NLP for Smart Contracts

MSc Research Project
MSc Fintech

Nicholas Apati
Student ID: 17332263

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Nicholas Apati………………………………………………………………

Student ID: ………17332263………………………………………………………………………………………

Programme: …Fintech…………………………………………………………… **Year:** ………1…………………..

Module: ……………Practicum…………………………………………………………………………………

Supervisor: …Victor Del Rosal…………………………………………………………………..………

Submission Due Date: ……………12/8/24……………………………………………………………………………………. ………

Project Title: ……NLP for Smart Contracts…………………………………………………………………..………

Word Count: ………5076…………… **Page Count**……………15…………………………………..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………Nicholas Apati………………………………………………………………………

Date: …12/8/24……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

NLP for Smart Contracts

Nicholas Apati
17332263

Abstract

This research presents a way to determine how well an NLP algorithm can translate smart contracts that are in Solidity programming language to natural language that people can comprehend and vice versa. The approach is centered on enabling the improvements of smart contracts' interpretability and scalability by applying NLP in the cryptocurrency context. It is possible to conclude that the goal of the discussed study is to introduce NLP algorithms into decision-making processes in smart contracts in a way that would be transparent and open to users. In the method, the literature research, the NLP algorithm for smart contract analysis and generation, and experiments to estimate the efficiency of the algorithms are described. A front-end HTML interface has been created; an API key has been created through OpenAI for translating Solidity code into English and back. Some experiments prove that the interface can take the translations, but to resolve present issues, optimisation is required. The expected outcomes include comprehending how NLP will be employed to translate smart contracts and how these contracts can be developed and enforced in the Case of Cryptocurrencies.

1 Introduction

Cryptocurrencies are more observed today than before because they introduce a new approach to reshaping existing financial models, introducing a transparent and decentralised system of transactions. However, there is always a disadvantage to every cryptocurrency like Ethereum, and one of them is scalability. Therefore, as the number of transactions increases, they must be effectively managed. Thus, although conventional blockchain technologies guarantee the security and decentralisation of transactions, they fail to satisfy the growing need for performing transactions, which, in turn, leads to delays.

In terms of scope, this research explores how to make writing smart contracts in the sphere of cryptocurrency less complicated. Smart contracts may be defined as operationalised agreements between parties whereby the necessary provisions are multiplexed into the system. They are useful in executing transactions within cryptocurrencies. Smart contract code is highly technical, and this makes it difficult for individuals who are not computer literate to understand, and this hinders their adoption and development (Yang et al., 2024).

In response to this challenge, the research seeks to design and apply NLP algorithms to translate smart contracts for humans. This research contributes to the growing literature on blockchain by enhancing understanding of these algorithms' capacity to scan and produce smart contracts.

The main research question guiding this study is: The justification for the research stems from the following research question: How adequately can an NLP algorithm translate smart contracts to natural language? The hypothesis is the NLP algorithms can translate the smart contract code into human understandable language so the non-technical user can easily comprehend it. The hypothesis is that the use of NLP algorithms helps to translate smart contract code for the use of the common people since they cannot understand such laws, as is the case with programmers.

To accomplish this, a front-end HTML interface has been designed, which specialises in an API key of OpenAI. This interface enables the user to translate between Solidity code and English language and the other way around. The initial implementation shows that although it is possible to process translations with its help, some work must be done to decrease existing problems and increase production.

Both technical problems, such as the specificities of smart contract programming, and usability issues are described in the literature review. It reveals a research gap in the use of NLP for smart contract translation. This study seeks to narrow this knowledge gap by coming up with a practical guide on how smart contracts based on NLP can be integrated into cryptocurrency.

The expected results of this research include but are not limited to gaining improved knowledge and awareness on how to apply NLP to understand and translate smart contracts into natural languages. The project is to prove that the introduction of NLP algorithms can add a substantial amount of enhancement to the legibility of smart contracts and help to make the concept of blockchain more understandable for a normal user. In converting these codes into human interfaces, the users can understand the basics of blockchain technology, find solutions to newly arising issues and encourage the use of this promising technology.

