Tables:  
dataset tables, results without scene with scene combined

Vqa2.0

Visual genome

Attributes and relationships

Results

Screenshots

Accuracy ss

Sections:  
methodology (this includes introduction about scene graphs, how scene graphs helps, datasets used)

Understanding images and text together is what makes Visual Question Answering (VQA) such a challenging problem it requires reasoning at the level of human intelligence, often referred to as an AI-complete task.

Traditional VQA models, built on deep learning, typically use convolutional neural networks (CNNs) like Faster R-CNN to extract features from images, while transformers process the accompanying text. These features are then combined through a multimodal fusion mechanism to predict an answer.

However, despite advancements in transformer-based models like LXMERT, VQA systems still face major hurdles:

* **Complex reasoning** – Understanding spatial relationships and functional dependencies remains difficult.
* **Ambiguous answers** – Questions can have multiple correct answers, especially when objects share similar attributes.
* **Lack of explicit object relationships** – For example, answering "Who is holding the umbrella?" requires not just detecting "who" and "umbrella" but also understanding their relationship.

To tackle these challenges, I introduce **scene graphs** a structured way of representing images that captures objects, their attributes, and relationships explicitly. Instead of relying solely on feature extraction, scene graphs provide a graph-based structure where:

* **Objects (nodes)** represent elements in the image (e.g., "dog," "table," "person").
* **Attributes** describe properties of these objects (e.g., "red ball," "wooden table").
* **Relationships (edges)** define interactions between objects (e.g., "dog chasing ball," "man sitting on chair").

By integrating scene graphs, we enhance VQA models in several ways:

* **Better object-level understanding** – The model does not just detect objects; it grasps their properties and relationships.
* **More accurate question-answer mapping** – Many VQA questions rely on understanding relationships (e.g., "Who is under the table?"), and scene graphs enable direct reasoning about such dependencies.
* **Reduced ambiguity** – Explicitly encoding relationships minimizes misinterpretation of complex scenes.
* **Improved generalization** – Instead of memorizing dataset-specific patterns, the model learns reusable object-relation structures, making it more adaptable to unseen questions.
* **Less dependency on massive datasets** – Traditional models need large amounts of data to learn implicit relationships, while scene graphs provide this knowledge explicitly, leading to more efficient learning.
* **More efficient attention mechanisms** – Instead of analyzing the entire image, the model can focus on relevant objects and their interactions, reducing computational overhead and improving accuracy.

**Datasets Used**

This project utilizes two primary datasets:

**VQA v2.0 Dataset**

* Contains **image-question-answer triplets**, where the goal is to answer questions based on image content.
* However, **it does not provide structured relational information**, making complex reasoning difficult.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Total Entries** | Number of QA pairs per image 3-5 Number of unique images in  Train: 82783 Validation: 40504  Total: Training: 443744  Validation: 214336 |
| **Answers** | List of 10 human annotated answers |
| **Question ID** | Unique identifier for each question |
| **Image ID** | Unique identifier for each image |
| **Question** | The natural language question about the image |
| **Image Features** | Extracted image feature vectors from Faster R-CNN Array of shape (N, 4), N varies |

**4.2. Visual Genome (VG) Dataset**

* Contains **detailed scene graphs** for images.
* Provides structured relationships between objects, enhancing **relational reasoning capabilities**.
* Serves as an auxiliary dataset to **train the model to understand explicit relationships**.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Total Entries** | Number of QA pairs per image ~17 avg Number of unique images in  Train: 64346 Validation: 43903  Total Used: Training: 443744  Validation: 214336 |
| **Answers** | 1 human annotated answer |
| **Question ID** | Unique identifier for each question |
| **Image ID** | Unique identifier for each image |
| **Question** | The natural language question about the image |
| **Image Features** | Extracted image feature vectors from Faster R-CNN Array of shape (N, 4), N varies |

Attributes and relationships

|  |  |
| --- | --- |
| **JSON File** | **Key Fields** |
| **Attributes** | Image id: ID of the image  objects: List of objects with attributes  object id: Unique ID for each object  names: List of object names  attributes: List of attribute labels for the object  Bounding box (coordinates defining object location) |
| **Relationships** | Image id: ID of the image  relationships: List of relationships  subject: Object initiating the relationship  predicate: Relationship type (e.g., "on", "next to")  object: Object receiving the relationship  Bounding boxes for both subject and object |

By integrating **Visual Genome’s scene graphs with the VQA v2.0 dataset**, we enable the model to:

* **Enhance object-relationship learning** through **graph-based reasoning**.
* **Provide explicit structural knowledge**, instead of requiring the model to infer relationships implicitly.

