

TACO - A Generative AI Copilot for Intent-based Telecommunication Core Network Analysis

Tamás Tóthfalusi^{*‡}, Zoltán Csiszár^{†‡}

^{*}Telecoms Division, [†]Industrial IoT Division

AITIA International Inc.

Czetz J. str. 48-50., H-1039 Budapest, Hungary

Pál Varga^{‡‡}

^{‡‡}Department of Telecommunications and Artificial Intelligence

Budapest University of Technology and Economics

Műegyetem rkp. 3., H-1111 Budapest, Hungary

Abstract—This paper presents the methodology for using LLMs to ease core network signaling analysis. This is done by applying RAG techniques to process standards that describe protocol data formats – and then asking natural language questions about actual capture traces.

Analyzing 5G networks is very challenging due to the complex and dynamic nature of signaling protocols. Unlike previous generations, protocol fields and values are described in a human-readable format, enabling textual post-processing, and the direct application of LLM models. Intent-based network management involves natural language-based human interaction with the networking equipment so the desired outcome is achieved without step-by-step instructions and settings by the human.

This paper proposes a novel approach that uses the combination of Retrieval Augmented Generation (RAG) and Langchain to automatically answer human questions regarding signalling data. This toolchain makes the analysis part of network fault management intent-based. Moreover, by training LLMs on a vast corpus of standardized signaling data, we demonstrate the model's ability to generate realistic test data. This approach improves the efficiency of automated test environments, ensuring the reliability and performance of networks in real-world conditions.

I. INTRODUCTION

Telecommunication core network analysis needs domain expert knowledge of the protocols and the connection between the various abstract terms. It also needs stamina in bit-hunting to go through vast amounts of data (even if filtered) and find the needle in the haystack. This job needs really dedicated experts who can tirelessly go after complex captures to find root causes of problems or to optimize certain services.

5G core network analysis is easier than for previous mobile core generations as the elements and values of the 5G core protocol set appear in a human-readable format. This also makes it possible to utilize them in RAG-enhanced LLMs.

The main idea of this paper is to feed 5G-related *standards* as base knowledge available to LLMs, and feed protocol *capture traces* in human-readable format. If we managed to create a suitable generative AI copilot, we could formulate questions in natural language about the *capture trace* – and expect to receive appropriate answers as we'd ask an expert. This paper describes TACO, a novel, intent-based network analysis system capable of fulfilling such intents.

RFC 9315, to define "intent-based network management", first clarifies that "intent is defined as a set of operational goals (that a network is supposed to meet) and outcomes (that a

network is supposed to deliver) defined in a declarative manner without specifying how to achieve or implement them" [1].

In this paper, we focus on fault management at the network and services levels, as defined by the TMN FCAPS model [2]. The TACO system completes 5G core network management intents, such as acting on the request: "What is the IMSI value of the subscriber whose authentication was unsuccessful?"

By training LLMs on a vast corpus of standardized signaling data, we also demonstrate the model's ability to generate realistic test data. This approach enhances the efficiency of the automated test environments, ensuring the reliability and performance of networks in real-world conditions.

The contribution of this paper is threefold, as it:

- proves that communication standards can be an efficient knowledge base for LLMs, with RAG;
- presents a working intent-based network analysis system TACO, and demonstrates its efficiency in troubleshooting through using LLM prompts in *protocol captures*,
- shows that synthetic 5G SBI protocol messages can be generated based on a purely standard-specific RAG vector store – resulting over 96% valid and relevant messages.

The paper is structured in the following way. Section 2 provides a brief overview of the related works. Section 3 describes the methodology. Section 4 describes the intent-based management system TACO. Section 5 demonstrates prompting scenarios and results. Section 6 describes the message content generation for the 5G SBI (Service Based Interfaces). Section 7 summarizes the results and the lessons learned, and Section 8 concludes with a brief summary.

II. RELATED WORKS

Retrieval-Augmented Generation (RAG) techniques offer a promising solution by integrating external knowledge into LLMs. Studies like [3] [4] [5] demonstrate that RAG can significantly enhance an LLM's ability to apply domain-specific knowledge. Moreover, combining LangChain [6] with RAG architecture [7] has shown the model's capacity to deliver high-quality, context-specific responses [8].

LLMs and RAG can be used in various areas of networking [9] [10]. Bariah, Debbah, and their co-authors rightfully argue in their IEEE ComMag article [11] that Telecom LLMs are the next big thing. A recent survey by Karapantelakis et al. [12] overviews the Generative AI scene in mobile networks.