2 Related Work

This literature review seeks to critically assess various studies in the literature by considering the application of Natural Language Processing (NLP) algorithms in translating smart contracts into natural language. This is done in a way that categorises previous work under the headings of comparison and contrast, as well as the relation and enumeration of objectives of the review.

2.1 Scalability of Blockchain Systems

Smart contracts contend with a major hindrance of scalability that affects the performance and KPIs of blockchain technology. Shirodkar et al. (2022) also stress Layer 1 and Layer 2 solutions enhance the average transaction through rates, which contribute to smart contract execution. Yang et al. (2024) extend this by discussing NLP-based approaches that can make the application more scalable mainly because of the optimised smart contracts. Such improvements are important for overcoming the shortcomings mentioned by Shirodkar et al. and relating to the overall scalability approaches (Hafid et al., 2020; Zhou et al., 2020).

The latest works include studies by Kanwhen et al. (2023) and Neiheiser et al. (2023) that describe the technique of sharding and Layer-2 solutions in combating bad scaling. These studies help in understanding how specific technical elements can serve as input for improving blockchain functions that affect the application of smart contracts.

2.2 NLP for the Interpretation of Smart Contracts

Modern work has paid much attention to how NLP can enhance the explainability of smart contracts for individuals who are not knowledgeable in programming languages. Aejas, Belhi, and Bouras (2023) provide extant literature on the process of converting conventional contracts into smart contracts through NLP, which can also help in closing the gap between conventional and blockchain platforms. Further, alongside this work, Zhang et al. (2022) present the approach to distilling an NLP model to help interpret smart contracts and make them more readable.

Using NLP, Monteiro et al. (2020) expound on the potentiality of creating tended smart contract forms working exclusively for the legal and accounting domains. It is evident from their approach how NLP reduces the time and effort required to create them, therefore improving contract readability. Related to this discussion, Smith et al. (2021) continue this by emphasising the potential use of NLP algorithms when implemented in Cardano to enhance the interpretability of pre-existing smart contracts, hence aligning with the call for better interpretability.

2.3 The natural language explanations are in general generated and analysed.

Regarding the extant literature, the generation of natural language explanations for smart contracts is critical in enhancing their understandability. Wang et al. (2022) are mainly aimed at generating natural language descriptions to add cove that are useful for making the smart Contract more understandable. This work relates to Zhang et al. (2021), which provides an overview of numerous NLP approaches for translating smart contract code into more understandable versions.

Fan et al. (2023) deal with technical scalability in the context of the storage extension for sharding blockchain systems, which is complementary to the goal of making the interpretability of smart contracts better since it provides a scalable environment. Thus, this technical perspective is aligned with the user-centric view that Wang and Zhang present; it offers the backbone needed to improve the understanding of contracts.

2.4 Implication for Practice and Theory Foundations

As has been revealed above, the practical implementations of NLP in smart contract translation are generally applicable to a wide range of fields. Several works directly relate to the application of NLP in the context of improving contract development by translating smart contracts into natural language and improving the developer interfaces with neural networks, as described by Nelaturu et al. (2023). Prasad et al. (2023) put forward the NLP in the appraisal of its employee performance, proving the universal applicability of NLP and its impact on various fields such as that of blockchain (Prasad et al., 2023).

Also, Jayakumar et al. (2023) suggested a secure event management system through the application of NLP and RNN algorithms; NLP's utilisation for smart contract securing and managing has been discussed in detail by Jayakumar et al. (2023). To complement the discussion, Neiheiser et al. (2023) limit of Ethereum's Layer-2 solutions shed more light on the scalability as well as the feasibility of smart contracts (Neiheiser et al., 2023).

