Poster Abstract: Text2Net: Transforming Plain Text into Dynamic, Interactive Network Simulations

Alireza Marefat
amarefatvayghani1@student.gsu.edu
Georgia State Univerisity
Atlanta, Georgia, USA

Abbaas Alif Mohamed Nishar
amohamednishar1@student.gsu.edu
Georgia State Univerisity
Atlanta, Georgia, USA

Ashwin Ashok
aashok@gsu.edu
Georgia State Univerisity
Atlanta, Georgia, USA

ABSTRACT

This paper introduces Text2Net, an innovative system designed to transform plain English descriptions into dynamic, interactive network simulations within the Emulator Virtual Engine–Next Generation (Eve-NG) environment. By integrating SOTA technologies from Natural Language Processing (NLP), Large Language Models (LLMs), and proposed adaptor software, Text2Net bridges the technical knowledge gap, enabling both technical and non-technical users to effortlessly create and interact with complex network topologies within a simulation environment. The system architecture combines an intuitive chat interface, GPT4 LLM to interpret user inputs, NLP key-value extraction, a simulation adaptor, and the EVE-NG engine. TextNet democratizes network emulation, empowering educators to efficiently construct simulations for interactive learning. It also benefits industrial prototyping and testing configurations.

CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; **Computer-assisted instruction**; **IT architectures**; • **Networks** → **Network design principles**; **Programming interfaces**; **Topology analysis and generation**; **Logical / virtual topologies**; **Network manageability**; **Programmable networks**; **Network management**; **Network monitoring**.

KEYWORDS

Network Simulation, Large Language Models, Natural Language Processing, Emulator Virtual Engine–Next Generation (Eve-NG), Domain-specific models, Dynamic Network Automation

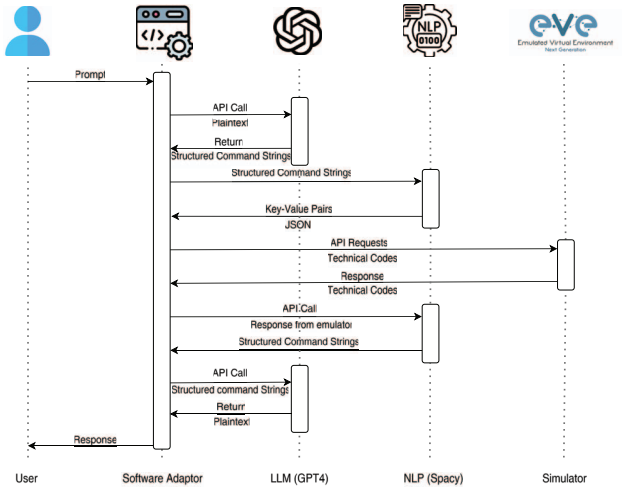

Figure 1: Proposed Text2Net System Design Model

1 INTRODUCTION

The rapid evolution of computer networks demands innovative approaches for emulation tools that bridge the technical knowledge gap and enhance accessibility for a diverse user base. This project introduces a groundbreaking solution that integrates advancements in Natural Language Processing (NLP), Large Language Models (LLMs), and simulation environments **to revolutionize the way computer networks can be designed and/or understood via simulations**.

The core objective of this research is to mitigate the challenges posed by existing Large Language Models' limitations in accurately interpreting non-technical language and generating precise technical outputs. Our approach involves the development of a domain-specific model tailored for computer networks that facilitates the intuitive interaction between users and computer networking conceptual simulations.

Targeting both academia and industry, this system is designed to serve educators, students, network engineers, and non-technical individuals alike. In academia, it offers a practical, hands-on approach to teaching complex network concepts, allowing for the seamless integration of theoretical knowledge and practical application. Industry professionals, on the other hand, can leverage this system for efficient design and testing of network architectures in both simulation and real production environments, while non-technical users gain the ability to interact with network setups without requiring specialized knowledge. The technical architecture of this project involves a multi-stage process that will be discussed in the System Model section. By introducing an innovative integration of LLMs for intuitive interaction and the use of NLP and simulation automation, this project stands at the forefront of technological advancement in computer network emulation.

2 SYSTEM MODEL

This section outlines our system's architecture, depicted in Fig 1, consisting of modules like the interactive Software Adaptor, LLM, NLP-based Key-value Pair Extractor, and Simulation Environment, with further details provided in subsequent sections.

