the object detector are used to train a relation detector model responsible for identifying potential edge pairs between the detected objects and predicting the type of relationship between them.

In this study, we utilized a scene graph generator based on (Knyazev et al., 2020) which leverages different loss functions for foreground and background edges in a graph, and represents them as a function of graph densities. While the foreground edges (FG) play a role similar to a "positive class" , the background (BG) edge type plays a similar role to a "negative" class in object detection tasks. The BG edges represent pairs of nodes that lack any meaningful relationship or connection. By training the model on these BG edges, it becomes capable of distinguishing between positive and negative edges, preventing it from labeling all node pairs as "positive" during testing. This approach ensures the model's proficiency in recognizing meaningful connections among nodes and aids in generating accurate scene graphs. An example of a scene graph that has been generated can be found in Figure 2.

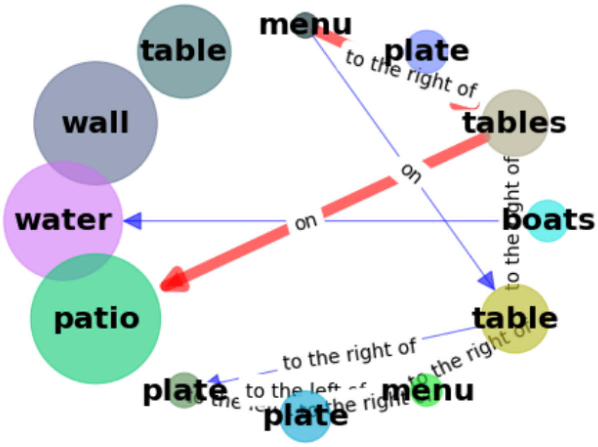


Figure 2: An example of a generated scene graph

# 4 Methodology

This study explores the impact of incorporating scene graphs as context for the Visual Question Answering (VQA) task. To this end, we experiment with two different configurations. In the first configuration, we append gold standard scene graph information (extracted from the GQA dataset) to the question and pass it to LXMERT to establish a baseline. This approach enables us to estimate the upper bound of the expected performance improvement that can be achieved with scene graph information. In the second configuration, which we term as the "pipeline", we utilize a scene graph generator to automatically generate scene graphs from the input image, instead of relying on manually annotated gold standard data. The generated scene graphs are then incorporated as context to the LXMERT model, and its performance is benchmarked.

More formally, given a question $q$ and an image $I$, the LXMERT model predicts an answer $\hat{a}$ as follows:

$$\hat{a} = \text{LXMERT}(q, I)$$

With the addition of scene graph information ($SG$), the prediction of $\hat{a}$ can be formulated as:

$$\hat{a} = \text{LXMERT}(q + SG, I)$$

| Question Type | Validation | Train |
|---|---|---|
| what | 42267 | 301452 |
| is | 34982 | 249212 |
| are | 16204 | 115227 |
| who | 7510 | 53475 |
| of | 7098 | 52272 |
| do | 6649 | 47583 |
| does | 5818 | 41428 |
| on | 5190 | 36646 |
| the | 2265 | 16110 |
| how | 2204 | 15996 |
| where | 1822 | 13248 |

Table 1: Validation and Training counts by question type in GQA Dataset

where $q + SG$ represents the concatenated input of the question and the gold standard scene graph information from the GQA dataset.

For the pipelined approach, given a scene graph generator $f(.)$, the predicted answer $\hat{a}$ can be represented as:

$$\hat{a} = \text{LXMERT}(q + f(I), I)$$

where $q + f(I)$ represents the concatenated input of the question and the scene graph generated automatically by the scene graph generator.

## 5 Dataset

For this project, we utilized the GQA dataset (Hudson and Manning, 2019), which consists of over 22 million compositional questions and more than 113,000 images, encompassing a diverse range of reasoning skills. Images and corresponding scene graphs were sourced from Visual Genome (Krishna et al., 2016), with each image paired with a scene graph that represents its semantics. The questions were synthesized using a question engine that leverages the scene graph information derived from Visual Genome. Each question is associated with a functional program that outlines the series of reasoning steps required to arrive at the answer as well.