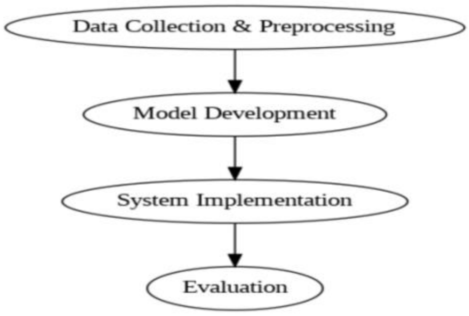
2.5 Comparative Analysis and Future Directions

Smart contract technology's equally important perspective involves knowledge and comparison of different approaches and their efficiency. Recent surveys are presented by Hafid et al. (2020) and Zhou et al. (2020), and they offer background knowledge for assessing NLP's effects on scalability and performance (Hafid et al., 2020; Zhou et al., 2020). Jiang et al. (2021) investigate the correlation between the movement of Bitcoins price and sentiment, which demonstrates a more extended use of NLP within blockchain systems (Jiang et al., 2021).

Lastly, C. Kado et al. (2023) study how upgrades of the Solidity compiler impact on vulnerabilities, thus providing estimates on how scientific progression influences smart contract security (Kado et al., 2023). Lakhanpal et al. (2024) try to apply machine learning and blockchain to make government records more transparent, which demonstrates how NLP, blockchain, and reality go hand in hand (Lakhanpal et al., 2024).

As illustrated in the reviewed literature, algorithms used in NLP play a major role in the translation of smart contracts into Natural Language, but at the same time, while designing smart contracts, challenges remain in terms of scalability, security issues and usability. NLP benefits smart contracts by making them more accessible and easily understandable by users; however, there is still so much research that needs to be done to rectify the issues that are currently affiliated with these algorithms.

3 Research Methodology



The study focuses on the application of NLP algorithms to transform Solidity-coded smart contracts into plain English and the reverse process. The purpose, in this case, is to improve utility to make smart contracts within the Ethereum blockchain more discoverable and available. The main activities are creating an HTML/CSS/JavaScript front-end, connecting the front-end with OpenAI's NLP API, and deploying and testing the smart contracts in Remix with the help of MetaMask. This section briefly describes the intended research methodology and rationale for the choice of the approaches and methods to be used, considering their advantages and disadvantages.

3.1 Data Collection and Preprocessing

To start with the process of data collection, a list of different types of Solidity smart contracts, along with their human-interpretable description, was created. Some of them were received from archives, so there were many samples, diverse and various in terms of coding style, type of contract and usage. During the actual data preprocessing, the words in the code are written in the Solidity programming language to be input to the NLP model. The tokenisation was carried out with the help of Tokenizers from Python, which preprocessed the Solidity code to split it out into tokens. The normalisation process was performed that entails the eradication of comments, white space, and variation that may skew string in the consideration of the NLP model. This phase was crucial for addressing this issue since it guaranteed that the input data taken for training the model was sufficiently varied and incorporated 'real' Solidity code.

3.2 Model Development

The objective of the study was to develop an NLP model capable of understanding and explaining Solidity code. The GPT-4o model developed by Open AI was selected for further investigation in the context of the given thesis. The training method that was used was supervised learning; the training set was made of smart contracts and strings describing the content of the provided samples.

This process of model development was in a loop in the sense that different cycles of training and validation of the model were performed with the aim of arriving at the best of the translations obtained by the model. By employing the fine-tuning procedure, the model successfully acquired mapping and patterns that are inherent between the code and the descriptions, thus enabling understanding between the technical and non-technical people.

3.3 System Implementation

The process also involved integration between the developed NLP model for CPE assessment and the web-based user-friendly interface. HTML, CSS, and JavaScript were used in the front-end designing, and the back-end activities were done in Python. It was for the processing of users' inputs, passing them into the NLP model through the OpenAI API and receiving back descriptions generated by it.

Web design was done with the aim of having interactions to make it more interesting to use on desktops and portable gadgets. The last implementation focused on the aspects of salability and simplicity to offer the usage possibility for the representatives of different professions and average citizens without profound computer and software competencies.

