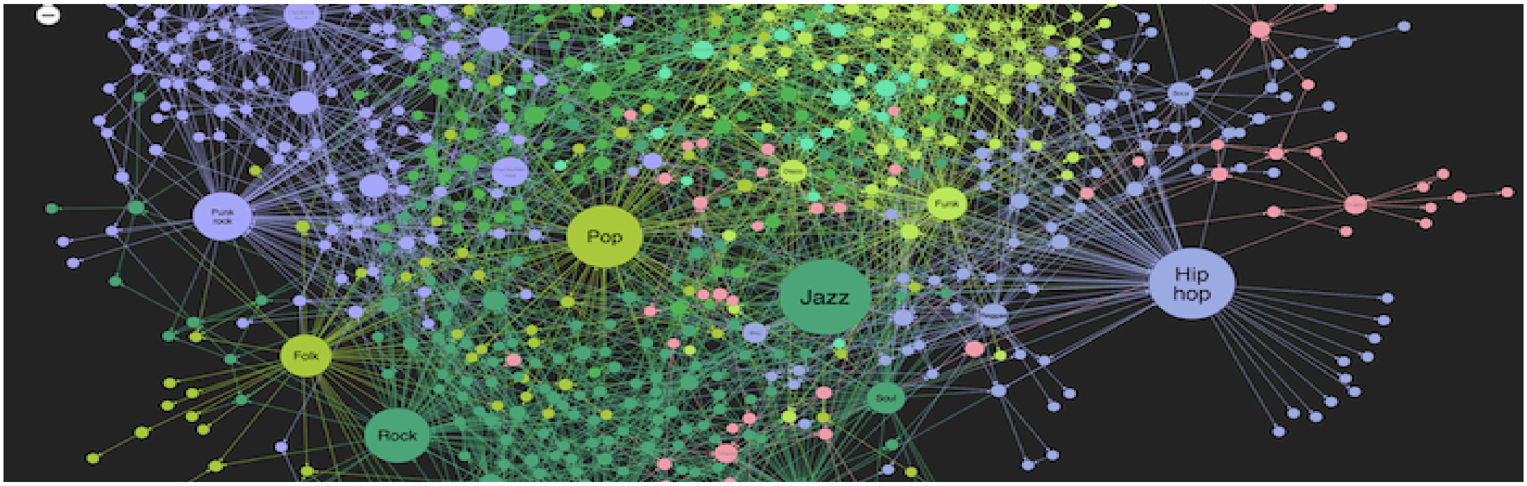
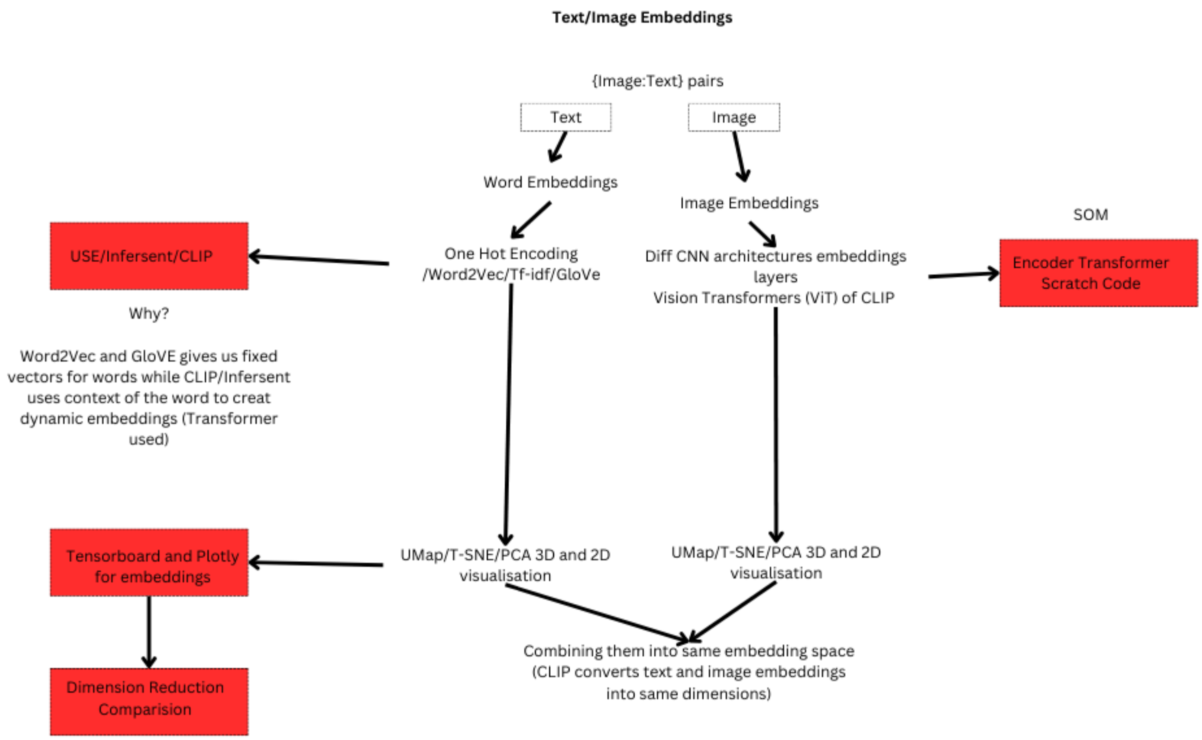