### 5.1 Dataset Distribution

The distribution of the different question types in the GQA dataset is illustrated in Figure 3 and Figure 4. "What" questions appear to be the most frequently occurring question type, followed by "is" and "are" question types. Moreover, the distribution of question types is consistent between the training and validation sets. Additional information on the count of each question type is presented in Table 1.

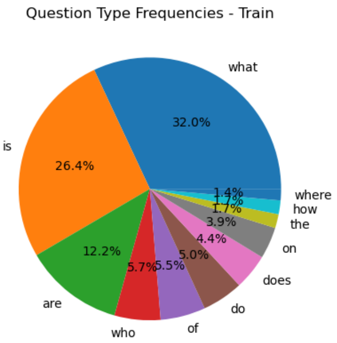


Figure 3: Train data distribution with respect to question type in GQA

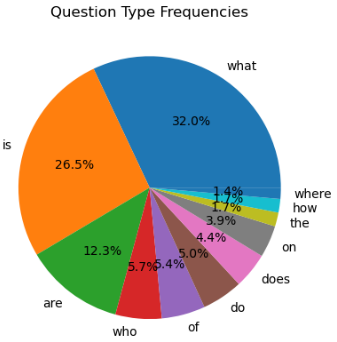


Figure 4: Validation data distribution with respect to question type in GQA

## 6 Experiments

In the first set of experiments, we investigated the integration of gold standard scene graphs into LXMERT by testing six different model configurations:

- **V1**: The model provided by the authors of LXMERT in the official implementation, which was fine-tuned on both the training and validation sets of GQA to maximize accuracy on the dev test set. However, since dev test scene graphs are not provided, we cannot use this model as the baseline.

- **V2**: A pre-trained LXMERT model fine-tuned on only the training set to establish a performance baseline (due to the reason stated above).

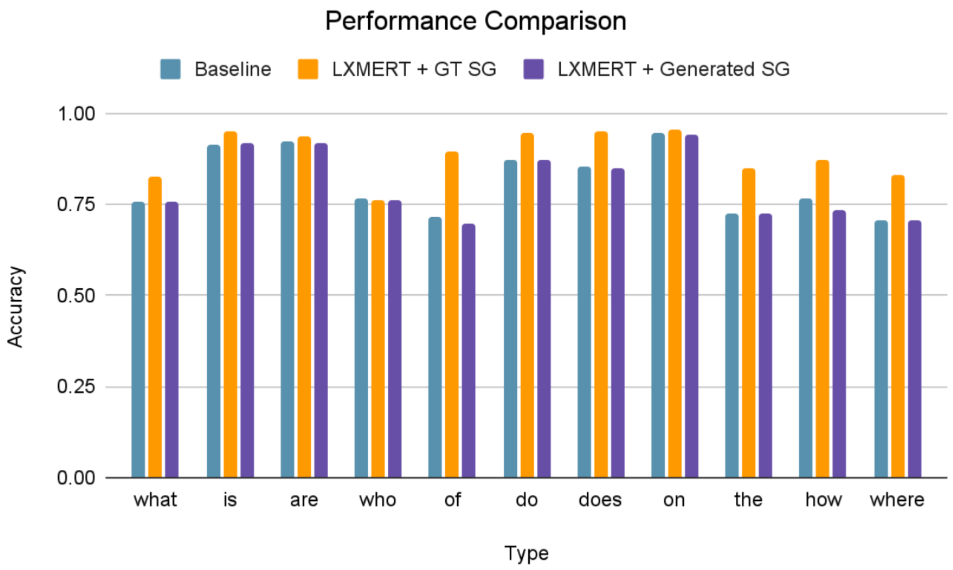


Figure 5: Performance comparison of different configurations

- **V3**: Adding the entire scene graph information as is, by concatenating it to the question before passing it to LXMERT along with the image.

- **V4**: Preprocessing the scene graph information before concatenation by removing redundant information, grouping by object name, and adding words like "is" and "are" to improve grammatical structure.