The research and innovation interest on LLM-supported intent-based network management has been booming in the last year. The applied methods and the knowledge base used naturally differ depending on the actual objectives.

Network configuration can be done defined in intents rather than in JSON or YAML as presented in [13] and for YANG within NETCONF in [14]. GeNet is a multimodal LLM-Based copilot for network topology and configuration, helping engineers streamline network design workflows [15]. Habib et al. used LLM-based intent processing in relation to O-RAN Network Optimization with xApps and rApps [16]. Orlandi et al. used LLMs to translate customer expectations expressed as intents into slice configuration at network elements [17]. Subramaniam and Sanjay utilized LLMs for intent-based Access Control management [18].

Network traffic monitoring and analysis is essential for 5G core network management; various papers survey the methods *before* intent-based networking appeared [19], [20]. Examples on how intent extraction could work for 5G core network management is shown by [21]. A very comprehensive intent fulfillment system called Emergence [22] uses a policy-based approach [23] to model functional abstractions, and decompose intents into a hierarchy of policies. The capabilities of this system is demonstrated through a cloud intent that includes a VNF and a health check service.

At this point, there are numerous, but still relatively few examples of intent-based 5G network management that are actually *publicly reported working*. Mobile-LLaMA, as an example, has been trained with real-world 5G network datasets and its instructions have been fine-tuned to allow packet analysis, IP routing analysis, and performance analysis [24]. Another good example of intent based 5G network management is shown by [25], with demonstration examples on slice provisioning, NLoS identification and service provisioning.

Feeding protocol specifications (namely, RFCs) to LLMs have been suggested previously with different objectives. The authors of [26] aimed for state machine comprehension rather than putting protocol field data in the troubleshooting context. Looking for protocol disambiguation in RFCs was proposed in [27] by using pre-LLM NLP methods.

In our previous works, we showed complex network monitoring and data analytics with non-ML methods for VoLTE [28] and 5G NSA [29]. In this current paper we show combined results of 5G protocol analysis, where the experience is built up through ML-based translation of XML-JSON [30], our Cyber-Physical System copilot [31] and the experiments with 5G telecommunications data generation [32].

III. METHODOLOGY

A. LangChain concept

LangChain [6] is a versatile framework designed to simplify the development of applications that leverage LLMs. It provides a generic API to work with LLM-powered models, which are capable of understanding and answering human questions. The framework introduces the concept of Agents, which are autonomous entities that can interact with their

environment. Agents can be configured to perform a wide range of actions, such as searching for information, executing tools, and making decisions. These tools could be Python functions or pre-trained models as well.

The modular design allows easy customization and integration with other components, making it a highly flexible framework. It supports a wide range of use cases, from simple chatbots to complex decision-making systems. The framework is open-source and can be easily extended with new components and features. As the field of AI continues to evolve, LangChain is likely to play an increasingly important role in shaping the future of LLM applications.

B. RAG - Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is a powerful technique that combines the strengths of information retrieval and generative models [33]. It involves retrieving relevant information from a large corpus of text.

RAG converts the text information into numerical representations (vectors) by applying embedding models. Embeddings capture semantic meaning and relationships between words and phrases. The calculated vectors will be stored in a database optimized for similarity search. After searching a keyword, RAG finds the most similar vectors to a given query, allowing for efficient retrieval of relevant information.

However, this method also has some limitations. One of the main challenges is ensuring that the retrieved information is accurate. If the corpus of text contains inaccurate or outdated information as well, the generated responses may also be incorrect. Furthermore, building the vector store can be computationally expensive, especially for large corpora of text.

C. 5G Core Networks

Unlike previous generations, 5G implements telecommunication services through virtual software components in a cloud environment. 5G embraces cloud-native principles, using containers and microservices to further improve scalability, flexibility, and automation. It leverages NFV (Network Function Virtualization) extensively, allowing core components to be deployed as software on general-purpose hardware. 5G adopts SBA (Service-based Architecture) [34], which breaks down network functions into smaller, independent services that communicate using well-defined interfaces (SBI - Service-based Interface). In contrast to previous generations, the signaling of the SBI is IP/TCP/HTTP2/JSON, where the parameters are standardized by the 3GPP organization.