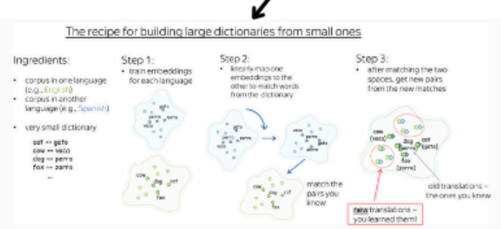
Part- 2 Knowledge Graph Embeddings

Part-1 Vector Embeddings

Work Done and modifications:

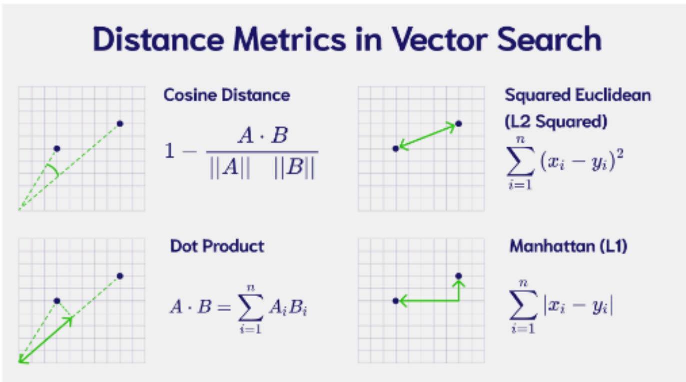


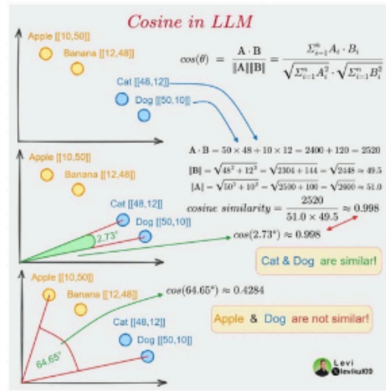
Linear Translation



Cosine Similarity Score/Euclidean Distance

Dimension Comparison methods used:



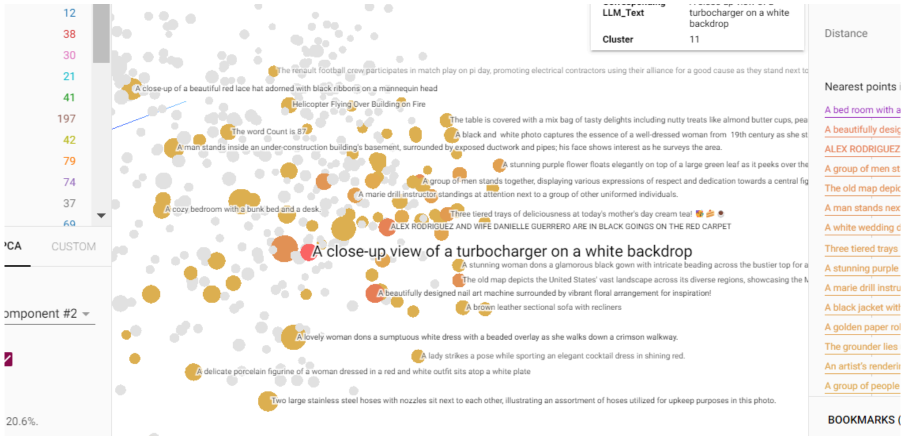


To be completed:

- Dimension Reduction technique Comparison- Advika
- Encoder Transformer Image/Text- Som
- Saving original & reduced embedding matrix as .csv files(Row index, 512 dimensions/3 dimensions then text and corresponding URL/Image pixels)- Simardeep
- Put images instead of dots in image embeddings (There should be image and corresponding text) [How to visualize feature vectors with sprites and TensorFlow's TensorBoard | by Andrew B. Martin | Looka Engineering | Medium]- Agam
- On hovering (changes to be made in metadata.tsv file)

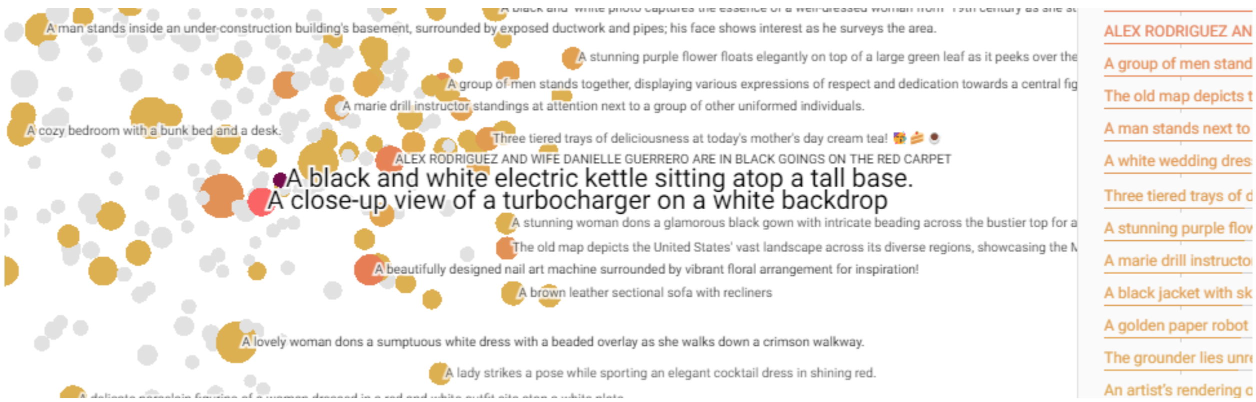
Output of Part-1

1. Text,Image original and reduced embeddings.
2. Tensorboard visualization of embeddings Image with corresponding text
3. Disparity between Image embeddings. 2 Images might be same in the 3d space but their corresponding text are quite different (object and person close to each other in text)
4. No contextual information about the image and text pair.



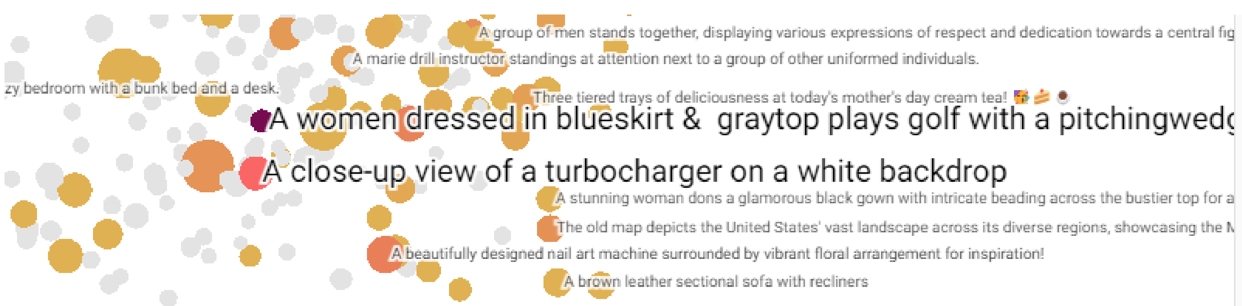
so, we have got the part of image embeddings with their corresponding LLM_Caption but this isn't fully accurate