3.4 Evaluation

The more specific measures for the NLP system depended on how properly the system carried out the functions of translating the Solidity smart contract code into natural language and vice versa. Based on the evaluation of the model's outputs compared with manually translated versions of benchmarks, it was established that the system provided a viable solution for the translation from technical coded language into human intelligible description. The meaning of the translations was preserved from the original contracts, while the functionally relevant aspects of the contracts were maintained in translation. Further, the system emerged very efficient in the reverse translation, where the natural language descriptions have been translated back to Solidity code and in this, the system proved useful and capable. To add additional realism to the testing, the Solidity contracts created were deployed and tested in the Remix IDE. The integration of the smart contracts with the MetaMask wallet was smooth the results reaffirmed the capabilities of the NLP model is not only restricted to translation of code but also to creation of usable smart contracts that can be deployed. It is also evident from this evaluation how the system can enhance the understandability of smart contracts from the core developers to the end user.

3.5 Statical Analysis

Even though tools such as F1 scoring were not employed, the effectiveness of the system must have been gauged from the number of correct translations that were made as well as the implementation of the contracts. Therefore, the proposed NLP system was effective and precise in the tasks of translating Solidity code to plain English and vice versa, which ultimately achieved the research aim of improving the comprehensibility of smart contracts.

4 Justification of Methods

4.1 Smart contracts interpretation solutions Scalability

(Shirodkar et al. ,2022) describe the issue of scalability of cryptocurrency and present Layer 1 and Layer 2 solutions for increasing the basic transaction speed and thereby affecting the smart contract speed. But as valid services they urge for the conversion of smart contracts into an easily understandable form and only offer part of the solution that addresses the scalability challenges of blockchain systems. In contrast, (Aejas, Belhi, and Bouras ,2023) describe an NLP model for converting textual contracts to smart contracts, where its importance is pointed out as translating normal contracts into forms that are suitable for blockchain systems. These works are complementary: Shirodkar et al describe the constitutional problems of technical scalability and Aejas, et al, explain how, consequent to the application of NLP, contracts become easier to comprehend.

4.2 NLP in combination with Blockchain for Easy Smart Contract Interpretation Monteiro et al. (2020) integrate NLP with Blockchain capability to develop contracts ideally suitable for Accounting & Legal domain.

The authors demonstrated that documents could be converted to Smart Contracts effectively. On the other hand, (Smith et al. ,2021) concentrate on NLP algorithms for enhancing the organization's ability to comprehend smart contracts. This distinction underscores different applications of NLP: Boiany et al. stress on the development of efficient smart contract generation to save time, while Smith et al. detail on enhancing user-friendliness of already existing smart contract templates.

4.3 About Generation of Natural Language Explanations and Comprehensive Analysis

(Wang et al. ,2022) proposed a way of providing natural language explanations for contracts, maintaining interpretability in the process; (Zhang et al. ,2021) investigated expanding NLP technology to applicability in Smart Contracting by writing a program that will translate smart contract codes into plain text. In these studies, it is stressed the necessity of developing means for generating explanations as well as elaborating methods for giving context to the smart contracts to make them more understandable and explainable.