The JSON format is a relatively new concept in telecommunication. The previous generations preferred binary, strict formats. 5G breaks with this tradition and applies human-readable signaling between the network functions. The 3GPP organization standardized the possible key-value pairs for each URL and added an OpenAPI section to the standards as an appendix. The OpenAPI schemas often refer to other standardized sources (YAML files), e.g., schema definition files. The standardized YAML files are also freely available as

an extension of the original 3GPP standards, which is also an innovation compared to the previous generations' documents.

In this work we used the 3GPP Rel16 OpenAPI specifications [35] as knowledge base.

IV. INTENT-BASED MANAGEMENT FOR 5G INSPECTION

Figure 1 represents the architecture of the proposed model. The scope of TACO is to combine a 3GPP standards-related vector store and LangChain's reasoning engine to determine which actions to take. Since the Agent module applies an LLM call to determine the plan, the user does not have to have deep telecommunication knowledge. The algorithm is not predefined, the plan will be calculated before solving a task or answering a question. We specified some basic and some telecommunication knowledge-specific tools as well in Python and offered them to the Agent module to use. The user can ask a question regarding the 5G control plane signaling without knowing which parameters should be investigated or which transactions should be collected and filtered. To examine the concept, we collected 10 often-asked questions by operators.

A. Architecture of TACO

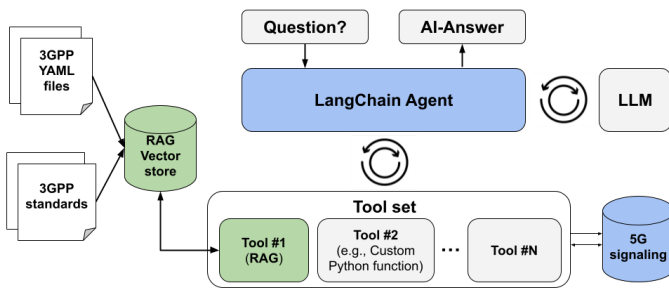


Fig. 1. The high-level architecture of TACO

The central element of the model is the LangChain Agent component, which applies two external resources: i) a Large Language Model to define and plan the AI steps that should be performed to solve a task; ii) a predefined tool set from which the Agent module chooses.

To analyze standardized signaling messages, the model should use information from the 3GPP standards [35], [36], [37]. We selected the RAG method and defined a RAG tool to embed this knowledge (see section IV. B.).

To create a tool for the Agent, a short description should be predefined with the usage method, and with the input and output parameters. To support 5G core signaling analysis, we defined five tools. The 'RAG' tool executes a relevancy search of the given phrase in the vector store of the YAML files describing the structure of the protocols and returns the n closest matches. The 'Search by IMSI' tool filters the messages by IMSI and all the messages that correlate the given IMSI. The 'Get searched messages' tool returns the first 100.000 characters of the currently filtered list of messages. It should be called multiple times to go through all the messages. The 'Filter by String' tool filters the messages, so it returns

```

ITEM NUMBER: 2245
TIMESTAMP: 2024.08.14 09:26:27.379'308'3
INTERFACE: 12
PATH: /nausf-auth/v1/ue-authentications
METHOD: POST
FLOW ID: 10128189181428096381

JSON:
{
  "servingNetworkName": "5G:mnc012.mcc216.3gppnetwork.org",
  "resynchronizationInfo": {
    "rand": "036236246EE11FD7B337687297E17941",
    "auts": "7C173FEFB0A954A6FB11F13DC859"
  },
  "supiOrSuci": "imsi-216123453086376"
}

```

Fig. 2. Example content from a preprocessed SBI message

only those that contain the given string. The 'Reset messages' tool Resets the filtered list of messages.

B. Embedding the 5G OpenAPI

Unlike previous 3GPP standards, in the case of 5G, the standardization organization defines not only a well-structured PDF for the specific interface but also standardizes the JSON keys allowed on the SBI interfaces, as well as their value sets and formats. These human-readable type- and pattern definitions are organized into separate YAML files. The initial idea of TACO was based on applying these YAML files to train a RAG model. The standardized files are freely available for the developers and researchers [35].

As a first step, we embedded the standardized YAML files for each SBI interface. However, after the first few analysis results, we realized that the standard description associated with the given interface refers to a parameter under different names in the PDF file and the YAML file, and also contains several cross-references to additional parameters. Therefore, embedding only the YAML files is not sufficient, the OpenAPI Annex part of the 3GPP PDF files was also required. This part of the standard is a formatted description of the available Resource URIs, the possible JSON keys, and their schemas.