If you see, for the image embedding of "Charger in white background" in the PCA reduced embedding space.



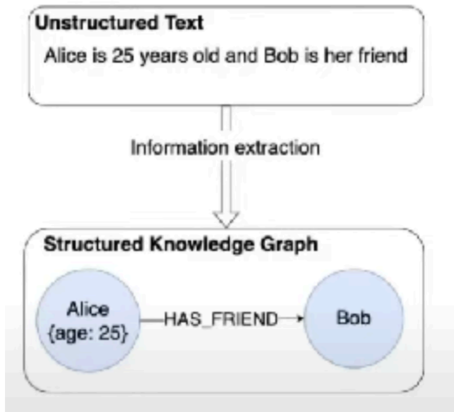
This embedding seems to be correct both are an object that too an electronic object but the problem is that it isn't true for all the other embeddings

If you see this example, how would a charger (an object) image be similar to a woman in a skirt?



Part2- Knowledge Graph Embeddings

Why do we need it?

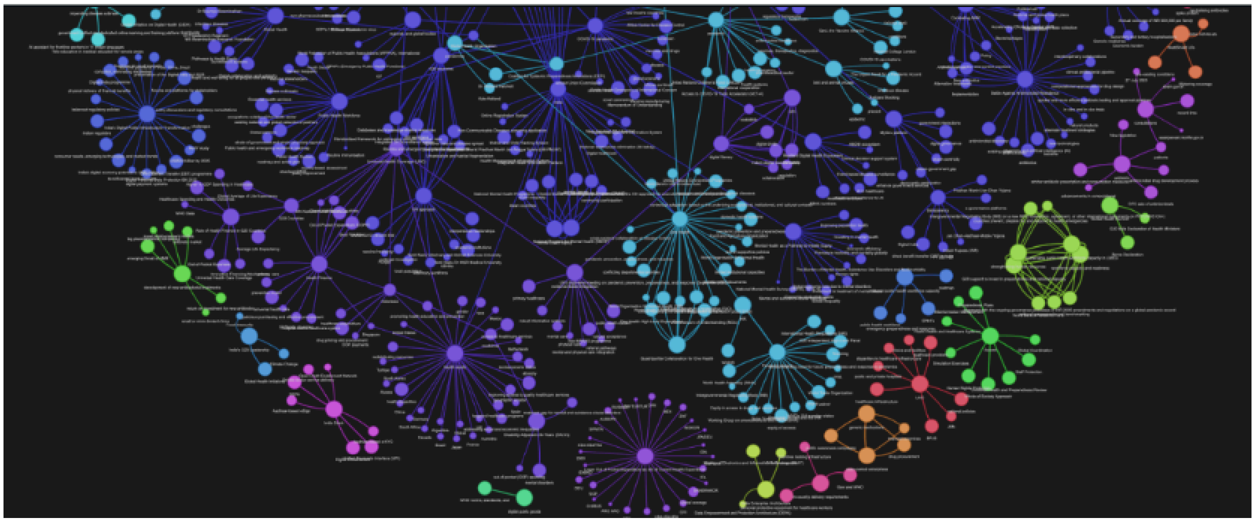


- Knowledge graph embeddings are designed to capture the relationships and structure within a graph of entities and their interconnections.
- They are used to represent entities and relationships in a continuous vector space, making it easier to perform various tasks such as link prediction, entity classification, and clustering.