Approach Implementation approach (this includes implementation details, how model was combined, how it was trained, different methods)

Preprocessing  
**Data Preprocessing**

**VQA v2.0 Dataset**

1. **Image Preprocessing:**
   * Images were processed using the **Faster R-CNN** model to extract **visual features**.
   * The extracted features were stored in a **PKL (Pickle) file** for efficient storage and retrieval.
2. **Combining Processed Data:**
   * The preprocessed image features (from the PKL file) were combined with the **questions and answers** dataset.
   * A **final PKL file** was created, containing the following fields:
     + **Answers:** The possible answers for each question.
     + **Question ID:** A unique identifier for each question.
     + **Image ID:** The corresponding image identifier.
     + **Question:** The natural language question related to the image.
     + **Image Features:** The extracted visual features from Faster R-CNN.

**Visual Genome Dataset**

1. **Image Preprocessing:**
   * Followed the **same process as the VQA v2.0 dataset**, where images were processed using Faster R-CNN and stored in a PKL file.
2. **Scene Graph Data Processing:**
   * **Relationships and attributes** were extracted from the dataset and stored as **JSON files**.
   * **Cleaning Process:**
     + Removed entries with **empty values** in relationships and attributes.
3. **Processing During Training (Data Loader):**
   * Instead of preprocessing scene graph data beforehand, relationships and attributes were processed dynamically in the **data loader**.
   * Extracted the following graph components:
     + **Nodes:** Objects detected in the image.
     + **Edges:** Relationships between objects.
     + **Edge Attributes:** Additional attributes defining the edges (e.g., spatial relations).

proposed model/architecture

+---------------------------+

| Modified LXMERT |

| Architecture Diagram |

+---------------------------+

+-------------------+ +-------------------+

| VQA v2.0 Loop | | Visual Genome Loop|

| (Image + Text) | | (Scene Graphs) |

+-------------------+ +-------------------+

| | | |

| Image Input | | Scene Graph Input |

| (Faster R-CNN) | | (Visual Genome) |

| | | | | |

| Visual Features | | +---------------+|

| | | | | GNN ||

| v | | | (GAT/GCN) ||

| +------------+ | | +---------------+|

| | Visual | | | | |

| | Embeddings | | | Object |

| +------------+ | | Embeddings |

| | | | (Attributes, |

| | | | Spatial) |

| Text Input | | | |

| (Questions) | | Relationship |

| | | | Embeddings |

| v | | (Edges) |

| +------------+ | | | |

| | Language | | | v |

| | Embeddings | | | +------------+ |

| +------------+ | | | Scene Graph | |

| | | | | Embeddings | |

| v | | +------------+ |

| | | | |

+-------------------+ +-------------------+

| |

v v

+---------------------------+

| Fusion Layer |

| (Visual Features + Scene |

| Graph Embeddings) |

+---------------------------+

|

v

+---------------------------+

| Multi-Modal Transformer |

| (Shared Across Loops) |

+---------------------------+

|

v

+---------------------------+

| VQA Classifier |

| (Answer Prediction) |

+---------------------------+

 **Standard LXMERT Structure** (Vision + Language Modalities).

 **Modification for Visual Genome** (Scene Graph Input via GNN).

 **Separate Training Loops for VQA v2.0 and Visual Genome**.

 **Scene Graph Integration (Visual Genome Only)**

* A **Graph Neural Network (GNN)** processes **scene graphs** from Visual Genome.
* **Object embeddings** (attributes, spatial relations).
* **Relationship embeddings** (edges in the graph).
* GNN processes relationships using **Graph Attention Networks (GAT) or Graph Convolutional Networks (GCN)**.

 **Two Separate Training Loops**

* **VQA v2.0 Training**: Uses **image + text inputs** with Faster R-CNN.
* **Visual Genome Training**: Uses **scene graphs** with GNN.
* Both trained separately but share the **Multi-Modal Transformer**.

