# A Graph-based Approach for Multi-Modal Question Answering from Flowcharts in Telecom Documents

Sumit Soman, H. G. Ranjani, Sujoy Roychowdhury, Venkata Dharma Surya Narayana Sastry, Akshat Jain, Pranav Gangrade, Ayaaz Khan
{sumit.soman,ranjani.h.g,sujoy.roychowdhury}@ericsson.com
Ericsson R&D
Bangalore, Karnataka, India

## Abstract

Question-Answering (QA) from technical documents often involves questions whose answers are present in figures, such as flowcharts or flow diagrams. Text-based Retrieval Augmented Generation (RAG) systems may fail to answer such questions. We leverage graph representations of flowcharts obtained from Visual large Language Models (VLMs) and incorporate them in a text-based RAG system to show that this approach can enable image retrieval for QA in the telecom domain. We present the end-to-end approach from processing technical documents, classifying image types, building graph representations, and incorporating them with the text embedding pipeline for efficient retrieval. We benchmark the same on a QA dataset created based on proprietary telecom product information documents. Results show that the graph representations obtained using a fine-tuned VLM model have lower edit distance with respect to the ground truth, which illustrate the robustness of these representations for flowchart images. Further, the approach for QA using these representations gives good retrieval performance using text-based embedding models, including a telecom-domain adapted one. Our approach also alleviates the need for a VLM in inference, which is an important cost benefit for deployed QA systems.

## CCS Concepts

• **Computing methodologies** → **Natural language processing**;
• **Applied computing** → *Document searching*.

## Keywords

Multi-Modal QA, Flowchart Representations, Retrieval Augmented Generation, RAG, Large Language Models, LLM, VLM, Telecom

**ACM Reference Format:**
Sumit Soman, H. G. Ranjani, Sujoy Roychowdhury, Venkata Dharma Surya Narayana Sastry, Akshat Jain, Pranav Gangrade, Ayaaz Khan. 2025. A Graph-based Approach for Multi-Modal Question Answering from Flowcharts in Telecom Documents. In *Proceedings of KDD '25 Workshop on Structured Knowledge for LLMs.* ACM, New York, NY, USA, 10 pages.


*KDD '25 Workshop on Structured Knowledge for LLMs, Toronto, CA*


## 1 Introduction

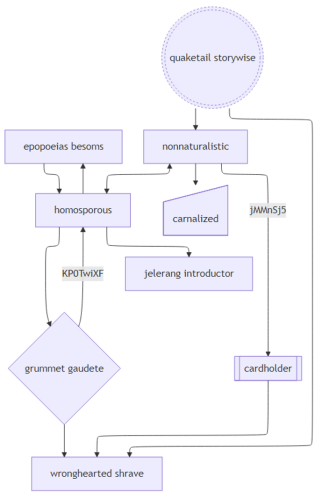
With advances in Large Language Models (LLMs), Retrieval Augmented Generation (RAG) based Question-Answering (QA) systems for technical domains like telecom have become prevalent [11, 12, 15], which leverage domain-adapted models [5, 13]. However, these works largely focus on text-based input and output modalities for RAG based QA task. A typical telecom engineer would require to refer to information present in Product Information Documents (referred to as PI Docs in this work) and assimilate information from images (or figures), tables and/or equations, apart from text, to provide answers to user queries, or for troubleshooting scenarios.

### 1.1 Background

In the literature, multi-modality pertaining to images interspersed with text are typically addressed as Visual QA (VQA) tasks that use Visual Language Models (VLMs) [6]. However, models trained on public datasets, which may include images of cat, dog, or background objects, are not useful for domains such as telecom. Typical images in telecom-domain documents pertain to categories such as screenshots of tools, flowcharts, block diagrams, sequence diagrams, icons, equations, schematic diagrams, among others, as can be seen in [1]. It is evident that images here often contain information primarily as text, and may also include icons, schematic representations, connections and dependencies among connected blocks or objects. The textual content is dominant and also domain-intensive.

Popular RAG systems either support text modality alone or use multi-modal embeddings in vector database to support images, in addition to text in the input. However, the cost of using VLMs for embedding, and as the generator for RAG in an inference setup, is formidable for business needs due to the large size of VLMs [8]. Approaches using knowledge graphs [3], as is common with textual data, require manual verification and hence are not suitable.

In this work, we focus on parsing, categorizing and processing images from proprietary technical documentation in the telecom domain. We specifically focus on flowchart images which are hard to interpret. These flowcharts present a unique issue of having mostly text-based content and are also indicative of decision rules and conditions. These, in turn, are a very valuable source of information, especially in configuration and troubleshooting of telecom networks. However, summarizing a flowchart as text via VLMs might be tedious, prone to hallucinations, difficult to quantify accuracy, and potentially may lose out on various conditions (refer Figure 1). Converting flowcharts to graph structures has been addressed in the earlier works of [10, 14]. However, these works are limited to evaluating the conversion of flowcharts to graphs and




Prompt: Provide a detailed description of uploaded image

Response: This image is a flowchart or process diagram that visually represents a series of interconnected elements through labeled shapes and directional arrows. Below is a detailed description:
---
### **General Layout**
1. The diagram begins at the top with a circular shape labeled **"quaketail storywise"**, which appears to be the starting point or central concept.
2. From this initial circle, various rectangular, trapezoidal, and diamond shapes are connected by arrows, indicating relationships or sequential steps.
3. The arrows flow in different directions, showing a non-linear progression, with some elements linking back to earlier nodes.
---
### **Key Components**
1. **Central Node:**
 - The circle labeled **"quaketail storywise"** is likely the main focus or origin point of the diagram.
 - It branches into two distinct paths:
 - **Left Path:** Leads to "epopoeias besoms."
 - **Right Path:** Leads to "nonnaturalistic."

2. **Left Path:**
 - "epopoeias besoms" connects to "homosporous" through two arrows.
 - From "homosporous," the flow moves downward to **"KP0TwiXF"** and ultimately to a diamond-shaped node labeled **"grummet gaudete"**.