GraphRAG: LLM-Derived Knowledge Graphs for RAG **(Very informative video, DON'T MISS THE END)**

Impact

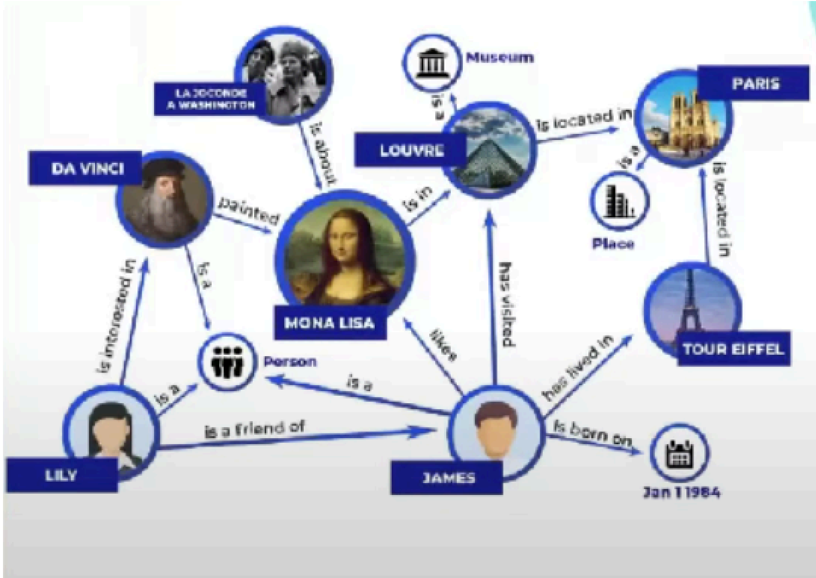
- **Relational Information**: Embeddings encode not just the entities (nodes) but also the relationships (edges) between them.
- **Graph Structure**: Capture the topological structure of the knowledge graph, including the types of relations and the paths between entities.
- **Inference and Reasoning**: Useful for tasks that require understanding and reasoning over the relationships and hierarchies within the graph.



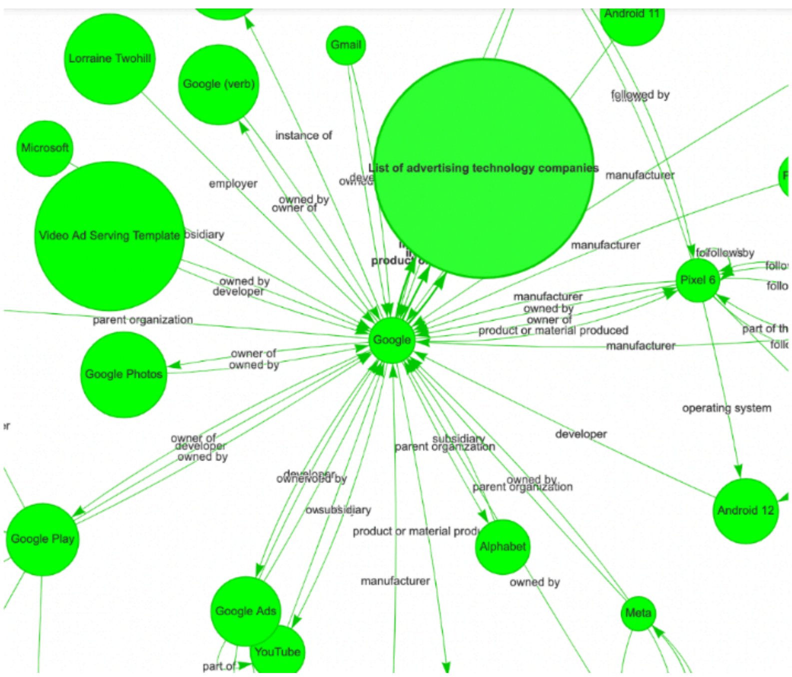
Use Cases for Knowledge Graph Embeddings with CLIP:

While CLIP embeddings provide powerful multimodal representations, knowledge graph embeddings can offer additional benefits, especially for tasks that require structured knowledge and relational reasoning. Here’s how they can complement each other:

1. **Enhanced Retrieval and Recommendation**:
○ Use knowledge graph embeddings to improve retrieval tasks by incorporating structured knowledge about entities and their relationships.
○ For example, if you have a knowledge graph of movies, actors, and genres, you can enhance image and text retrieval by leveraging relationships such as "actor in movie" or "movie belongs to genre".
2. **Contextual Augmentation**:
○ Enrich CLIP embeddings with knowledge graph information to provide deeper contextual understanding.
○ For instance, augment text descriptions with related entities and relationships from a knowledge graph to improve image-text alignment.
3. **Entity Disambiguation**:
○ Use knowledge graph embeddings to disambiguate entities in text descriptions.
○ If a text description mentions "Paris", knowledge graph embeddings can help distinguish between "Paris, France" and "Paris Hilton" based on relational context.



A sample knowledge graph of an article on Google:



Search and Information Retrieval

- Enhanced search engines
- Semantic search

Recommendation Systems

- Personalized recommendations
- Context-aware recommendations

Natural Language Processing (NLP)

- Entity recognition and linking
- Question answering

Fraud Detection and Risk Management

- Anomaly detection
- Risk assessment

This repo shows how to create a basic knowledge graph from sentence corpus:

- GitHub - rahulnyk/knowledge_graph: Convert any text to a graph of knowledge. This can be used for Graph Augmented Generation or Knowledge Graph based QnA

Library for creating graph nodes:

- NetworkX

Code and Intuition Resources:

- https://www.nlplanet.org/course-practical-nlp/02-practical-nlp-first-tasks/16-knowledge-graph-from-text
- https://paperswithcode.com/task/relation-extraction
- Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering (not necessary atm)
- https://arize.com/blog/evaluate-rag-with-llm-evals-and-benchmarking/ (LLM RAG response benchmarking with KGE)
- Node4j used to query, manage graph data

(Neo4j is a versatile and powerful graph database that offers a wide range of applications across various domains. By leveraging its graph-based data model and powerful querying capabilities, organizations can gain deeper insights, improve performance, and build sophisticated applications that are difficult to achieve with traditional databases. Integrating Neo4j with other technologies, such as machine learning models and multimodal embeddings, further enhances its utility and opens up new possibilities for advanced data analysis and decision-making.)

- NODES 2023 - Using LLMs to Convert Unstructured Data to Knowledge Graphs

- https://www.frontiersin.org/articles/10.3389/fnbot.2019.00093/full (IMPORTANT!)

https://medium.com/@shubham.shardul2019/end-to-end-multimodal-knowledge-graph-creation-from-texts-and-images-querying-in-natural-a28fa2053856

- Model that we will be using for Relation extraction will be REBEL

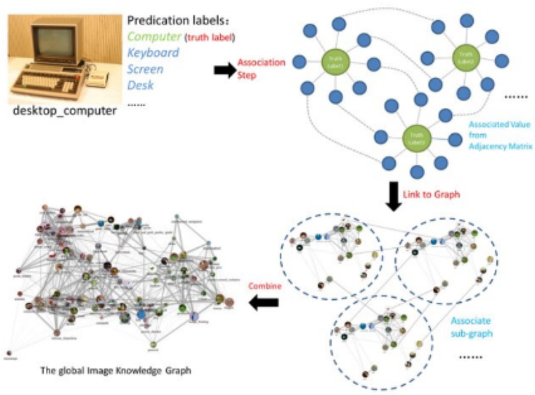
REBEL (Relation Extraction By End-to-end Learning) is a transformer-based model designed for relation extraction. Relation extraction is a natural language processing (NLP) task that involves identifying relationships between entities within a text.

The "large" in REBEL-large indicates that this is a larger, more complex version of the model, which typically means it has more parameters and thus can capture more nuanced patterns in the data.

Why is REBEL Used?