C. Preparing the SBI signaling traffic for the LLM

Since our research group has close cooperation with one of the largest telecommunications operators in Hungary, we had the opportunity to collect real SBI traffic from a 5G core network. Before the AI-based processing steps, the 5G message contents were anonymized. The lossless traffic monitoring was based on the vendor's cloud monitoring solution, which provides defragmented and assembled HTTP2/JSON messages. The captured traffic was mapped into a human-readable text format (see Figure 2) for the AI model. The preprocessed signaling message contained the identifier of the source SBI interface, the HTTP2 Method, path, location and Status code parameters, a unique hash as a transaction identifier, and the original JSON content of the message.

V. ASKING QUESTIONS REGARDING REAL-TIME TRAFFIC

In this section, we have collected frequently asked questions by telecommunication operators that can be answered using the standard-related knowledge base and the predefined tools.



Fig. 3. AI steps to answer Q#1 - Q#3, Q#5 - Q#7

We have tried to select both simple and complex questions, where the search key can be reached through multiple cross-references. The questions could be differentiated into two groups: 1) Simple, subscriber-related questions, 2) Complex questions requiring telecommunication-specific knowledge. We asked these predefined questions from the model multiple times and examined the results and the performed AI steps.

The AI model was allowed to use all tools defined in section IV. A., including the 3GPP standards-related RAG tool, and the filtering and searching tools.

A. Simple questions regarding the 5G signaling

Question #1: Which MSISDN belongs to the "12330404305555" IMSI value?

AI Answer: The MSISDN associated with the IMSI value "12330404305555" is "36309700555".

Results: The answer is correct, Figure 3 presents the applied AI steps.

Question #2: What is the subscriber's cell identifier whose IMSI is "12330404305555"?

AI Answer: The subscriber with IMSI "12330404305555" has a cell identifier (nrCellId) of "015590033".

Results: The answer is correct, Figure 3 presents the applied AI steps.

Question #3: What is the Tracking Area ID (TAI) value in mcc/mnc/tac format for "12330404305555" IMSI?

AI Answer: The Tracking Area ID (TAI) value in 'mcc/mnc/tac' format for IMSI "12330404305555" is '216/30/0007E3'.

Results: The answer is correct, Figure 3 presents the applied AI steps.

Question #4: List the cell informations from the message sequence in mcc/mnc/nrCellId format.

AI Answer: Here are the extracted cell information in the 'mcc/mnc/nrCellId' format from the provided messages:
 1. '216/30/015590033'
 2. '216/30/01558C033'

Results: The answers are correct, the AI model applied only the "MCC" keyword as an input parameter of the "Filter by String" tool. In some cases, the results contained the same values multiple times, but the expected values were always included in the result list.

B. Complex questions that require telecommunication skills

Question #5: Give the PDU Session identifier of the user with IMSI of "12330404305555" in [IMSI, PDU Session ID] format.

AI Answer: The PDU Session identifiers associated with the user having the IMSI "12330404305555" are as follows:

- IMSI: 12330404305555, PDU Session ID: 1 - IMSI: 12330404305555, PDU Session ID: 2

Results: The answer was correct, but the format didn't match with the expected. Figure 3 presents the applied AI steps.

Question #6: Give the PDU Session identifier of the user with IMSI of "12330404305555" in [IMSI, PDU Session ID, DNN] format.

AI Answer: ["12330404305555", "2", "IMS.mnc030.mcc216.gprs"]

Results: The answer was correct. Figure 3 presents the applied AI steps.

Question #7: Was the 5G authentication of IMSI "123304043044444" successful?

AI Answer: The subscriber's 5G authentication was not successful. The message indicates an AUTHENTICATION_FAILURE for the subscriber with IMSI '123304043044444'.

Results: The answer was correct. Figure 3 presents the applied AI steps.

Question #8: List the unique slice IDs.

AI Answer (Failure): It seems there are no messages containing the exact string "slice ID." Please provide a different keyword or specify more details if needed.

AI Answer (Success): The unique slice IDs found in the messages are as follows:
 1. 'sst: 1, sd: 000001'

Results: The model tried to use the RAG tool, which is a correct way to find the slice identifier information from the JSON values. However, when the model applied the "slice id" keyword as a searching parameter, it could not find the required JSON keys. In contrast, when the model applied the "pattern" keyword in the RAG searching step, it could use the results ("SNSSAI" keyword) for the further steps. Figure 4 presents the applied AI steps.