 **Final Fusion Before Prediction**

* Scene Graph embeddings are **merged with visual features**.
* Final representation passed to LXMERT's **cross-modality transformer**.
* Standard **VQA classifier** used for answer prediction.

Implementation  
**3. Model Combination Strategy**

**3.1. Confidence-Based Model Selection (CBMS) for VQA & VG**

To effectively merge the outputs of **LXMERT (VQA model)** and **Scene Graph Model (SG-VQA)**, we implement a **confidence-based decision mechanism**.

**How It Works:**

1. **Compute confidence scores** for both models:
   * Each model outputs **logits (raw prediction scores before softmax)**.
   * The confidence score is computed as the **softmax probability of the predicted answer**.

CVQA=max⁡(softmax(logitsVQA))C\_{\text{VQA}} = \max(\text{softmax}(\mathbf{logits}\_{VQA}))CVQA​=max(softmax(logitsVQA​))CSG=max⁡(softmax(logitsSG))C\_{\text{SG}} = \max(\text{softmax}(\mathbf{logits}\_{SG}))CSG​=max(softmax(logitsSG​))

1. **Select the more confident model** for final prediction:
   * If CVQA>CSGC\_{\text{VQA}} > C\_{\text{SG}}CVQA​>CSG​, use the VQA model’s answer.
   * Otherwise, use the Scene Graph model’s answer.

Pfinal=PVQA⋅1(CVQA>CSG)+PSG⋅1(CSG≥CVQA)P\_{\text{final}} = P\_{\text{VQA}} \cdot \mathbb{1}(C\_{\text{VQA}} > C\_{\text{SG}}) + P\_{\text{SG}} \cdot \mathbb{1}(C\_{\text{SG}} \geq C\_{\text{VQA}})Pfinal​=PVQA​⋅1(CVQA​>CSG​)+PSG​⋅1(CSG​≥CVQA​)

1. **Why This Approach?**
   * The **VQA model performs better on direct visual questions** (e.g., "What color is the car?").
   * The **Scene Graph model is superior for relational queries** (e.g., "Who is sitting next to the woman?").
   * By dynamically selecting the **more confident model per question**, we get the **best of both models**.

**4. Training Process**

The model is trained in **two phases** before combining them using CBMS.

**4.1. Training the VQA Model (LXMERT)**

* **Dataset:** VQA v2.0
* **Loss Function:** Cross-Entropy Loss
* **Optimizer:** AdamW (β1=0.9,β2=0.999\beta\_1 = 0.9, \beta\_2 = 0.999β1​=0.9,β2​=0.999)
* **Learning Rate Schedule:**
  + Warmup for **10% of total steps**, then decay using a **cosine scheduler**.
* **Batch Size:** 64
* **Epochs:** 10

**4.2. Training the Scene Graph Model (SG-VQA)**

* **Dataset:** Visual Genome Scene Graphs
* **GNN Architecture:**
  + Node embeddings: **128-dim features** (GloVe embeddings)
  + Edge embeddings: **64-dim features**
  + Aggregation: **Graph Attention Networks (GAT)**
* **Loss Function:** Cross-Entropy Loss
* **Optimizer:** Adam
* **Batch Size:** 32
* **Epochs:** 15

**5. Alternative Methods Considered**

Before finalizing **Confidence-Based Model Selection (CBMS)**, we explored several alternative fusion strategies:

**5.1. Late Fusion (Ensemble)**

* **Method:**
  + The final predictions were computed as **an ensemble of both models**.
  + The answer probabilities were averaged:Pfinal=αPVQA+(1−α)PSGP\_{\text{final}} = \alpha P\_{\text{VQA}} + (1 - \alpha) P\_{\text{SG}}Pfinal​=αPVQA​+(1−α)PSG​
* **Limitation:**
  + This method **failed to dynamically choose the better model**, reducing accuracy on relational questions.

**5.2. Multimodal Feature Fusion**

* **Method:**
  + Instead of selecting between models, we **concatenated** both outputs:hfinal=Concat(hVQA,hSG)h\_{\text{final}} = \text{Concat}(h\_{\text{VQA}}, h\_{\text{SG}})hfinal​=Concat(hVQA​,hSG​)
  + Trained a **meta-classifier** to determine the best answer.
* **Limitation:**
  + Increased computational complexity.
  + Required **additional training** on a combined dataset.