3. **Right Path:**
 - "nonnaturalistic" connects to "carnalized."
 - Another arrow from "nonnaturalistic" leads downward to **"jMMnSj5"**, which in turn connects to **"cardholder."**
 - "carnalized" connects to "jelerang introductor," which also leads to "grummet gaudete."

4. **Diamond Node:**
 - The diamond-shaped node **"grummet gaudete"** acts as a junction where multiple paths converge.
 - It flows downward into two separate paths:
 - **"wronghearted shrave"**
 - A path that loops back upward, connecting to "cardholder."
---
### **Connections and Relationships**
- The diagram includes multiple feedback loops, where nodes like "cardholder" reconnect to earlier nodes in the flow, such as "wronghearted shrave" and "jelerang introductor."
- The layout suggests a system with a mix of linear and cyclical relationships, potentially representing processes or dependencies.
---
### **Visual Design**
- Shapes:
 - Circular node: Represents the central concept.
 - Rectangles and trapezoids: Represent intermediate or secondary ideas.
 - Diamond: Represents a decision point or convergence of multiple flows.
- Arrows: Represent directional flow or relationships between elements.
- Colors: All elements are shaded in light purple, with a consistent style used throughout.
---

**Figure 1: A sample description of synthetic flowchart from Flowlearn using GPT4 Vision model. The incorrect part of description are highlighted in gray. This illustrates the issues faced in using VLMs directly for technical QA involving flowcharts.**

evaluating the best representation of flowcharts (graphs or UML) for QA. The latter work, in fact, assumes that the correct flowchart is available for answering the question. Hence, there exists a gap in the study of retrieving the right flowchart or its representation. We build on some of these existing works to propose a solution which uses fine-tuned VLMs to convert flowcharts' images to graph representations, and benchmark it for retrieval for a typical QA task based on these flowcharts.

## 1.2 Problem Statement

We propose the enhancement of text-only RAG with the following capabilities:

- Categorize parsed images (from telecom documents - PI Docs) into image categories identified as relevant for telecom domain QA.
- Use a fine-tuned VLM to convert flowchart images to graph structures, using nodes connected through edges (unidirectional and/or bidirectional). The nodes and edges also have attributes associated.
- Jointly represent flowchart based graph structures interspersed with text using LM domain-adapted embeddings.
- Utilize the graphical structures in RAG pipeline for improved coverage during retrieval.

## 1.3 Contributions

The contributions of this work are as follows:

- Automatically categorize images parsed from technical documentation using fine-tuned Document Image Transformer (DIT) model [7].
- Convert flowchart images to graph structures using a fine-tuned VLM.
- Evaluate various chunking approaches to introduce these graphs in vector database for retrieval.
- Benchmark retrieval performance on QA dataset based on flowcharts.

The rest of the paper is organized as follows: Section 2 details the proposed approach, with details of image classification (Section 2.1), conversion of flowcharts to graphs (Section 2.2), chunking and ingestion (Section 2.3) and evaluation (Section 2.4). Experimental setup details are provided in Section 3 followed by the results and analysis in Section 4. Finally, conclusions and future work are presented in Section 5.

# 2 Proposed Approach

As mentioned earlier, QA on flowcharts can be challenging using VLMs directly (since they may comprise information related to flow of information between nodes, decisions based on conditions or sequence of steps). Evaluation of these textual description of flowcharts can be challenging as there is no ground truth available and these descriptions can tend to be verbose based on complexity of flowcharts (refer Fig 1).

Our proposed approach entails classification of images into various categories. Next, we consider only flowchart images. We use

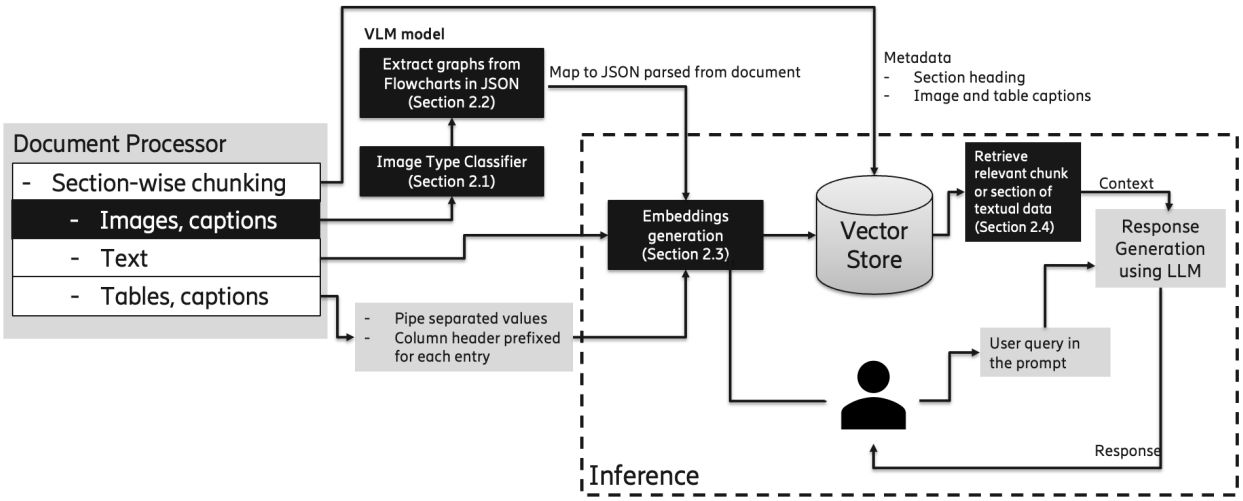



**Figure 2: The end-to-end approach for multi-modal QA, our approach for images is indicated in black-shaded boxes.**

fine-tuned VLMs (trained using publicly available database) to convert these domain-specific flowchart images to graph representations. In the subsequent step, we use text-based chunking and obtain embeddings of these for retrieval. This section details the proposed approach and evaluation metrics. Fig. 2 depicts the proposed end-to-end approach for multi-modal QA.