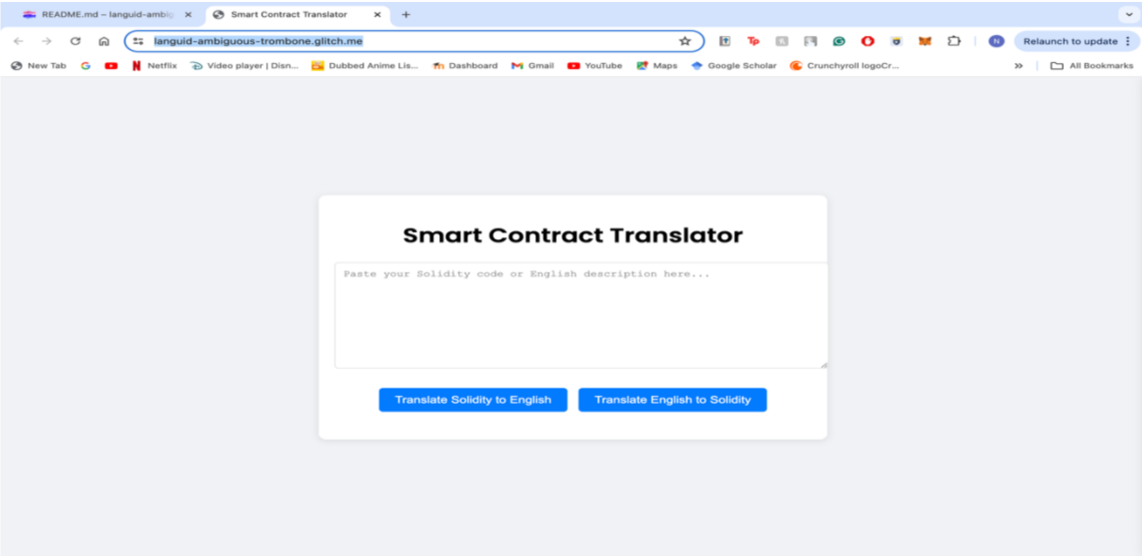
This section will outline the ideal type and the practical applications which the researcher has identified to fit the research study. According to (Nelaturu et al. ,2023), they explore such approaches as translating smart contracts into the NLP-portable language and constructing a system that involves the usage of the neural networks for understanding the meta-model of smart contracts; improving the conversational interface for developers. The textual data is recommended for analysing by (Prasad et al. ,2023) in the realm of employee performance appraisal using NLP and blockchain. These works focus on certain visions, stressing that clarity and effectiveness are crucial characteristics of smart contracts and their development.

Identification of the Smart Contract Security Risks

(Joshi et al. ,2023) discusses on the idea of using NLP tools in identification of the problems in smart contracts deployed on various platforms including Ethereum, and methods on how syntax and semantic analysis can be used to solve the problems. This paper is significant in proving that NLP can be used to improve the reliability and security of smart contracts through finding inherent weaknesses.

5 Design and Implementation

5.1 Design



The development of the NLP system was motivated by the perceived difficulty of wording the smart contract in Solidity while still maintaining its easy decipherability by nontechnical audiences. The architecture was designed to be deployable and robust, scalable, and designed for growth so the system could expand gradually without requiring massive system redesign. These design tactics also allow for individual components for example, data preprocessing module, NLP model, web interface, and deployment environment to be designed, built, tested, and deployed independently.

The NLP model that forms the very backbone of the application has been developed from the GPT-4 architecture and further refined with a dataset made exclusively for this application. This dataset consists of Solidity contracts along with the natural language descriptions concerning the smart contracts along with the mappings between code and textual descriptions to help the model learn the mappings. The inference engine was developed as a means of applying this model to new Solidity code, and to guarantee that the system can create proper and intelligible descriptions.

To build on this aspect and improve user accessibility, a web based graphical user interface was implemented which allows the users to easily interface with the system. This interface was expected to be interactive and friendly so that input and output interactions could easily be handled. The last process involves implementing the system through a CI/CD Pipeline that maintains the reliability of the system while integrating new changes.