**6. Final Model Implementation**

After evaluating different fusion techniques, **CBMS was chosen for its simplicity, efficiency, and dynamic model selection capability**. The final pipeline consists of:

1. **Preprocessing:** Extract visual features (Faster R-CNN) and scene graphs (GNN).
2. **Training Phase 1:** Train LXMERT on VQA v2.0.
3. **Training Phase 2:** Train Scene Graph-based model on Visual Genome.
4. **Model Fusion:**
   * Compute **confidence scores** for each model.
   * Select the **higher confidence** model dynamically.
   * Output the final answer using the **logits from the selected model**.

Results

**Performance Comparison Across Datasets**

To evaluate the effectiveness of **scene graphs**, we conducted experiments using the **VQA v2.0 dataset** and the **Visual Genome (VG) dataset**. The results are summarized in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Input | Training | Validation |
| VQAv2.0 | Q + A + I | 28% | 25% |
| Visual Genome | Q + A + I | 29% | 24% |
| VG + Scene Graph | Q + A + I + A + R | 39% | 23% |

**Observations:**

🔹 **Baseline Performance (VQA v2.0 & VG)**

* Training accuracy for **VQA v2.0 and Visual Genome** remained close (~28-29%), showing similar learning capacity.
* Validation accuracy was **lower (24-25%)**, indicating overfitting on training data.

🔹 **Impact of Scene Graphs (VG + Scene Graph)**

* Training accuracy **jumped to 39%**, proving that **structured relationships help the model learn object dependencies better**.
* However, validation accuracy **remained similar (23%)**, suggesting a need for better generalization strategies.

**Final Model Performance (VQA + Scene Graph Model)**

After implementing **Confidence-Based Model Selection (CBMS)** to combine the **VQA v2.0-trained model** and the **VG Scene Graph-based model**, the accuracy improved significantly:

|  |  |
| --- | --- |
| Dataset | Accuracy |
| VQAv2.0 Sample1k | ~43% |
| Visual Genome Sample1k | ~59% |

**Key Takeaways**

✅ **Accuracy Improvement**

* The final **VQA + Scene Graph model outperformed the standalone models** on both datasets.
* **VQA v2.0 accuracy increased from 25% → 43%**, proving that **scene graphs enhance reasoning capabilities**.
* **Visual Genome accuracy improved significantly (59%)**, showing that scene graphs are highly effective in datasets rich in relational information.

✅ **Better Relational Understanding**

* The **VQA model struggled with relationship-based questions** ("Who is sitting next to the man?").
* By integrating **scene graphs**, the model **captured spatial and functional relationships**, leading to better answers.

✅ **Generalization & Challenges**

* While accuracy increased, **validation accuracy did not improve drastically** (23%), indicating **potential domain shift issues**.
* This suggests that while **scene graphs help in understanding relationships**, there is a **need for better fine-tuning strategies** for diverse question types.

**3. Why Did Scene Graphs Improve Performance?**

1️⃣ **Structured Representation:** Scene graphs provide **explicit object-relationship mappings**, reducing ambiguity in image understanding.  
2️⃣ **Better Context Understanding:** The model **learns spatial and functional relationships**, improving responses to complex queries.  
3️⃣ **Confidence-Based Selection Works:** CBMS ensures that **whichever model has higher confidence is used**, leading to **higher final accuracy**.

The VQA model learns deeper relationships from the VG dataset and can transfer this knowledge when answering questions on VQA v2.0.

It will handle unseen questions better due to structured learning.

**1. How do scene graphs help a VQA model?**

Scene graphs provide a structured representation of an image by describing objects, their attributes, and the relationships between them. A typical VQA model processes an image as raw pixels or high-level visual features (e.g., extracted using Faster R-CNN). However, these features lack explicit relational and structural information. Scene graphs introduce:

* **Object-Level Understanding**: They help the model understand not just what objects are in the image but also their attributes (e.g., "red ball") and relationships (e.g., "boy holding a ball").
* **Better Question-Answer Mapping**: Many VQA questions are relationship-dependent, such as *"What is on the table?"* or *"Who is holding the umbrella?"*. Scene graphs make answering such questions more direct.
* **Reduction of Ambiguity**: By explicitly defining relationships, the model avoids confusion in scenarios where multiple objects exist in the image.