## 2.1 Classification of images

Telecom documentation can have various categories of images. Typically, PI Docs processing involves parsing various formats (such as HTML, PDF) of documents and extracting the text, tables, equations and images from the paragraphs of various sections. Typically, the textual components are chunked (optimally) converted to embedding vector using domain-adapted embedding model [16] and ingested into a vector database. In our approach, we aim to introduce only flowcharts using textual embedding for retrieval. To achieve this, we train a classifier to categorize the images parsed from PI Docs into various categories, *viz*., block diagrams, equations, flowcharts, graphs, hardware diagrams, icons (navigation, logos), schematic diagrams, screenshots, sequence diagrams. This helps categorize images for subsequent downstream QA task. In this work, we use the flowchart images for further processing and obtain their corresponding graph representation.

## 2.2 Graph representation for flowcharts

We filter on the images identified as flowcharts from the classifier model discussed in Section 2.1. On these, we propose to use a fine-tuned VLM to generate graph representations of the images.

A flowchart consists of a number of blocks and interconnecting links between them. We create a directed graph out of the flowchart - we represent each block as a node and capture the text within the block as a node attribute. Links between blocks are considered as edges, and any text on the link is considered as an edge attribute. Although flowcharts may have different shapes of blocks, we do not capture that in the node information. A sample representation of a flowchart and the corresponding graph (in JSON format) is shown in Figs. 3a and 3b, respectively.

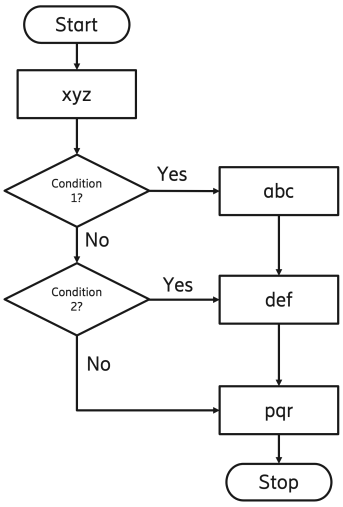
*2.2.1 Using a fine-tuned VLM.* In order to improve the graph representation generated by the open source VLM, we also fine-tuned a open-source VLM using the publicly available synthetic flowcharts of Flowlearn dataset [10].

It may be noted here that the model was fine-tuned on a publicly available dataset of flowcharts and used to generate graphs for telecom-domain flowcharts, as shown in Fig. 4. Details of VLM considered and training dataset are given in Section 3.

*2.2.2 Evaluation metric for VLM.* The output of the VLM is a graph representations of the flowchart. Hence, we use Graph Edit Distance (GED) [2] as a measure of the performance of the model, apart from number of nodes and edges accurately detected. A lower value of GED indicates close similarity of the generated graph representation when compared with the ground-truth representation of that flowchart. We show in our results that we obtain lower GED with the fine-tuned VLMs.

## 2.3 Chunking, ingestion into vector store and retrieval

The graph structures obtained from the previous step (detailed in Section 2.2) must be introduced into the vector store to ensure these are included in the retriever stage of RAG. Hence, it is essential to be able to obtain embedding vectors for these graph structures. There have been studies which perform experiments to find optimal chunking of textual data and for tables to improve retrieval accuracy [16]. Similarly, it is of importance to understand the optimal




**(a) A sample image of a flowchart.**


```
{
  "nodes": [
    {
      "id": "start",
      "label": "Start"
    },
    {
      "id": "xyz1",
      "label": "xyz"
    },
    {
      "id": "condition1",
      "label": "Condition1?"
    },
    {
      "id": "abc",
      "label": "abc"
    },
    {
      "id": "condition2",
      "label": "Condition2 ?"
    },
    {
      "id": "def",
      "label": "def"
    },
    {
      "id": "pqr",
      "label": "pqr"
    },
    {
      "id": "stop",
      "label": "Stop"
    }
  ],
{"edges": [
    {
      "from": "start",
      "to": "xyz1",
      "value": " "
    },
    {
      "from": "xyz",
      "to": "condition1",
      "value": " "
    },
    {
      "from":
"condition1",
      "to": "abc",
      "value": "Yes"
    },
    {
      "from":
"condition1",
      "to": "condition2",
      "value": "No"
    },
    {
      "from": "abc",
      "to": "def",
      "value": " "
    },
...
...
    {
      "from": "condition2",
      "to": "def",
      "value": "Yes"
    },
    {
      "from": "condition2",
      "to": "pqr",
      "value": "No"
    },
    {
      "from": "def",
      "to": "pqr",
      "value": " "
    },
    {
      "from": "pqr",
      "to": "stop",
      "value": "edge8"
    }
  ]
}
```


**(b) Graph representation in JSON (JavaScript Object Notation) with nodes and edges.**

**Figure 3: A sample flowchart and its graph representation in JSON format.**

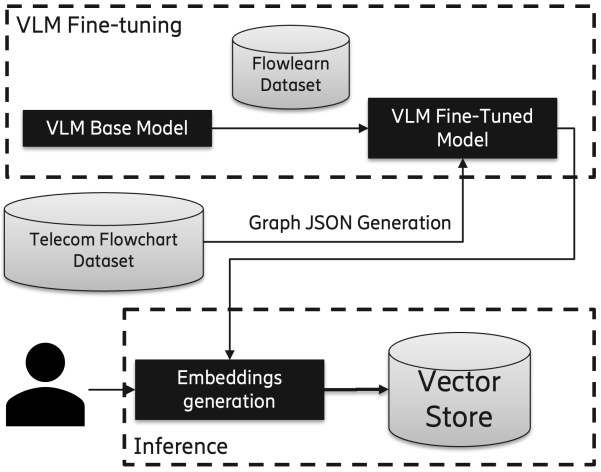



**Figure 4: Fine-tuning the VLM with a publicly available dataset and using it on telecom data for generating graph representations.**

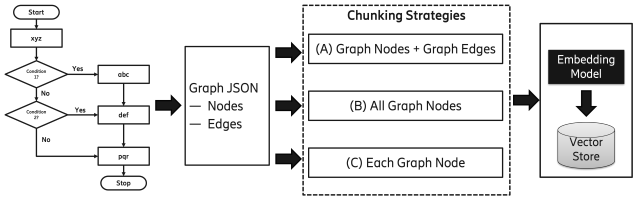



**Figure 5: Chunking approaches for flowchart JSONs.**

chunking mechanism for embedding graph structures. We consider the following options to generate embeddings, as shown in Fig. 5:

- **Each node as one chunk**: Embed *each node's textual information* as a single embedding vector.
- **All nodes as one chunk**: Embed *all the node textual information* as a single embedding vector.
- **Entire graph JSON as one chunk**: Embed the *entire textual information* from graph JSON as a single embedding vector.