2.1 Software Adaptor

The adaptor is a Python-based software that communicates with different modules in the system model such as the user, LLM, SpaCy,

979-8-3503-6201-5/24/$31.00 ©2024 IEEE
DOI 10.1109/IPSN61024.2024.00058

and the EVE-NG to deliver the overall task. Its operation is segmented into several key functionalities, and the overall flow is segregated and explained below:

2.2 User to Adaptor:

The user interaction with our system begins via an intuitive chat panel, accommodating inputs from network engineers, and university lecturers, to non-technical individuals in plain English. This interface adeptly handles a spectrum of inputs, from simple inquiries like network status to complex network topology descriptions.

2.3 Adaptor to LLM (GPT4):

The system utilizes one of the current most powerful LLM models, GPT4 OpenAI. The adaptor sends the plaintext from the previous step to GPT4 to get the "Structured Command Strings", using GPT4 API calls. Structured Command Strings are short sentences derived from the plaintext which contains one key-value pair. The transition from plaintext to structured command strings is crucial, as existing LLMs lack the precision to directly interpret complex network configurations into a comprehensive `JSON` format for simulation provisioning [2]. We aim to retrain GPT4 for future research to directly output the desired JSON file, eliminating the NLP module and currently focus on the system's other components.

2.4 Adaptor to NLP (SpaCy):

Following LLM processing, the system utilizes NLP techniques, specifically leveraging SpaCy, to extract key-value pairs from the structured command strings. This extraction is pivotal for identifying specific network components, such as devices, connections, and configurations. The result of this process is a structured JSON output, which encapsulates all the necessary information for emulating the described network topology.

2.5 Adaptor to Simulation Environment:

The proposed model is able to be integrated with any real production environment, but since we did not have any access to a live production we opted for a simulation environment. EVE-NG is chosen for its superior ability to use real device IOS images and broad emulation capabilities, surpassing other simulators like Cisco Packet Tracer and GNS-3 in creating realistic network simulations [1]. Its support for a wide array of network devices and configurations enables the development of complex, interactive simulations, making it essential for accurate network emulation. The interaction between the adaptor and simulation involves the below subtasks:

EVE-NG Login and Session Management: Utilizes the `'requests'` library to authenticate against EVE-NG's API, establishing a session by capturing and maintaining cookies for subsequent API calls.

Device and configuration Provisioning: Device provisioning dynamically generates network devices, such as routers and switches, within a simulation by sending API requests, with each device's type, image, and configuration specified in a `JSON` payload, also incorporating randomized node placement on the EVE-NG canvas. Concurrently, `JSON` input parsing processes a file detailing network architecture, including device definitions, connections, and configurations ranging from basic settings to advanced networking concepts like "Routing" (with 'NAT', 'Static Routing', 'OSPF', 'EIGRP', and 'RIP' for Layer 3) and "Network Access" (covering 'VLAN', 'TRUNK', and 'EtherChannel' for Layer 2), each with parameters in the `JSON` file managed through separate functions. We leverage Function Calls to match the desired functions and parameters.

Graphical User Interface (GUI) Automation: utilizes `'selenium'` for GUI automation to log into EVE-NG through a web browser, navigate to the appropriate lab, and execute actions like opening device consoles. Uses `'pyautogui'` for further GUI interactions, such as clicking on specific screen coordinates, to facilitate operations not directly manageable via EVE-NG's API or where direct API interaction is less practical.

Link Creation and Interface Configuration: Automates the process of defining links and associating them with device interfaces through EVE-NG's API. This includes creating networks and configuring the interfaces of source and destination devices.

Telnet Management: Prepares for telnet connections to devices by extracting and mapping device telnet URLs from EVE-NG's API response, for device configuration beyond initial setup.

3 ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation(1901133) and was supported in part by UDSA-NIFA(2021-67019-34337)

4 CONCLUSION

This project stands at the forefront of technological advancement in computer network dynamic automation and provisioning, addressing static automation tools limitations by opening new avenues for accessible, user-friendly technology engagement. Through the integration of LLMs and NLP for intuitive interaction and the use of our proposed adaptor software and EVE-NG simulation, we offer a novel solution that promises to transform the landscape of network education and design. In future work, the emphasis would be on fine-tuning and retraining the GPT4 to enhance its understanding of complex network scenarios and further develop the domain-specific model.

REFERENCES

[1] P Segeč, M Moravčík, M Kontšek, J Papán, J Uramová, and O Yeremenko. 2019. Network virtualization tools–analysis and application in higher education. In *2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 699–708.

[2] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems* 32 (2019).