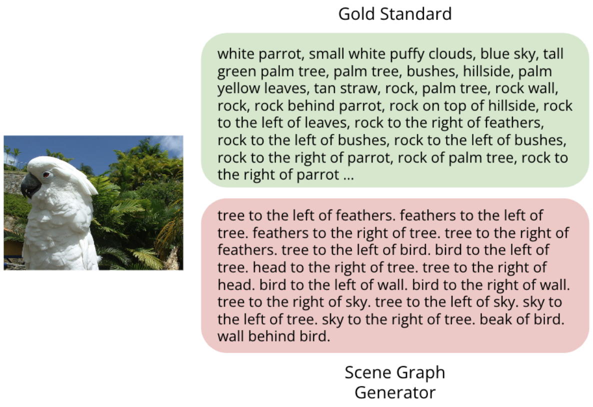
- **V5**: Appending the scene graph information before the question (as opposed to after in V3).

- **V6**: Preprocessing the scene graph to be query-specific by including only objects and related predicates referenced in the query, which were then added as context to the question.

The aim of this experiment was to assess the effectiveness of these different configurations in improving the performance of LXMERT through scene graph integration. The results of this experiment can be found in Table 2.

In the subsequent set of experiments, the gold standard scene graphs were replaced with automatically generated scene graphs produced by the scene graph generator (Knyazev et al., 2020). Under these new conditions, the performance of configurations V1, V2, V3, and V6 was evaluated again. The scene graph generator used a Faster RCNN Detector (Ren et al., 2016) and Resnet 50 Backbone for object detection, as well as Neural Motifs (Zellers et al., 2018). The primary objective of these experiments was to assess the performance of different model configurations when operating with non-gold standard scene graphs generated by the scene graph generator. The results of these experiments can be found in Table 3.

## 7 Results and Error Analysis



white parrot, small white puffy clouds, blue sky, tall green palm tree, palm tree, bushes, hillside, palm yellow leaves, tan straw, rock, palm tree, rock wall, rock, rock behind parrot, rock on top of hillside, rock to the left of leaves, rock to the right of feathers, rock to the left of bushes, rock to the left of bushes, rock to the right of parrot, rock of palm tree, rock to the right of parrot ...

tree to the left of feathers. feathers to the left of tree. feathers to the right of tree. tree to the right of feathers. tree to the left of bird. bird to the left of tree. head to the right of tree. tree to the right of head. bird to the left of wall. bird to the right of wall. tree to the right of sky. tree to the left of sky. sky to the left of tree. sky to the right of tree. beak of bird. wall behind bird.

Figure 6: Gold Standard vs Generated Scene Graph comparison

Table 2 presents compelling evidence that incorporating scene graphs can generally enhance the accuracy of LXMERT, albeit to varying degrees depending on the specific configuration. Particularly noteworthy is the substantial improvement (almost 5.1 %) seen in V6 when using gold standard scene graphs. Conversely, V5 experiences a significant

| MODEL | TRAINING ACCURACY (%) | VALIDATION ACCURACY (%) | DEV TEST (%) |
|---|---|---|---|
| V1 | - | 94.58 | 59.73 |
| V2 (BASELINE) | 92.50 | 83.73 | 59.65 |
| V3 | 92.53 | 83.82 | 59.37 |
| V4 | 92.47 | 83.78 | 59.34 |
| V5 | 23.53 | 19.91 | 52.76 |
| **V6** | **95.24** | **88.83** | **58.89** |

Table 2: LXMERT metrics after addition of Gold Standard Scene Graphs

| MODEL | TRAINING ACCURACY (%) | VALIDATION ACCURACY (%) | DEV TEST (%) |
|---|---|---|---|
| V1 | - | 94.58 | 59.73 |
| V2 (BASELINE) | 92.50 | 83.73 | 59.65 |
| V3 | 92.53 | 83.66 | 59.63 |
| V6 | 95.24 | 88.65 | 59.58 |

Table 3: LXMERT metrics after addition of Generated Scene Graphs

decline in performance with the addition of scene graphs. It is plausible that the decrease in performance may be attributed to LXMERT's inability to effectively handle large contexts, resulting in difficulty comprehending the question when given a substantial scene graph. For V3 and V4, the accuracy improvement is relatively minor.