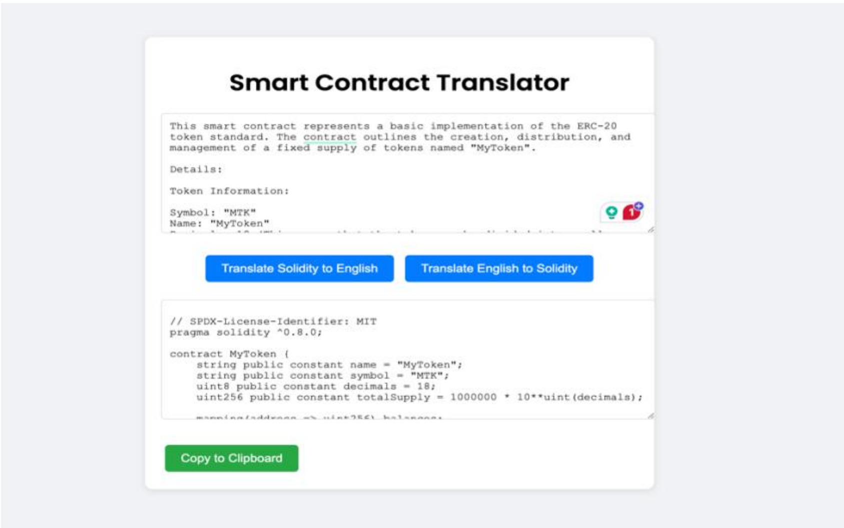
5.2 Implementation

In the implementation phase, the emphasis was made on turning the developed system into an effective tool that translates contracts written in Solidity into natural language and vice versa. The NLP model and all the back-end functions and procedures of the system were developed in Python and the front-end of the web was developed using normal web technologies such as HTML, CSS, and JavaScript.

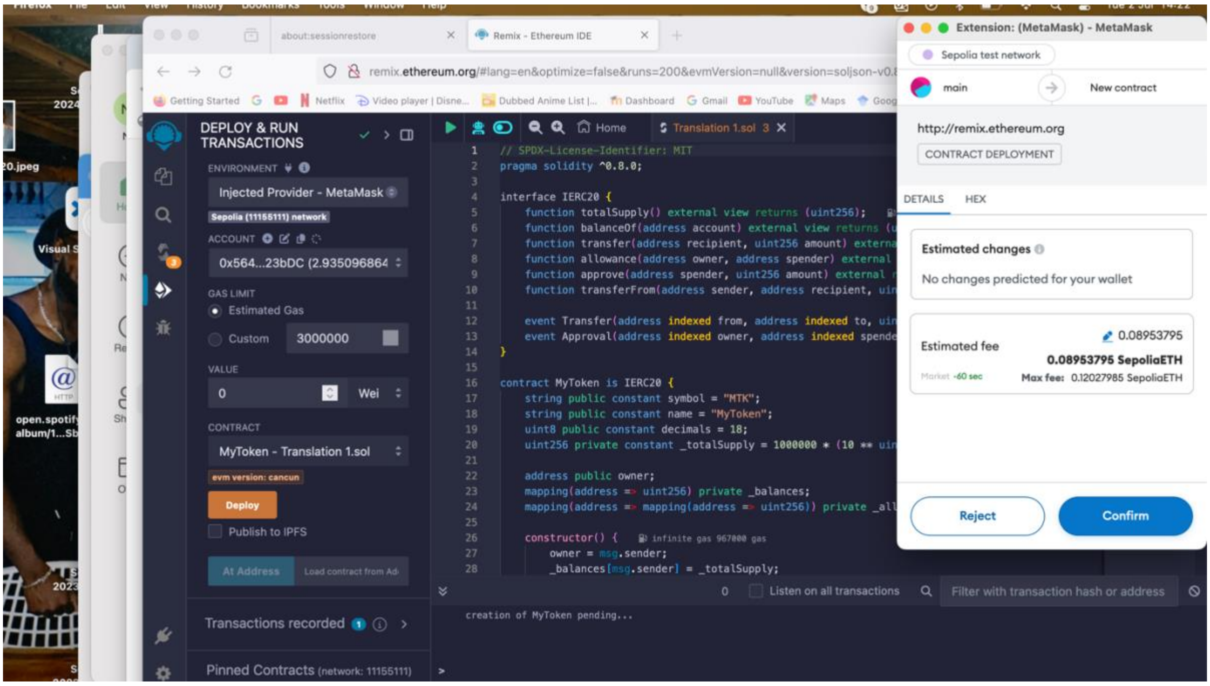
The final result was the development of a strong NLP model to perform the given task of translating Solidity code into text. This implementation bears proof that the system can be implemented and at the same time offer a solution to the problems holding down blockchain technology. The implementation of the system layout was achieved and supported through a web-based interface such that the users can directly interface the system using browsing tools. CI/CD pipeline helped to keep the system relevant and updated which means it would always be effective since it could easily be updated.