## 2.4 Evaluation of Retrieval for RAG

We evaluate retrieval using two embedding models, the *bge-large* [18] which is a publicly available embedding model and the *TeleRoBERTa* [4], a telecom-domain adapted embedding model. For evaluating the retrieval, we compute top-$k$ accuracy for $k = \{1, 3, 5\}$. Following is the evaluation criteria adopted for the respective chunking approaches:

- When textual information of *each node* is embedded as a vector, the retrieval is correct if *any* of the nodes from the top-$k$ retrieved graph JSONs appears in the ground truth.
- When textual information of *all the nodes* is embedded as a single vector, the retrieval is considered correct if *all* of the nodes from the ground truth appear in the top-$k$ retrieved graph JSONs.
- When the *entire textual information* in the graph JSON is embedded as a single vector, the top-$k$ retrieval is considered correct one of the graph JSON among the $k$- retrieved graphs corresponds to the ground truth.

# 3 Experiments

## 3.1 Datasets

For the dataset creation, we initially considered two sources - (i) publicly available 3GPP (Rel 18) documents (ii) proprietary PI Docs - both of which pertain to the telecom domain.

*3.1.1 Image Sources.* From the 36 series of the publicly available 3GPP (Rel 18) documents [1], we parse 6342 images comprising of various categories such as schematic diagrams, graphs, frequency plots, block diagrams, sequence diagrams. However, there are very few examples of flowcharts (less than 2%) and many of these can be considered as a mix of flowchart as well as block diagram (refer Fig. 9 in Appendix B). Hence, we do not consider the flowcharts from this source, as it is not representative of the data we encounter during troubleshooting.

From the proprietary PI Docs, we have identified few documents to include in this dataset, extracted 1586 images of multiple types such as block diagrams, equations, flowcharts, graphs, hardware diagrams, icons (navigation, logos), schematic diagrams, screenshots, sequence diagrams, among others. Fig. 8 shows sample images for the categories of images seen in PI Docs. Since this is a proprietary dataset, all content in images pertaining to PI Docs source have been obfuscated in this manuscript to retain confidentiality. Table 2 shows the image categories considered and their statistics. We have manually annotated this dataset to construct training and test sets with 1268 and 318 images respectively using stratified sampling across these image types. This dataset was used to fine-tune the classifier model that predicted the image type.

*3.1.2 Image to graph representation.* For fine-tuning, we consider synthetic flowcharts component of Flowlearn dataset [10] for training VLM model. This consists of 10,000 flowchart images generated with *Mermaid* tool. The mermaid tool script is mapped to the required JSON format and is used for fine-tuning. However, we found that this dataset did not consider the following scenarios typically seen in flowcharts:

- **Shapes of nodes** such as rectangles with semicircular ends, parallelograms, decision boxes (typically shaped as rhombus), connectors (shaped as circles and pentagons)
- **Edges,** including bi-directional edges, multiple arrow heads (small, medium, large), various edge connectors (solid, dotted, dashed), straight sharp edges
- **Node lines** - solid, dotted, dashed and sometimes, no outer lines
- **Edge attributes** with text associated

In order to deal with such scenarios, we synthetically created images using *Mermaid* tool, using existing Flowlearn dataset as the starting point. These synthetically generated images were augmented with existing images of Flowlearn dataset. The augmented images for training (fine-tuning) and testing are kept separate.

*3.1.3 QA Dataset for Retrieval.* For testing retrieval accuracy, we consider 105 flowchart images from PI Docs which have been associated with ground truth graph structures. With inputs from Subject Matter Experts (SMEs), we carefully curate a set of 502 QA pairs from these images, with a mode of $\sim 5$ questions for each image. Each QA pairs is tagged as 'Decision related' (D), 'Edge related' (E) and 'Node related' (N) - based on how to arrive at the answer from the question. Details of the QA dataset are listed in Table 1 for different hops. The categories listed are based on ability of retriever to identify the correct chunk of data which contains the graphical structure from the flowchart image.

| Category | # of QA |
|---|---|
| Decision related | 359 |
| Node related | 487 |
| Edge related | 479 |

**Table 1: Distribution of QA categories considered**

## 3.2 Experimental Setup

*3.2.1 Image Category Classifier.* We use the manually annotated PI Docs image dataset (1586 images) described in Section 3.1 to train (fine-tune) the image classification model. This dataset is split into train and test dataset in the ratio 80-20 split (1268-318 train-test split) using stratified sampling. We fine-tune the "*microsoft/dit-base*" model [7] with batch size of 16 for this dataset.

While it is possible to explore models other than DIT, we note that image classification is not the primary focus of this work and this model performance can be considered as a baseline for further improvements.

*3.2.2 VLM for graph representation.* The top-performing open-source VLMs [1] available at the time of conducting our experiments were *Qwen2-VL* [17] and *Llava 1.5* [9]. We considered *Qwen2-VL* for fine-tuning due to better performance on few samples. The prompt used for generating the graph representation of a flowchart is:

> *"I have uploaded an image of a flowchart and here is its ground truth JSON representation, image_json ={}. Now generate JSON for the next image, from and to should be the node IDs. In the edges section, make sure that the edge value is present. If there are multiple identical nodes, create different IDs for them and their edges accordingly."*

The synthetic flowcharts from Flowlearn dataset [10] is considered for fine-tuned VLM. It has 10,000 images, split as 64-16-20% for train, validation and test respectively. We augment this training set with synthetic data for improved coverage of various nodes and edges (details described in 3.1). The fine-tuning was performed for few choices of parameters $R$ and $\alpha$, and the best fine-tuned model was used in the pipeline.