However, the integration of scene graphs generated by the scene graph generator seems to have resulted in a decline in accuracy, as Table 3 demonstrates. The validation accuracies across all model configurations seem to have declined. This could potentially be attributed to the relatively poor quality of the generated scene graphs, which appear to encode only simple positional information, as shown in the Figure 6. Thus, the significant difference in information richness between the ground truth and generated scene graphs could be one possible reason for the observed decline in performance when using generated scene graphs. The performance of different models across various question types is illustrated in Figure 5. It is evident that the LXMERT model with ground truth scene graphs shows a clear and notable improvement across all question types, with remarkable enhancements observed in questions requiring positional reasoning, such as "where", "the", "of", and so forth. However, no such improvement can be observed when generated scene graphs are used. Specific qualitative case studies of misclassification can be found in the appendix.

Figure 6 provides an example of the contrast between the quality of the gold standard scene graph and that of the generated scene graph. The difference is clearly noticeable, with the gold standard scene graph containing more diverse and richer information, which appears to have contributed to the improved performance of the model.

This finding underscores the importance of high-quality scene graphs in improving the performance of visual question answering models that rely on them. Further research and development of scene graph generators may be necessary to improve their accuracy and ensure that the scene graphs they produce contain richer and more informative data.

## 8 Conclusion and Future Work

This study has highlighted the importance of incorporating scene graphs as additional context in visual question answering using the state-of-the-art model, LXMERT. The findings reveal that adding gold standard scene graphs significantly improves the model's accuracy, with query-specific scene graphs boosting accuracy by **5.1%** overall. However, the use of automatically generated scene graphs from the scene graph generator resulted in decreased accuracy, as similarly observed in (Damodaran et al., 2021). The lower quality of generated scene graphs compared to gold standard scene graphs may have contributed to this outcome.

Future research could focus on enhancing the quality of generated scene graphs or consider alternative methods for generating high-quality scene graphs for VQA tasks. One possible approach to achieve this is by exploring the use of advanced techniques such as attention mechanisms or graph neural networks. Attention mechanisms can aid in selecting and weighting relevant information in the scene graph, while graph neural networks can help capture more intricate and nuanced relationships among the objects present in the scene. By utilizing these sophisticated techniques, it may be possible to enhance the accuracy and efficiency of LXMERT models for natural language processing and scene understanding tasks. Another interesting perspective could be exploring modification of scene graphs based on natural language commands similar to (He et al., 2020). In this paper, the authors propose a novel encoder-decoder architecture that utilizes graph-conditioned transformers and cross-attention to address the problem of Scene Graph modification. It may be possible to apply a similar approach to this work.

## Limitations

A notable drawback of this study is the absence of a completely separate test dataset and evaluation metric. At present, the validation accuracy is the sole metric used to assess the models' performance. The reason behind this is that the Scene Graph Generator is trained on MSCOCO data, which has considerable overlap with the GQA and Visual Genome datasets. This overlap raises concerns about the possibility of data leakage, where the model may learn to rely on features unique to the MSCOCO dataset, resulting in inflated results that may not generalize well to other datasets. Moreover, while the dev test set is relatively isolated, it does not provide ground truth scene graphs, making it challenging to perform a direct comparison between the generated and gold standard scene graphs.

Furthermore, upon the incorporation of gold standard scene graphs, there was a noticeable 5.1 % increase in validation accuracy. However, the performance of the Scene Graph Generator remained inferior, leading to a decline in the overall performance of LXMERT. This outcome underscores the need for a more in-depth investigation into the Scene Graph Generator to identify and address the reasons behind its subpar performance.

## Ethics Statement

Both the Scene Graph Generator and LXMERT, which are computer vision models, rely on the MSCOCO dataset for their image data. However, it has been noted that this dataset has gender biases, which may raise concerns when these models are implemented in real-world scenarios (Hirota et al., 2022). These biases may be the result of various factors, such as the demographics of the individuals who collected the data or the methods used to select and label the data. These biases can have unintended consequences when these models are used in real-world settings, potentially perpetuating existing inequalities or stereotypes. Therefore, it is essential to carefully consider the potential biases of datasets and models and strive to mitigate them to ensure fair and ethical deployment.