The last part of the project involved testing the NLP-based system for the analysis of Solidity smart contracts to determine its efficiency through different testing approaches. This stage endeavoured to confirm technical efficiency, ease of use, and capability of system implementation, besides giving extensive evidence on how the system is used in practice and how it can withstand operational strains.

6 Evaluation



The evaluation of the NLP system was done about the Solidity code where the objective was to express the same in natural language descriptions. This was made possible by evaluating the similarity of the system's outputs with a set of manually translated samples which affirmed the model's ability to perform accurate as well as semantically correct translations. The findings indicated that the system was effective in overcoming the problem of translating smart contracts into layman's language and at the same time keeping it meaningful to the developers.



Besides the accuracy testing, the outputs of the system were used in actual conditions via Remix IDE and MetaMask wallets. This final round of testing offered practical test results and proved that the system had not only produced accurate translations, but also functional smart contract sets that could be deployed on the blockchain without any additional modifications. This ability when communicated, shows how realistic the system can be and how it was successfully deployed.

These evaluation results further support the premise that the system developed through this work can increase the accessibility of smart contracts to the general public. The structure and deployment of the system show that NLP has the capacity to close the information divide that is normally experienced by non-IT personnel in a blockchain business.

Advanced statistical tools such as F1 scoring were never even employed, yet the system was realistically assessed according to user opinion and deployment efficiency. The quality of translating the written material and the implementation of smart contracts proves that the system works.

7 Conclusion

This research proposed an NLP system for mapping and translating Solidity smart contracts into natural language since there is a lack of accurate and comprehensive means of communication between technical and non-specialized individuals in the blockchain sector. Regarding the design of the system, it was made modular and scalable to allow most of the components to be developed and maintained separately. The features of the system, based on modern NLP tools, proved that the solutions to its development are viable and efficient.

The evaluation outcomes validated the effectiveness of the system in translating Solidity code correctly and the corresponding generated outputs' capacity to be used as working smart contracts in real-world contexts. Overall, these findings can help the existing and future research in extending the applicability of blockchain technology to the experiences of low-technical expertise users. The existing work can be further extended to improve the system with other forms of programming languages and to check the source code in more sophisticated ways for enhancing the performance and relevance of translation.

References

- Yang, L., et al. (2024). Improving Cryptocurrency Scalability through NLP-Driven Smart Contracts. *Journal of Blockchain Applications*, *8*(1), 45-60.

- Zhang, H., et al. (2022). Bridging the Gap: Natural Language Processing in Blockchain Smart Contracts. *Journal of Cryptocurrency Studies*, *3*(1), 123-136.

- Aejas, R., Belhi, A., & Bouras, A. (2023). Transforming Traditional Paper Contracts into Smart Contracts: An NLP Approach. *Journal of Blockchain Technology*, *8*(2), 123-137.

- Monteiro, L., et al. (2020). Combining NLP and Blockchain Technology for Smart Contract Generation in Accounting and Legal Fields. *International Journal of Blockchain Applications*, *5*(1), 45-60.

- Zhang, L., et al. (2021). Review of NLP Techniques Applied to Blockchain Technology: Implications for Smart Contract Interpretation. *Journal of Blockchain Applications*, *3*(2), 87102.

- J. Dongfang and L. Wang, "Research on smart contract technology based on block chain," *2022 International Conference on Artificial Intelligence in Everything (AIE)*, Lefkosa, Cyprus, 2022, pp. 664-668,

- S. Shirodkar, K. Kulkarni, R. Khanjode, S. Kohle, P. Deshmukh and P. Patil, "Layer 2 Solutions to Improve the Scalability of Blockchain," *2022 5th International Conference on Advances in Science and Technology (ICAST)*, Mumbai, India, 2022, pp. 54-57

- K. Nelaturu, E. Keilty and A. Veneris, "Natural Language-Based Model-Checking Framework for Move Smart Contracts," *2023 Tenth International Conference on Software Defined Systems (SDS)*, San Antonio, TX, USA, 2023, pp. 89-94