*3.2.3 Retrieval with chunking approaches.* As detailed in Section 2.4, we evaluate various chunking approaches via retrieval accuracies. Three chunking approaches are proposed and evaluated for top-$k$ accuracies. The chunking approaches are evaluated in two scenarios: (i) Embedding vectors of only graph-structures are considered for retrieval (ii) Embeddings of graph-structure and accompanying text are considered for retrieval. This results in 6 variations of retrieval for each model. We consider two embedding models - *bge-large* and

[1] Since PI Docs are proprietary in nature, it is preferable to use open-source VLMs to avoid data-sharing outside the organization

TeleRoBERTa (domain-adapted) [4, 18] and hence will have 12 sets of retrieval results.

## 4 Results and Analysis

In this section, we tabulate and detail results of the experiments listed in experimental setup section.

### 4.1 Image Category Classifier

We consider dataset of 1586 images from the proprietary dataset and labeled them manually to create training and test datasets. The distribution of number of images in the respective categories is shown in Table 2. We fine-tuned the "*microsoft/dit-base*" model [7] with batch size of 16 for this dataset.

| S. No. | Class | #Train | #Test | Total |
|---|---|---|---|---|
| 1. | Block Diagram | 123 | 39 | 162 |
| 2. | Equation | 154 | 42 | 196 |
| 3. | Flowchart | 171 | 38 | 209 |
| 4. | Graph | 41 | 9 | 50 |
| 5. | Hardware | 7 | 3 | 10 |
| 6. | Icon | 15 | 3 | 18 |
| 7. | Others | 131 | 43 | 174 |
| 8. | Schematic Diagram | 168 | 29 | 197 |
| 9. | Screenshot | 388 | 102 | 490 |
| 10. | Sequence Diagram | 70 | 10 | 80 |
| | Total | 1268 | 318 | 1586 |

**Table 2: Image categories for telecom dataset.**

The fine-tuned model was evaluated on the test set for predicting the image categories. The performance of the model on various categories of images is shown in Fig. 6. We observe that the accuracy of prediction is high for images categories like icons and equations, while it is above 80% for sequence diagrams, screenshots and flowcharts.

We also show the confusion matrix for the test set indicating the correct and incorrect classification of images in the test set in Fig. 7. We observe that most of the flowchart images are classified correctly, while some are misclassified as block diagram, schematic diagram or others. This is expected since these images can be similar and belong to multiple categories. However, since most of the flowchart images are categorized correctly, we use this classifier in the pipeline. The performance can potentially be improved as more annotated images are available for fine-tuning.

### 4.2 VLM for graph representation

The results for fine-tuning of the *Qwen2-VL* [17] VLM are shown in Table 3, for various parameter values. The columns indicate the average number of nodes and edges in the ground truth graph JSON representations, the number of nodes after the transformation operations and the number of edges detected for the model outputs. The GED metric is shown in the last column. We obtain the lowest GED of 2.74 using both Lora R and $\alpha$ as 512 for the finetuned *Qwen2-VL* model, which is a significant improvement over the base model which has a GED of 10.21 on the test set.

Similar performance evaluation of flowcharts from PI Docs is reported using the best performing fine-tuned model in Table 3. We observe that the average number of nodes and edges in the flowcharts from PI Docs is almost twice that of those seen in Flowchart dataset. Results show that GED on this unseen flowchart data is quite low (3.14).

### 4.3 Retrieval with chunking approaches

Table 4 shows the retriever performance on top-$k$ accuracy for the chunking approaches using a publicly available base model (*bge-large*) and domain-adapted (*TeleRoBERTa*) embedding models. We highlight here that the *TeleRoBERTa* model has been domain-adapted on publicly available telecom data (3GPP), and does not include any images related information during its training phase. Hence, there is no data contamination for the data considered and evaluated in this work. The objective here is to compare it[2] with a publicly available embedding model. We observe that the best top-$k$ retrieval results for $k = \{1, 3\}$ are obtained when using for *TeleRoBERTa* (57.17% and 71.91% respectively). Further, using each node as one chunk (embedding vector) gives better results for top-1 (57.17%) for *TeleRoBERTa*.

We also observe that retrieval accuracy reduces in the scenario when embeddings of graph structures are interspersed with text in the vector store. This is expected, as in a typical scenario, the retrieved top-$k$ embeddings are no longer limited to only graph structures.

Table 5 shows the top-$k$ retrieval accuracy for the embedding models for the various QA categories. Across the chunking strategies, higher performance is most commonly seen when using the entire graph JSON as one chunk, for both models. Better performance is obtained for node-related questions, followed by decision-related and finally, edge-related questions.

## 5 Conclusions and Future Work

In this work, we have considered an approach to introduce flowchart images with dominant textual content into retrieval (for RAG) using textual embeddings. We first categorize images present in the domain dataset using a fine-tuned DIT model. We observe that the accuracy of flowchart category of images is sufficiently high. Next, the flowchart images are converted to graph structures using a fine-tuned VLM. Here, we show that the fine-tuned VLM has lower GED for the flowchart graph representations. These graph structures are then embedded into text-based vector store and benchmarked for retrieval accuracy on a QA dataset based on these graphs. We observe that embedding the whole graph as one vector shows higher accuracy. This is because the chunk of data embedded includes all the information related to the nodes and edges of the graph. This is also shown to have better performance when the QA category is node-related.

Future work includes evaluation of generator output when JSON structures are passed as context to textual generator component of RAG. Analyzing cases involving errors in graph generation and retrieval, and performance with interspersed (document) text would be of interest. Additionally, extending the capabilities to other types

[2] *TeleRoberta* model is much smaller than *bge-large* model in terms of parameter size.