## Deployment Challenges

Deploying this pipeline has several challenges, including the high computational cost due to the large size of models involved, the requirement for specialized hardware such as GPUs for efficient processing, the complexity of integrating scene graphs with natural language processing models, and the need for careful curation and labeling of training data to ensure accurate and reliable performance. Additionally, maintaining performance during real-time testing may be challenging due to domain shift.

## Computational Costs

For the project, GCP Credits worth approximately $350 were utilized. The instance used for the project had an n1-highmem-16 configuration, with an Intel Skylake CPU platform, 112 GB RAM, and a GPU comprising of either Nvidia T4 or Nvidia Tesla P100. It took around 21 hours to finetune a single LXMERT model using this configuration. About 10 finetuning experiments were performed throughout the course of this project.

## Contributions

Anirudh Kannan was the sole contributor to this project and received guidance and mentorship from Syeda Akter.

## Acknowledgements

The project owes a debt of gratitude to Professor Eric Nyberg and Professor Teruko Mitamura for their invaluable guidance and support. Furthermore, I express my deep appreciation for the invaluable mentorship provided by Syeda Akter, whose insightful guidance and contributions were indispensable to the success of this project.

# References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. *CoRR*, abs/1505.00468.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning.

Vinay Damodaran, Sharanya Chakravarthy, Akshay Kumar, Anjana Umapathy, Teruko Mitamura, Yuta Nakashima, Noa Garcia, and Chenhui Chu. 2021. Understanding the role of scene graphs in visual question answering.

Xuanli He, Quan Hung Tran, Gholamreza Haffari, Walter Chang, Trung Bui, Zhe Lin, Franck Dernoncourt, and Nhan Dam. 2020. Scene graph modification based on natural language commands.

Yusuke Hirota, Yuta Nakashima, and Noa Garcia. 2022. Quantifying societal bias amplification in image captioning.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering.

Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W. Taylor, Aaron Courville, and Eugene Belilovsky. 2020. Graph density-aware losses for novel compositions in scene graph generation.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. 2019. Visual question answering over scene graph. In *2019 First International Conference on Graph Computing (GC)*, pages 45–50.

Weixin Liang, Yanhao Jiang, and Zixuan Liu. 2021. GraghVQA: Language-guided graph neural networks for graph-based visual question answering. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 79–86, Mexico City, Mexico. Association for Computational Linguistics.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: Towards real-time object detection with region proposal networks.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context.

Cheng Zhang, Wei-Lun Chao, and Dong Xuan. 2019. An empirical study on leveraging scene graphs for visual question answering.

Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, and Mohammed Bennamoun. 2022. Scene graph generation: A comprehensive survey.

# A Appendix

For code and additional information, see Github.