Question #9: What is the IMSI value of the subscriber whose authentication was unsuccessful?

AI Answer (Failure): It appears that there are no messages indicating an "authentication unsuccessful" event.

Question #9 (modified): What is the IMSI value of the subscriber whose authentication was unsuccessful? Use RAG first.

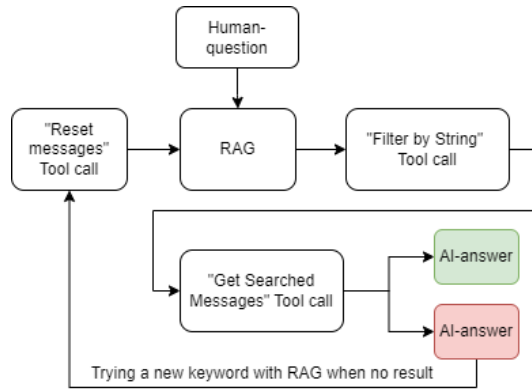


Fig. 4. AI steps to answer successfully Q#8 - Q#10

AI Answer (Success): The IMSI value of the subscriber whose authentication was unsuccessful is 'imsi-123304043044444'.

Results: The model used synonyms for "unsuccessful authentication" to filter messages and find related content for Question #9. However, it couldn't find the right messages. To force the use of the RAG tool as a first step, we modified the prompt – hence the model iteratively re-tried the same steps with the keywords the RAG tool returned. Finally, it solved the task after 3 iterations. Fig. 4 shows the applied AI steps.

Question #10: Was there any error associated with the subscriber whose IMSI is '12330404303333'?

AI Answer (Success): The subscriber with IMSI '12330404303333' encountered an error on '2024.08.14 09:33:26.704'160'7'. The status code for the flow ID '11140904671710795637' was '415', indicating "Unsupported Media Type".

Results: The answer was correct, the AI model correctly used the "Search by IMSI" tool for the initial steps. It identified the error event based on the Status code after examining the filtered transactions. We also tried cases with multiple errors during a time interval. The result was more than expected, the model effectively summarized the error events and highlighted the most common error.

VI. SBI MESSAGE CONTENT GENERATION

By utilizing the knowledge contained within the standards, we can also use the model to generate the content of SBI signaling. For the generation process, the model must identify multiple correlations across various documents. The JSON key names found in the standards [35] may appear under different names in the schema definition files. Additional challenges include generation based on regex parameters and maintaining the correct value set. We have examined the fulfillment of these conditions through Section VI. A., and VI. B.

A. Generating content for a predefined JSON key

When generating values for JSON keys, we can distinguish between two cases: simple and complex objects. In our ex-

```

"ueLocation": {
  "nrLocation": {
    "ncgi": {
      "plmnId": {
        "mnc": "12",
        "mcc": "216"
      },
      "nrCellId": "015590033",
      "tai": {
        "plmnId": {
          "mnc": "12",
          "mcc": "216"
        },
        "tac": "0007E3"
      }
    }
  }
},
"nrLocation": {
  "tai": {
    "plmnId": {
      "mcc": "310",
      "mnc": "260"
    },
    "tac": "0001"
  },
  "ncgi": {
    "plmnId": {
      "mcc": "310",
      "mnc": "260"
    },
    "nrCellId": "00000001"
  },
  "ageOfLocationInformation": 120,
  "ueLocationTimestamp": "2023-10-05T14:48:00Z",
  "geographicalInformation": "1234567890ABCDEF",
  "geodeticInformation": "12345678901234567890",
}

```

Fig. 5. Real-life example of the "ueLocation" object (left), and part of an AI-generated JSON object (right)

ample, we examined the "ueLocation" (see Figure 5). During the generation process, we analyzed the results for different embedding levels, and applied the following prompt template:

Generate an example JSON content for the '...' JSON key which is part of a 5G SBI message. Also consider the possible value ranges. Print only the JSON object without comments.

To examine the embedding levels, we disassembled the "ueLocation" object. Figure 5 presents an example content from real-life traffic, and a snapshot from an AI generated content. If we compare the objects the difference is that the AI model generates more JSON key-value pairs. Since the standard defines these optional parameters as well, the result could be a possible content of an SBI message.

Upon reaching the topmost level, we attempted to access the 'ueLocation' JSON key, but the generated results did not contain any valid values. Instead, it contained the pattern definition. This result indicates that we have reached the model's limitation in handling embedding objects.