**2. How will the model perform better if we add scene graphs?**

Adding scene graphs can enhance the model’s performance in multiple ways:

* **Improved Reasoning Capabilities**: Scene graphs enable relational reasoning, making the model better at answering complex questions that require understanding spatial and functional dependencies.
* **Better Generalization**: With structured knowledge, the model can generalize better to unseen questions by leveraging known object-relationship patterns.
* **Reduced Dependency on Large Datasets**: Traditional VQA models require massive datasets to learn implicit relationships. Scene graphs explicitly encode this information, reducing the data requirements.
* **Efficient Attention Mechanism**: Instead of attending to the entire image, the model can focus on relevant object relationships, improving accuracy and efficiency.

**3. How do Visual Genome dataset scene graphs help a VQA dataset-trained model?**

The **Visual Genome (VG) dataset** contains detailed scene graphs for images, whereas the **VQA v2.0 dataset** primarily provides image-question-answer pairs. By using scene graphs from VG, we can:

* **Enhance Object-Relationship Learning**: The VQA model learns deeper relationships from the VG dataset and can transfer this knowledge when answering questions on VQA v2.0.
* **Bridge the Gap Between Vision and Language**: Scene graphs act as an intermediate representation that aligns visual content with the structure of natural language questions.
* **Provide Explicit Structural Information**: Instead of forcing the model to infer relationships implicitly, we give it structured information, improving reasoning and accuracy.
* **Improve Interpretability**: A scene graph-based model can show **why** it made a certain prediction, making the results more interpretable and explainable.

**4. How does combining the VG scene graph-trained model with the VQA v2.0 dataset-trained model help prediction?**

A combined approach, where the model is trained separately on:

1. **VQA v2.0 dataset (image + question → answer)**
2. **Visual Genome dataset (scene graphs for structured representation of objects and relationships)**

...allows the model to:

* **Leverage Two Types of Knowledge**: The **VQA v2.0 model** understands direct image-to-answer mappings, while the **VG-trained model** enhances relational reasoning.
* **Improve Complex Question Answering**: Some questions require explicit reasoning (e.g., "Who is sitting next to the man in a hat?"). The scene graph-trained model helps answer these better.
* **Fuse Scene Graph Features for Multi-Modal Learning**: Instead of relying solely on image embeddings, the model incorporates structured object-relation embeddings.
* **Boost Accuracy in Relationship-Based Questions**: By training on VG scene graphs, the model can handle questions that involve **spatial, semantic, and functional relationships** better.

**5. What and how should I explain my project to anyone?**

To explain your **Visual Question Answering (VQA) project using scene graphs**, follow this structured approach:

**1. What is the project about?**

* My project focuses on improving Visual Question Answering (VQA), where a model takes an image and a question as input and provides an answer.
* Traditional VQA models use deep learning to extract features from images and map them to answers. However, they often struggle with complex reasoning tasks.
* To enhance reasoning, I integrate **scene graphs**, which explicitly represent objects, attributes, and relationships in an image.

**2. Why is it important?**

* VQA is an AI-complete problem, meaning it requires understanding **both vision and language**.
* Existing models fail in scenarios where **relationships** between objects are crucial (e.g., "What is the person in the red shirt holding?").
* Adding scene graphs improves relational reasoning, interpretability, and efficiency.

**3. How does my approach work?**

* **Step 1**: Train a model on the **VQA v2.0 dataset**, which provides images, questions, and answers.
* **Step 2**: Train another model using the **Visual Genome dataset**, which provides scene graphs (structured information about images).
* **Step 3**: Combine both models so that the VQA model can use scene graphs for better reasoning.
* **Step 4**: Modify the LXMERT transformer model to process scene graph embeddings along with traditional image features.

**4. What are the expected improvements?**

* The model will **answer complex questions more accurately** by understanding object relationships.
* **Better generalization**: It will handle unseen questions better due to structured learning.
* **More explainable AI**: Since scene graphs explicitly define object relationships, we can visualize how the model reaches an answer.