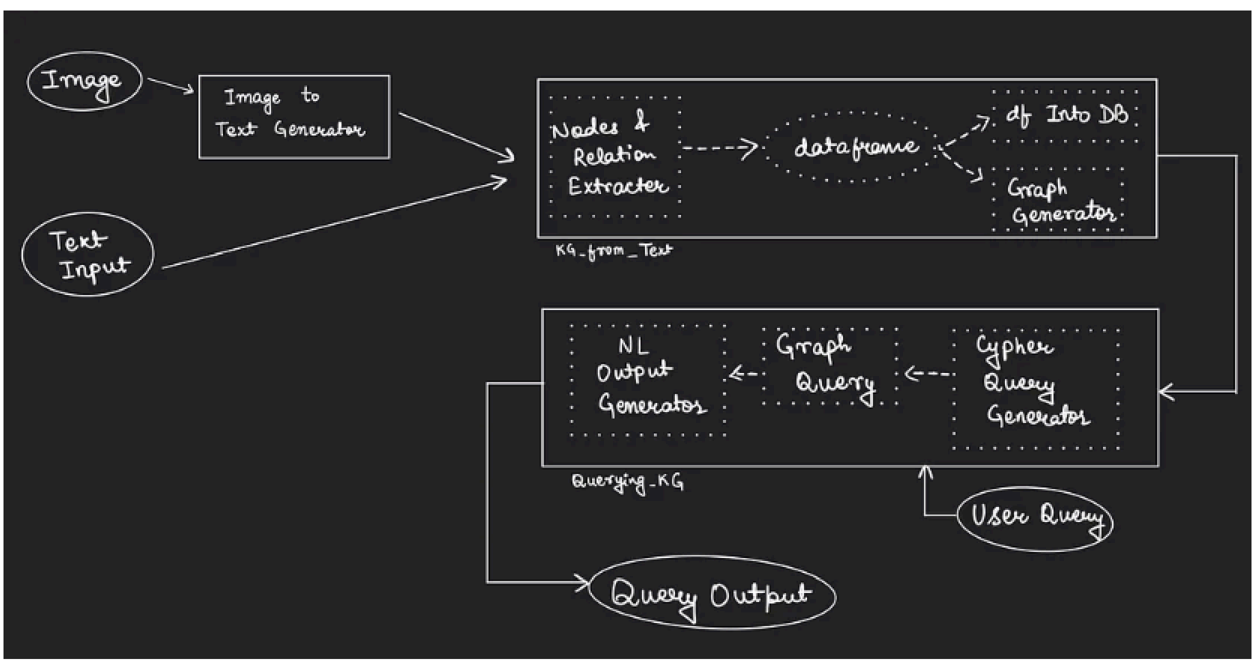
REBEL is used primarily for the task of relation extraction due to several reasons:

1. **End-to-End Learning**: Unlike traditional methods that might require multiple stages of processing, REBEL is designed to handle relation extraction in a single end-to-end framework. This simplifies the pipeline and can improve efficiency and performance.
2. **Transformer Architecture**: Being based on the transformer architecture, REBEL can leverage the advantages of transformers such as parallel processing and attention mechanisms. This allows it to handle large amounts of text and capture complex relationships between entities.
3. **Large-Scale Pretraining**: Models like REBEL-large benefit from pretraining on vast amounts of data, which helps them understand language patterns and relationships more effectively. This pretraining can significantly boost performance on downstream tasks like relation extraction.



OUR AIM?

Give an image- generate Knowledge Graph- KGE



While papers have shown use of CNNs to extract features from Images for KG, we aim to use Vision Transformers like CLIP to generate text from the image and create its knowledge graph

- By bringing together textual and visual relation extraction, we are particularly interested in dependencies between both modalities and how synergies lead to more robust representation learning
- knowledge graphs (KGs) is decomposed into two phases: (1) detecting the entities (or objects) as nodes, and (2) extracting relations between entities as edges
- The first phase can be reduced to a Named Entity Recognition for textual paragraphs (Lample et al., 2016) or Object Detection for images (Ren et al., 2015). Usually, the more challenging part receiving more attention is how to determine the relations between entities and is usually cast as a classification problem

Paper:

1. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering
2.