B. Generating Message Content based on Resource URI

Automated test systems or traffic generation systems need complete message contents instead of generating a specific value for a JSON key. Since the documents embedded in the RAG tool contain all the information for creating a complete SBI message, we examined the effectiveness of the proposed model. We applied the following prompt template:

Generate an example JSON content for a 5G SBI message. The resource URI of the message is '...'.

We tried the following Resource URI values:

- 1) '/nausf-auth/v1/ue-authentications'
- 2) '/nsmf-pdusession/v1/sm-contexts'

By examining the performed AI steps, the Agent module always called the RAG tool as a first step.

The request message to access the '/nausf-auth/v1/ue-authentications' resource has two mandatory JSON key-value pairs: "supiOrSuci" and "servingNetworkName", and several other optional parameters as well. The generated output contained the required mandatory parameters, but the LLM added extra key-value pairs, which were not related to the asked URI. These parameters were also retrieved from the RAG tool.

To support a more effective RAG search, instead of asking for the Resource URI, we tried the data structure reference supported by the 5G function. The '/nausf-auth/v1/ue-authentications' URI is referenced by the 'AuthenticationInfo' data structure. We modified the prompt as follows:

Generate an example JSON content for a 5G SBI message. The structure definition is: 'AuthenticationInfo'. Generate a value for each JSON key based on the pattern.

After this modification, the model generated valid SBI message contents. However, the question had to be modified based on standard-related knowledge, the Resource URI - structure definitions could be embedded into the RAG database.

By examining the '/nsmf-pdusession/v1/sm-contexts' URI, which could contain even 34 JSON objects, we experienced the same. We had to change the URI to the 'SmContextCreateData' structure definition reference to select and generate valid JSON objects. However, some of the mandatory parameters were missing from the content. We tried directed questions to also add the mandatory objects as well, but the model could not decide between the optional and mandatory parameters.

VII. RESULTS AND LESSONS LEARNED

The TACO model was built and examined to analyze real-life 5G core network traffic and automatically answer telecommunication standard-related questions. To create a 3GPP standard-specific RAG vector store, SBI schema definition files were embedded. Initial results showed the need to include the OpenAPI Annex of 5G function-specific standards [35] to handle cross-references, data structure, and JSON definitions effectively. To support the LangChain Agent module, prompts should guide using the RAG tool first to identify proper 3GPP JSON keys or schema definitions for filtering tasks.

The proposed model demonstrated effectiveness in searching subscriber-related parameters (e.g., IMSI-MSISDN pairs, cell data) and event-related queries. It could identify error causes and efficiently summarize the main events. Processing times include the RAG search and LLM API call, taking under 5 seconds for analysis use cases. However, Python functions used as tools may take several minutes. For generation tasks, creating standard-related JSON objects typically takes 4-6 seconds, including RAG and LLM processing.

The model's primary limitation is the size of the LLM's context window. Filtering messages by key values allowed processing around 1000 SBI messages (120 tokens per JSON message) per context window using GPT-4o. To handle longer sequences, JSON message inputs should be split into a limited number of messages and processed iteratively.

By examining the generation results (see Figure 6), the model achieved 98% accuracy in generating relevant JSON keys for Resource URI-related cases. It handled 2 or fewer JSON object embedding levels in 96% of cases and 3 levels in 79%, but lost context beyond 3 levels. For generating JSON values, the model adhered to pattern definitions over 90% of the time when provided with keys but dropped to 76–88% when inferring both keys and values. An effective solution for

Use case	Number of generated JSON key-value pairs	Rate of standard-related, valid JSON keys	Rate of standard-related, valid JSON values
AuthenticationInfo	1347	98%	76,0%
SmContextCreateData	865	100%	88,4%
ncgi	474	96,6%	90,8%
nrLocation	1558	79,1%	95,8%

Fig. 6. Generation results based on 100 independent running outcomes

obtaining more accurate values is to retrieve the regex pattern and force its use in multiple steps [32].

Directly applying Resource URIs in prompts caused issues. Replacing URIs with structure reference names allowed the model to generate complete message content but failed to distinguish mandatory from optional parameters. Embedding Resource URI and data structure pairs with mandatory-optional indicators into the RAG vector store could address this limitation.

VIII. SUMMARY

In this paper, we proposed a robust, LLM-based approach for analyzing 5G core network signaling, and generating standard-compliant content. The presented model combines the RAG method and the LangChain Agent reasoning engine to work with 3GPP standards.