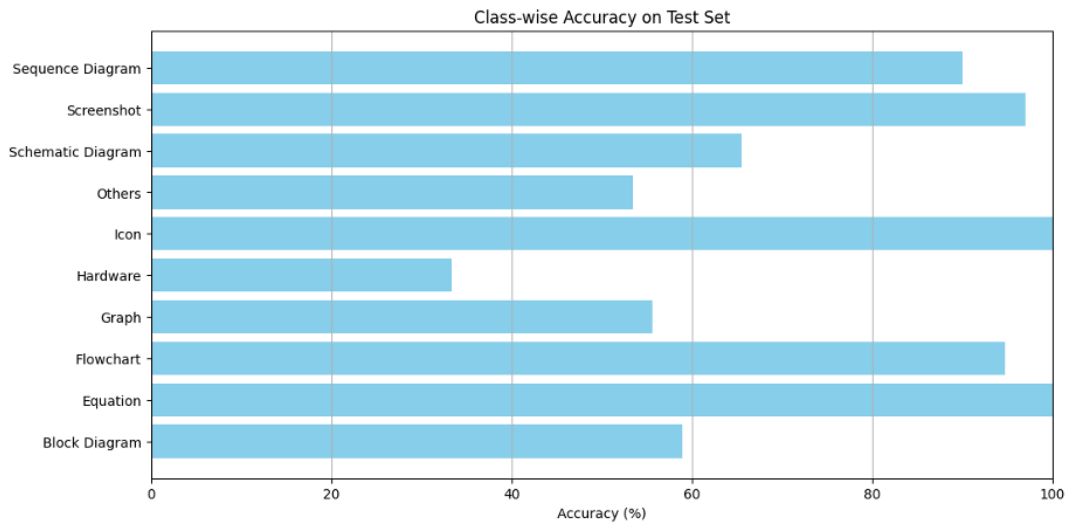



**Figure 6: Performance of the classifier on the test set for various image categories from PI docs.**

| Model | Avg. #Nodes (Ground Truth) | Avg. #Edges (Ground Truth) | Avg. #Nodes Detected | Avg. #Edges Detected | Avg. Graph Edit Distance (GED) |
|---|---|---|---|---|---|
| Flowcharts from test set of Flowlearn dataset | | | | | |
| Qwen2-VL Base | 6.36 | 6.82 | 6.55 | 8.57 | 10.21 |
| Qwen2-VL FT - Lora R=8, Alpha=16 | 6.36 | 6.82 | 6.3 | 6.23 | 4.24 |
| Qwen2-VL FT - Lora R=32, Alpha=32 | 6.36 | 6.82 | 6.2 | 6.53 | 5.09 |
| Qwen2-VL FT - Lora R=128, Alpha=128 | 6.36 | 6.82 | 6.34 | 6.43 | 4.43 |
| Qwen2-VL FT - Lora R=256, Alpha=256 | 6.36 | 6.82 | 6.35 | 6.7 | 3.82 |
| Qwen2-VL FT - Lora R=512, Alpha=512 | 6.36 | 6.82 | 6.36 | 6.42 | **2.74** |
| Flowcharts from PI Docs | | | | | |
| Qwen2-VL FT - Lora R=512, Alpha=512 | 12.54 | 11.77 | 12.32 | 11.11 | **3.14** |

**Table 3: Graph Metrics for fine-tuned VLM with various parameter settings reported for test set of Flowlearn dataset and flowcharts from PI Docs.**

| Embedding Model | bge-large | | | TeleRoBERTa | | |
|---|---|---|---|---|---|---|
| Chunking Approach | Top-1 | Top-3 | Top-5 | Top-1 | Top-3 | Top-5 |
| Embeddings of graph structures only for retrieval | | | | | | |
| Each node as one chunk | **56.17**% | 65.33% | 66.93% | **57.17%** | 62.74% | 65.93% |
| All the nodes as one chunk | 50.29% | 68.12% | 75.23% | 49.80% | **71.91%** | **76.89**% |
| Entire graph JSON as one chunk | 53.19% | **71.12**% | **78.29%** | 49.06% | 71.71% | 76.69% |
| Embeddings of graph structures interspersed with text for retrieval | | | | | | |
| Each node as one chunk | **41.05** | **55.53** | **59.76** | **32.42** | **44.08** | **48.86** |
| All the nodes as one chunk | 38.84 | 43.63 | 43.82 | 30.92 | 38.33 | 42.29 |
| Entire graph JSON as one chunk | 30.08 | 31.47 | 32.03 | 24.27 | 28.15 | 39.83 |

**Table 4: Retriever performance of chunking approaches for the embedding models, best top-$k$ values are indicated in bold for (i) only embeddings of graph structures considered for retrieval, and (ii) with embeddings of graph structures and text considered for retrieval.**

of diagrams like UML Sequence diagrams which have a semantic structure are areas of potential future research for the community.

## References

[1] 3GPP. 2022. *3GPP Release 18.* Technical Report. Accessed: 2024-05-19.
[2] Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In *4th International Conference on Pattern Recognition Applications and*

| Chunking Approach | Top-1 | | | Top-3 | | | Top-5 | | |
|---|---|---|---|---|---|---|---|---|---|
| Question Category | *D* | *N* | *E* | *D* | *N* | *E* | *D* | *N* | *E* |
| bge-large | | | | | | | | | |
| Each node as one chunk | 46.11 | **59.20** | 40.67 | 63.72 | **77.53** | 60.03 | 66.22 | **81.43** | 63.61 |
| All the nodes as one chunk | 46.96 | 52.44 | 47.88 | 65.12 | 68.27 | 65.61 | 70.33 | 78.19 | 66.42 |
| Entire graph JSON as one chunk | **47.35** | 52.57 | **51.98** | **67.97** | 70.43 | **70.15** | **71.49** | 77.41 | **77.45** |
| TeleRoBERTa | | | | | | | | | |
| Each node as one chunk | 43.21 | **56.71** | 37.02 | 58.12 | **73.72** | 55.73 | 63.32 | **78.55** | 59.61 |
| All the nodes as one chunk | **47.13** | 50.29 | 49.77 | 61.36 | 65.22 | 64.44 | 67.80 | 73.41 | 65.16 |
| Entire graph JSON as one chunk | 46.75 | 49.20 | **51.87** | **65.64** | 66.09 | **69.08** | **71.22** | 72.34 | **70.77** |

**Table 5: *D*, *N*, *E* indicative of Decision based, Node based and Edge based QA. Retriever performance of chunking approaches for the two embedding models. Best top-*k* values are indicated in bold for various chunking approaches.**