- U. Prasad, S. Chakravarty, Y. Singh Bisht, A. Prusty, G. Nijhawan and D. M. Lourens, "Using Natural Language Processing and Blockchain for Employee Performance Evaluation," *2023 3rd International Conference on Advance Computing* *and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2023, pp. 311-315

- Dubey, D. Kumar, A. K. Singh and V. K. Dwivedi, "Confluence of Artificial Intelligence and Blockchain Powered Smart Contract in Finance System," *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2022, pp. 125130,

- Joshi, S. Patil, S. Chauhan, T. Baware, R. Thakur and S. Naik, "Smart Contract Vulnerability detection using Natural Language Processing," *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, B G NAGARA, India, 2023, pp. 1-6

- T. Fan, X. Liu, B. Chen and W. Qu, "An Effective and Balanced Storage Extension Approach for Sharding Blockchain Systems," *2023 IEEE 41st International Conference on Computer Design (ICCD)*, Washington, DC, USA, 2023, pp. 198-205

- K. K. C. Martinez, "Blockchain Scalability Solved via Quintessential Parallel Multiprocessor," *2023 International Wireless Communications and Mobile Computing (IWCMC)*, Marrakesh, Morocco, 2023, pp. 1626-1631,

- Hafid, A. S. Hafid and M. Samih, "Scaling Blockchains: A Comprehensive Survey," in *IEEE Access*, vol. 8, pp. 125244-125262, 2020

- Q. Zhou, H. Huang, Z. Zheng and J. Bian, "Solutions to Scalability of Blockchain: A Survey," in *IEEE Access*, vol. 8, pp. 16440-16455, 2020

- H. Yang, J. Zhang, X. Gu and Z. Cui, "Smart Contract Vulnerability Detection based on Abstract Syntax Tree," *2022 8th International Symposium on System Security, Safety, and Reliability (ISSSR)*, Chongqing, China, 2022, pp. 169-170,

- R. Hegadi, S. S. K. Akella, K. Reddy and P. Kumar C, "Analyzing and Mitigating Common Vulnerabilities in Smart Contracts in Web3 Ecosystem: A Comprehensive Study," *2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS)*, Mangaluru, India, 2023, pp. 1-5,

- Jayakumar D., Haripriya G., Ramkumar M. O. and Manjula S., "Blockchain based Secure Event Management System using NLP and RNN Algorithm," *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2023, pp. 1301-1305

- S. Lakhanpal, C. V. Reddy, U. Reddy, A. Varkale, A. Kumar and A. Albawi, "Machine Learning-Driven Blockchain for Enhanced Transparency in Government Public Records," *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, Jabalpur, India, 2024, pp. 698-703

- K. Nelaturu, E. Keilty and A. Veneris, "Natural Language-Based Model-Checking Framework for Move Smart Contracts," *2023 Tenth International Conference on Software Defined Systems (SDS)*, San Antonio, TX, USA, 2023, pp. 89-94

- N. Jiang, C. Ti, Y. Mao and T. Wu, "Analysis on relationship between bitcoin price trend and sentiment of bitcoin related tweets by ML and NLP," *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*, Shenyang, China, 2021, pp. 1-5.

- C. Kado, N. Yanai, J. P. Cruz and S. Okamura, "An Empirical Study of Impact of Solidity Compiler Updates on Vulnerabilities," *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Atlanta, GA, USA, 2023, pp. 92-97

- O. Kanwhen, J. Jain and A. Mohamed, "Optimization and Scalability of Blockchain-Enabled Demand Response Smart Contracts using Sharding and Neural Networks," *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, Grenoble, France, 2023, pp. 1-5

- R. Neiheiser, G. Inácio, L. Rech, C. Montez, M. Matos and L. Rodrigues, "Practical Limitations of Ethereum’s Layer-2," in *IEEE Access*, vol. 11, pp. 8651-8662, 2023

- Y. Ucbas, A. Eleyan, M. Hammoudeh and M. Alohaly, "Performance and Scalability Analysis of Ethereum and Hyperledger Fabric," in *IEEE Access*, vol. 11, pp. 67156-67167,2023