## A.1 Case Studies

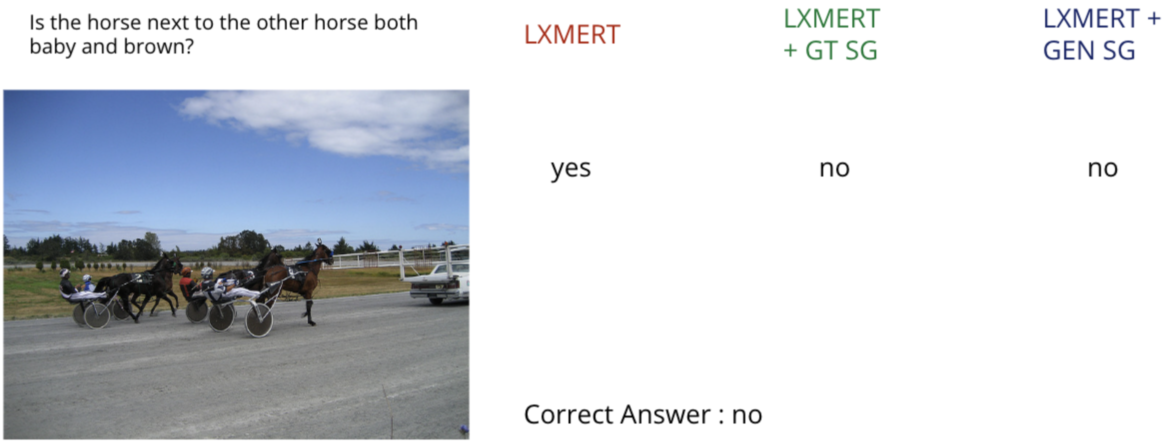


Figure 7: Case Study 1

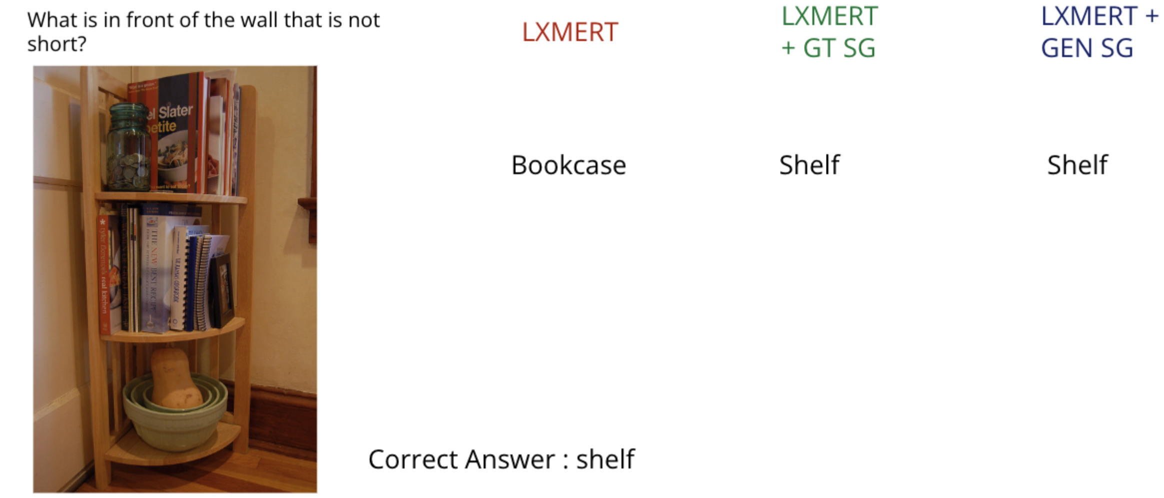


Figure 8: Case Study 2

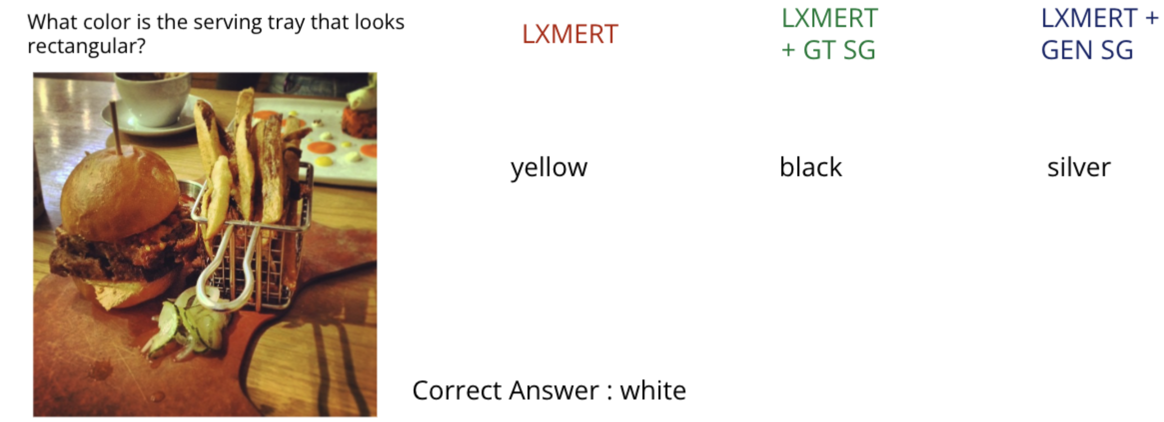


Figure 9: Case Study 3