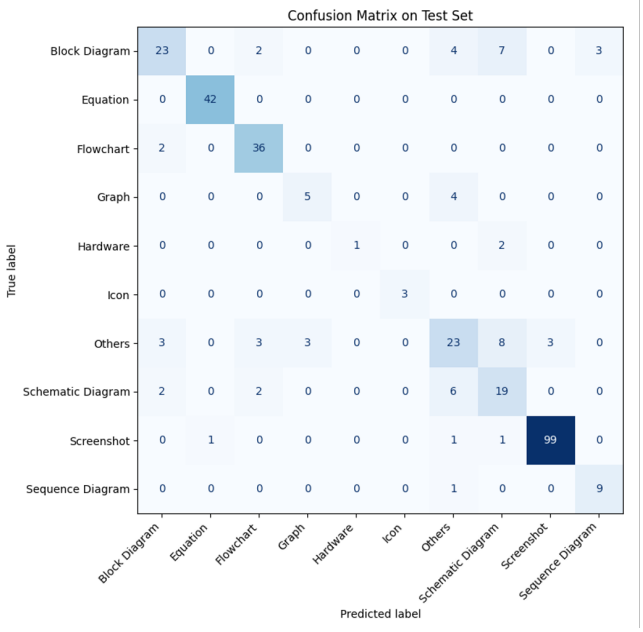



**Figure 7: Confusion matrix on the test set for categorization of images from the PI dataset.**

*Methods 2015.*

[3] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309* (2024).

[4] Henrik Holm. 2021. Bidirectional encoder representations from transformers (bert) for question answering in the telecom domain.: Adapting a bert-like language model to the telecom domain using the electra pre-training approach.

[5] Athanasios Karapantelakis, Mukesh Thakur, Alexandros Nikou, Farnaz Moradi, Christian Olrog, Fitsum Gaim, Henrik Holm, Doumitrou Daniil Nimara, and Vincent Huang. 2024. Using large language models to understand telecom standards. In *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE, 440–446.

[6] Aryan Keskar, Srinivasa Perisetla, and Ross Greer. 2025. Evaluating multimodal vision-language model prompting strategies for visual question answering in road scene understanding. In *Proceedings of the Winter Conference on Applications of Computer Vision*. 1027–1036.

[7] David D. Lewis, Gady Agam, Shlomo Engelson Argamon, Ophir Frieder, David A. Grossman, and Jefferson Heard. 2006. Building a test collection for complex document information processing. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006).

[8] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13299–13308.

[9] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[10] Huitong Pan, Qi Zhang, Cornelia Caragea, Eduard Dragut, and Longin Jan Latecki. 2024. Flowlearn: Evaluating large vision-language models on flowchart understanding. In *ECAI 2024*. IOS Press, 73–80.

[11] Sujoy Roychowdhury, Nishkarsh Jain, and Sumit Soman. 2024. Unlocking telecom domain knowledge using llms. In *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 267–269.

[12] Sujoy Roychowdhury, Sumit Soman, HG Ranjani, Neeraj Gunda, Vansh Chhabra, and Sai Krishna Bala. [n. d.]. Evaluation of RAG Metrics for Question Answering in the Telecom Domain. In *ICML 2024 Workshop on Foundation Models in the Wild*.

[13] T. Saraiva, M. Sousa, P. Vieira, and A. Rodrigues. 2025. Telco-DPR: A Hybrid Dataset for Evaluating Retrieval Models of 3GPP Technical Specifications. In *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. 01–06. doi:10.1109/WCNC61545.2025.10978393

[14] Shubhankar Singh, Purvi Chaurasia, Yerram Varun, Pranshu Pandya, Vatsal Gupta, Vivek Gupta, and Dan Roth. 2024. Flowvqa: Mapping multimodal logic in visual question answering with flowcharts. *arXiv preprint arXiv:2406.19237* (2024).

[15] Sumit Soman and Ranjani HG. 2023. Observations on LLMs for telecom domain: capabilities and limitations. In *Proceedings of the Third International Conference on AI-ML Systems*. 1–5.

[16] Sumit Soman and Sujoy Roychowdhury. [n. d.]. Observations on Building RAG Systems for Technical Documents. In *The Second Tiny Papers Track at ICLR 2024*.

[17] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191* (2024).

[18] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

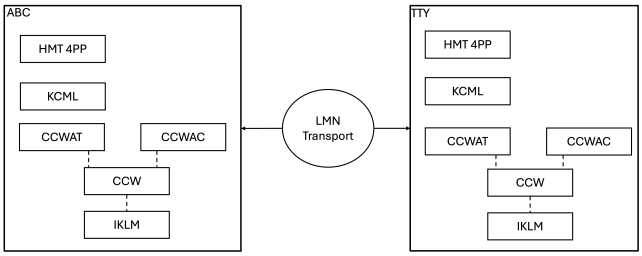
## A Image Categories in Telecom Dataset

Representative examples of various types of images present in the telecom (PI) dataset are shown in Fig. 8.

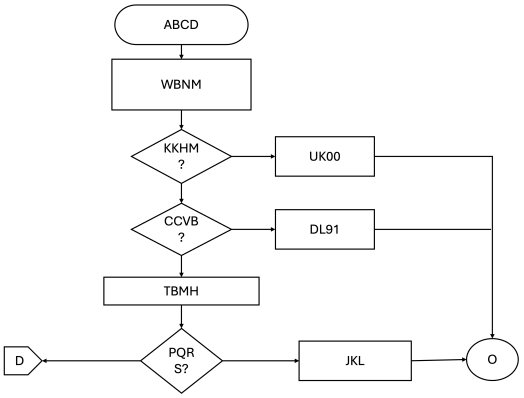
## B Ambiguous Flowchart images in 3GPP documents

A representative image from the 3GPP document that is ambiguous as a flowchart is shown in Fig. 9.