**5. Applications and Future Scope**

* **Assistive Technology**: Helps visually impaired users by answering questions about their surroundings.
* **Autonomous Systems**: Used in self-driving cars to understand relationships in traffic scenes.
* **Medical AI**: Can assist doctors by answering questions based on X-ray or MRI images.

Would you like a **presentation slide outline** or a **written project abstract** to complement this explanation? 😊

Here’s a structured **presentation slide outline** for your **VQA project using scene graphs**. This will help you present your work clearly and effectively.

**Slide 1: Title Slide**

* **Title:** "Enhancing Visual Question Answering with Scene Graphs"
* **Your Name**
* **Institution (if applicable)**
* **Date**

**Slide 2: Introduction to VQA**

* **What is Visual Question Answering (VQA)?**
  + AI model that takes an image and a question and generates an answer.
* **Why is VQA important?**
  + Useful in assistive AI, autonomous systems, and medical imaging.
* **Challenges in VQA**
  + Difficulty in understanding object relationships.
  + Struggles with complex reasoning.

**Slide 3: Limitations of Traditional VQA Models**

* Existing models use deep learning but lack explicit relational reasoning.
* Struggle with **questions requiring object relationships** (e.g., *"Who is holding the umbrella?"*).
* Rely heavily on **large-scale datasets** to learn implicit reasoning.

**Slide 4: What Are Scene Graphs?**

* **Definition:** A structured representation of an image where:
  + **Objects** (nodes) = *e.g., dog, ball, man*
  + **Relationships** (edges) = *e.g., "dog chasing ball" or "man holding umbrella"*
* **Example Visualization**
  + Show an **image** with its **scene graph**

**Slide 5: How Scene Graphs Improve VQA**

* **Explicit Relationship Understanding** (instead of inferring them from raw pixels).
* **Better Question-Answer Mapping** (e.g., "What is under the table?" is directly mapped to relationships).
* **Reduces Data Requirements** (by leveraging structured knowledge).
* **More Explainable AI** (users can see how the model reaches an answer).

**Slide 6: Dataset Used**

**VQA v2.0 Dataset**

* Image-question-answer triplets.
* Provides training data for general VQA tasks.

**Visual Genome Dataset**

* Contains **scene graphs** for images.
* Provides structured relationships for better reasoning.

**Slide 7: Model Architecture**

* **Base Model: LXMERT Transformer**
* **Two Training Steps:**
  1. Train on **VQA v2.0 dataset** (image + question → answer).
  2. Train on **Visual Genome dataset** (scene graphs + questions).
* **Fusion Strategy:** Combine scene graph embeddings with visual features.

**Slide 8: Implementation Details**

* **Libraries Used:**
  + PyTorch, Hugging Face Transformers, NetworkX (for scene graphs).
* **Hardware:**
  + **NVIDIA GeForce GTX 1650** for model training.
* **Preprocessing:**
  + Extract scene graphs using **Faster R-CNN** and **Graph Neural Networks (GNNs)**.

**Slide 9: Expected Improvements**

* **Better performance in complex reasoning questions.**
* **Improved accuracy on relational questions** (e.g., spatial & attribute-based).
* **Generalizes better to unseen questions.**
* **More interpretable results using scene graph visualization.**

**Slide 10: Results & Evaluation (Once Experiments Are Done)**

* **Comparison:**
  + Baseline LXMERT vs. Scene Graph-enhanced Model
* **Metrics Used:**
  + Accuracy, BLEU score, and human evaluation.
* **Visual Examples:**
  + Show improved model predictions using scene graphs.

**Slide 11: Applications & Future Work**

**Applications**

* **Assistive AI:** Helps visually impaired users understand images.
* **Autonomous Vehicles:** Helps self-driving cars reason about traffic scenes.
* **Medical AI:** Answers questions from medical images.

**Future Work**

* Integrate **more advanced GNN architectures**.
* Train on **larger multi-modal datasets**.
* Improve **real-time inference** capabilities.

**Slide 12: Conclusion**

* Scene graphs significantly improve **VQA reasoning capabilities**.
* Combining **VQA v2.0 & Visual Genome datasets** enhances predictions.
* The approach **reduces ambiguity, improves accuracy, and increases explainability**.
* **Future work can make VQA models even more powerful!**

**Slide 13: Q&A**

* Open the floor for questions.