We examined two use cases: real-life signaling analysis, and standardized JSON content generation. Regarding the signaling analysis, we examined 10 relevant questions to retrieve information from real-life traffic (e.g., subscriber IDs, cell information, error events). After the first results, we identified which part of the standard should be embedded in the RAG vector store, and also proposed prompt templates. We present the planned AI-steps and the applied tools to answer the human-questions. We consider this to be a significant contribution to intent-based network traffic analysis.

The model was also effective in the generation of standardized content. It could produce relevant JSON contents in 98% of the cases, but this drops to 96% and even further if we increase the embedding levels. If we examine the accuracy of the pattern-related JSON values, it could generate key-related values in more than 90% of the cases. When the model had to work with structure definitions, which could contain more dozen parameters, the effectiveness was reduced to 76%. As shown by the results, the model can process standardized data and make sense of the schemas defined in the standard.

In future work, we will focus on the limitations and fine-tune the embedded information.

ACKNOWLEDGMENT

This research has been funded by the European Commission, through the Horizon Europe programme, Chips Joint Undertaking, and National Funding Authorities from 10 involved countries under the research project Arrowhead fPVN with Grant Agreement no. 101111977.

REFERENCES

- [1] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, "Intent-based networking-concepts and definitions," IETF RFC 9315, Oct, Tech. Rep., 2022.
- [2] International Telecommunication Union - ITU-T, *M.3400 TMN management functions*, ITU-T Std. M.3400, 2000.
- [3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *arxiv:2005.11401*.
- [4] Y. H. Ke, L. Jin, K. Elangovan, H. R. Abdullah, N. Liu, A. T. H. Sia, C. R. Soh, J. Y. M. Tung, J. C. L. Ong, and D. S. W. Ting, "Development and testing of retrieval augmented generation in large language models - a case study report," in *arxiv:2402.01733*.
- [5] R. Al-Shabandar, A. Jaddoad, T. A. Elwi, A. Mohammed, and A. J. Hussain, "A systematic review for the implication of generative ai in higher education," *Infocommunications Journal*, vol. 16, no. 3, pp. 31–42, 2024.
- [6] *LangChain framework*, Std. [Online]. Available: <https://www.langchain.com/>
- [7] OpenAI, "Embeddings," OpenAI Docs, Tech. Rep., 2024. [Online]. Available: <https://platform.openai.com/docs/guides/embeddings>
- [8] O. Cárdenas, S. Falconi, E. Tusa, and A. Rodríguez, "Development of a chatbot model for health telecare: Integration of langchain, embeddings with openai, and pinecone using the question answering technique," in *Journal of Applied Research and Technology*.
- [9] Y. Huang, H. Du, X. Zhang, D. Niyato, J. Kang, Z. Xiong, S. Wang, and T. Huang, "Large language models for networking: Applications, enabling techniques, and challenges," *IEEE Network*, pp. 1–1, 2024.
- [10] K. Choudhury, A. Kumar, A. Das, V. Novelli, A. K. Roy, A. Vamos, and J. Hajdu, "How to make better use of network insights with generative ai," Ericsson, Tech. Rep., 2024. [Online]. Available: <https://www.ericsson.com/en/blog/2024/2/how-to-make-better-use-of-network-insights-with-generative-ai>
- [11] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, and M. Debbah, "Large generative ai models for telecom: The next big thing?" *IEEE Communications Magazine*, 2024.
- [12] A. Karapantelakis, P. Alizadeh, A. Alabassi, K. Dey, and A. Nikou, "Generative ai in mobile networks: a survey," *Annals of Telecommunications*, vol. 79, no. 1, pp. 15–33, 2024.
- [13] A. Mekrache, A. Ksentini, and C. Verikoukis, "Intent-based management of next-generation networks: An llm-centric approach," *Ieee Network*, 2024.
- [14] G. Hollosi, D. Ficzer, and P. Varga, "Generative ai for low-level netconf configuration in network management based on yang models," in *20th International Conference on Network and Service Management (CNSM)*. IEEE/IFIP, 2024.