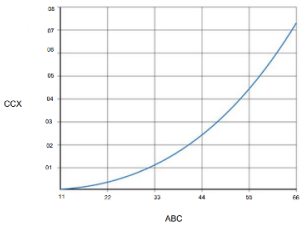
Received 30 May 2025



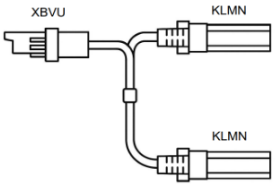

(a) Block Diagram




(b) Flowchart



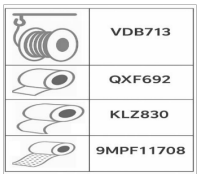

(c) Graphs



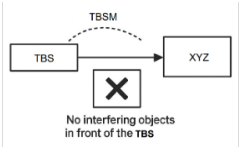

(d) Hardware Diagrams


(e) Icons

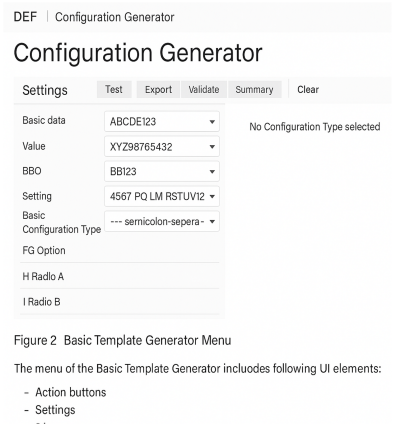
(f) Instruction Diagrams



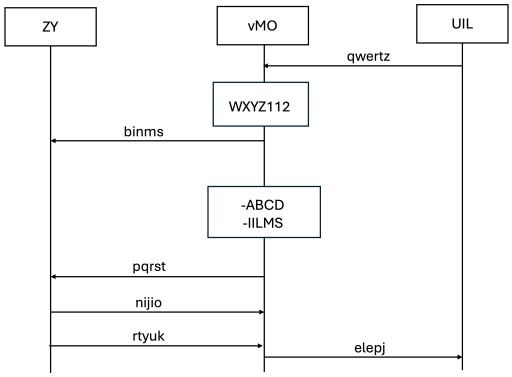

(g) Schematic Diagram




(h) Representation Diagrams




(i) Screenshots




(j) Sequence Diagrams

**Figure 8: Representative images for various categories from the proprietary telecom dataset. (Note: content has been obfuscated based on the confidentiality of data involved.)**

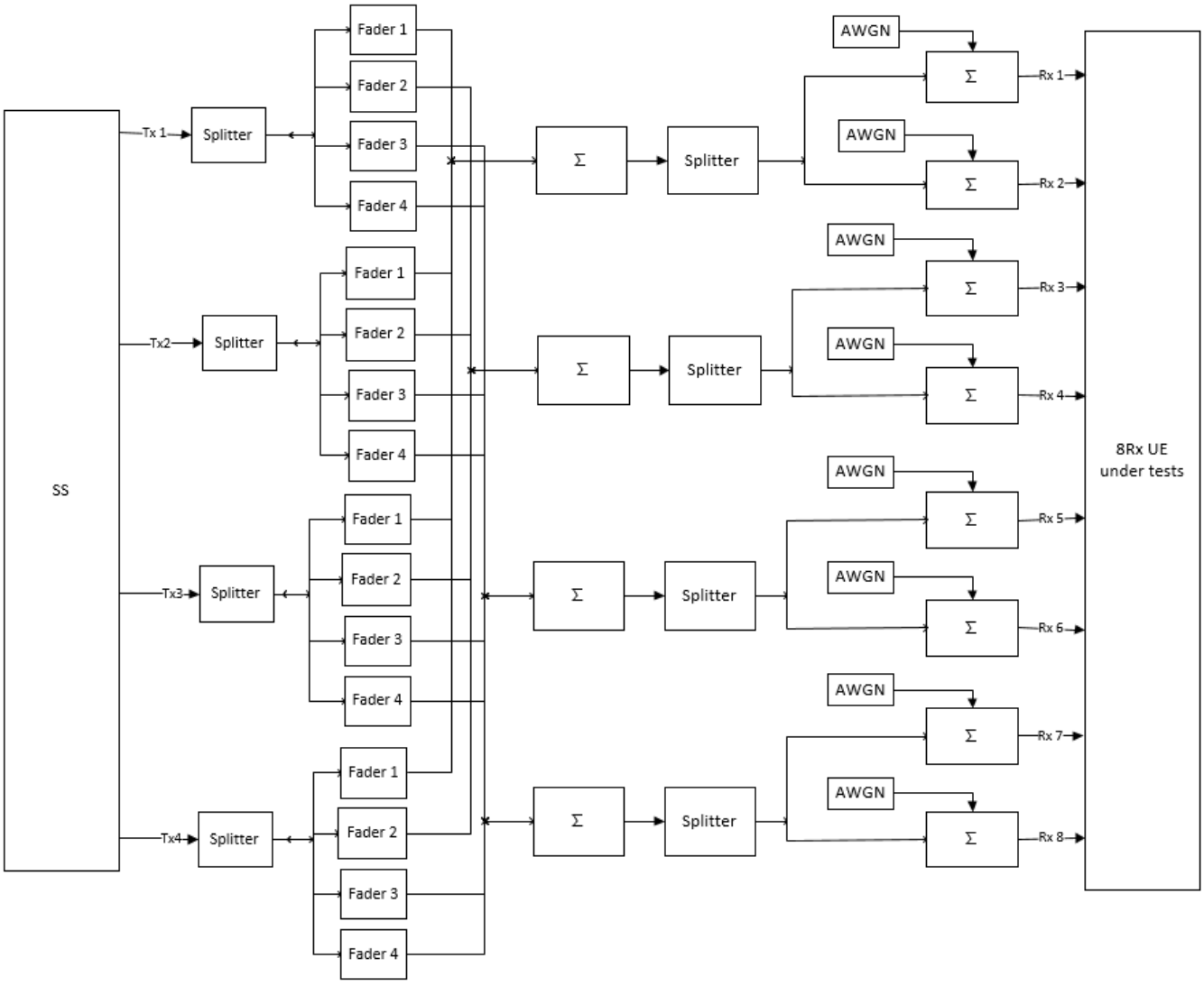



**Figure 9: A sample of ambiguous flowchart image in 3GPP document. One can consider this to also be a block diagram.**