- [15] B. Ifland, E. Duani, R. Krief, M. Ohana, A. Zilberman, A. Murillo, O. Manor, O. Lavi, H. Kenji, A. Shabtai *et al.*, "Genet: A multi-modal llm-based co-pilot for network topology and configuration," *arXiv preprint arXiv:2407.08249*, 2024.
- [16] M. A. Habib, P. E. I. Rivera, Y. Ozcan, M. Elsayed, M. Bavand, R. Gaigalas, and M. Erol-Kantarci, "Llm-based intent processing and network optimization using attention-based hierarchical reinforcement learning," *arXiv preprint arXiv:2406.06059*, 2024.
- [17] B. Orlandi, S. Lataste, S. Kerboeuf, M. Bouillon, X. Huang, F. Fauchaux, A. Shahbazi, and P. Delvallet, "Intent-based network management with user-friendly interfaces and natural language processing," in *2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN)*. IEEE, 2024, pp. 163–170.
- [18] P. Subramaniam and S. Krishnan, "Intent-based access control: Using llms to intelligently manage access control," *arXiv preprint arXiv:2402.07332*, 2024.
- [19] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (ntma): A survey," *Computer Communications*, vol. 170, pp. 19–41, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366421000426>
- [20] A. El Hanjri, I. Ben Abdel Ouahab, and A. Haqiq, "Survey on handover techniques for heterogeneous mobile networks," *Infocommunications Journal*, vol. 16, no. 3, pp. 64–74, 2024.
- [21] D. M. Manias, A. Chouman, and A. Shami, "Towards intent-based network management: Large language models for intent extraction in 5g core networks," in *2024 20th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2024, pp. 1–6.
- [22] K. Dzevaroska, A. Tizghadam, and A. Leon-Garcia, "Emergence: An intent fulfillment system," *IEEE Communications Magazine*, vol. 62, no. 6, pp. 36–41, 2024.
- [23] K. Dzevaroska, J. Lin, A. Tizghadam, and A. Leon-Garcia, "Llm-based policy generation for intent-based management of applications," in *2023 19th International Conference on Network and Service Management (CNSM)*. IEEE, 2023, pp. 1–7.
- [24] K. B. Kan, H. Mun, G. Cao, and Y. Lee, "Mobile-llama: Instruction fine-tuning open-source llm for network analysis in 5g networks," *IEEE Network*, vol. 38, no. 5, pp. 76–83, 2024.
- [25] J. Mcnamara, D. Camps-Mur, M. Goodarzi, H. Frank, L. Chinchilla-Romero, F. Cañellas, A. Fernández-Fernández, and S. Yan, "Nlp powered intent based network management for private 5g networks," *IEEE Access*, vol. 11, pp. 36 642–36 657, 2023.
- [26] P. Sharma and V. Yegneswaran, "Prosper: Extracting protocol specifications using large language models," 2023.
- [27] J. Yen, T. Lévai, Q. Ye, X. Ren, R. Govindan, and B. Raghavan, "Semi-automated protocol disambiguation and code generation," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 272–286. [Online]. Available: <https://doi.org/10.1145/3452296.3472910>
- [28] P. Varga, T. Tothfalusi, Z. Balog, and G. Sey, "Complex solution for volte monitoring and cross-protocol data analysis," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–4.
- [29] G. Soós, D. Ficzer, P. Varga, and Z. Szalay, "Practical 5g kpi measurement results on a non-standalone architecture," in *IEEE/IFIP network operations and management symposium (NOMS)*. IEEE, 2020, pp. 1–5.
- [30] T. Tothfalusi, E. Varga, Z. Csiszar, and P. Varga, "MI-based translation methods for protocols and data formats," in *2023 19th International Conference on Network and Service Management (CNSM)*. IEEE, 2023, pp. 1–5.
- [31] C. Hegedűs and P. Varga, "Co-pilots for arrowhead-based cyber-physical system of systems engineering," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. IEEE, 2024, pp. 1–6.
- [32] T. Tóthfalusi, Z. Csiszár, and P. Varga, "Utilizing generative ai for test data generation-use-cases for iot and 5g core signaling," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. IEEE, 2024, pp. 1–6.
- [33] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [34] *3GPP TS 23.501, System architecture for the 5G System (5GS)*, Std.
- [35] 3GPP, "Rel-16 OpenAPI," 5G, Tech. Rep., 2020. [Online]. Available: <https://www.3gpp.org/ftp/Specs/Archive/OpenAPI/Rel-16/>
- [36] *3GPP TS 29.571, 5G System; Common Data Types for Service Based Interfaces*, Std.
- [37] *3GPP TS 29.509, 5G System; Authentication Server Services*, Std.