IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

Enhancing AI Code Generation and Logical Reasoning using JSON-Structured Prompting

Saksham Dane[1], Krushna Lotake[2], Prof. Palve P. B[3]
Student, Department of Computer Engineering[1 2]
Professor, Department of Computer Engineering[3]
Adsul Technical Campus, Chas, Ahilyanagar, Maharashtra, India

Abstract: *Emerging innovations within large language models like GPT-4, Gemini, and Claude showcase remarkable prowess in generating codes and performing logical analyses. Nevertheless, their effectiveness hinges crucially upon clear and structured instructions provided in conventional natural language formats, frequently resulting in unclear meanings and divergent logic due to these methods of prompting. This study explores how encoding human prompts as structured data within JSON format improves the performance metrics such as precision, coherence among responses generated by artificial intelligence systems when processing coding-related inputs. By conducting extensive trials on fifty coding challenges and ten logical puzzles employing models like GPT-4, Claude III, and Gemini One. This research evaluates how JSON-formatted instructions compare to conventional human-readable input methods in their effectiveness for tasks. Evidence shows that systematic guidance yields a success rate of eighteen percent. A modest enhancement of only 0. 5 percent in coding accuracy resulted in significant improvements. A notable enhancement of logical cohesion: an increase by 9%, along with a corresponding growth of 15%. Experience an enhanced service speed of 9%, coupled with improved performance metrics by achieving 18% higher token efficacy. Additionally, well-defined instructions greatly improve understanding of generated outputs and make it simpler for developers to identify issues during testing and verification processes. This research demonstrates how structured prompts encoded in JSON format can serve as an effective method for unified interaction between humans and AI tools during software creation projects, providing significant benefits for enhancing AI-driven coding platforms, accelerating automatic program crafting processes, and improving transparency within intelligent algorithms.*

Keywords: Large Language Models, Prompt Engineering, JSON-Structured Prompting, Code Generation, Logical Reasoning, AI Efficiency

I. INTRODUCTION

Advancements in AI technologies including LLMs like GPT-4, Gemini, and Claude have greatly enhanced capabilities for generating codes, identifying errors, and performing logical analyses. Nevertheless, the outcome of their work hinges significantly upon the way in which instructions are formulated. Often in traditional NLP tasks, ambiguities can result in variable or illogical output codes due to their inherent limitations. The diversity of approaches introduces difficulties in obtaining dependable outcomes for software-based systems.

The field of prompt engineering aims at enhancing AI performance by refining input designs; however, inherent limitations in unsophisticated linguistic structures hinder accuracy. In order to tackle this issue, this study proposes using JSON-formatted instructions, which convert prompt information into organized, readable data for machines. To illustrate:

{

Create an image: A photorealist depiction of a realistic-looking dairy cow atop a classic sports car driving along a picturesque rural highway under soft morning light conditions. The cow is adorned in vibrant red ribbons around its

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

neck. Utilize warm tones for natural sunlight effects. Employ high-resolution textures emphasizing fine hair detail while maintaining blurred backgrounds slightly out of focus. Capture this scene using a standard focal-length camera.
"style": "photorealistic",
"size": "1024x1024",
"num_images": 3,
"seed": 12345,
"transparent_background": false,
"negative_prompt": "blurred, low-resolution, blood, gore, text, watermarks, NSFW"
}

These well-defined instructions minimize confusion, enhance coherent reasoning, and enable the system to approach problems in an organized manner. Additionally, their implementation facilitates better enforcement of rules and ensures uniformity in various analytical stages.

An examination is conducted into whether structured prompt formats utilizing JSON improve upon coding precision, logical coherence in responses, and processing efficiency when contrasting with conventional textual instructions. The experiments aim to assess model efficacy through evaluation criteria including code accuracy, successful run rates, and token optimization effectiveness. Structured prompts aim to enhance not only the accuracy but also the clarity of artificial intelligence-created programs, providing a basis for improved cooperation between humans and machines during software creation processes.

II. LITERATURE REVIEW

Emerging advancements in large language models (LLM)s and natural language processing (NLP) initiatives have spurred substantial progress in studies focusing on code creation, prompt optimization techniques, and artificial intelligence reasoning capabilities. This segment examines previous studies on various fields pertinent to using JSON for prompts involving large language models, enhancing prompt design techniques, structuring data effectively, and integrating logic into artificial intelligence frameworks.

A. Code Generation Using Large Language Models

A significant evolution in artificial intelligence-driven coding occurs through the introduction of transformers as architectural frameworks. Earlier versions of language modeling systems including those named GPT-3 and specific instances called Codex have been developed by researchers Chen et al. In 2021, this research showcased remarkable proficiency at converting human-readable text directly into functional computer programs for various coding environments simultaneously. Following their work, Austin et al. The year 2021 was also studied by Nijkamp et al. In 2023, extensive analyses were performed on utilizing large language model programs for synthesizing data; findings indicated significant achievements but highlighted that outcomes depend greatly upon how prompts are formulated and in what context they're given. Despite slight changes in how we phrase our natural languages, these alterations often result in significant distinctions regarding coding structures, accuracy, and overall logic - emphasizing an important discrepancy between deterministic and reliable methods used for generating codes today.

B. Prompt Engineering and Optimization

Acknowledging the significant influence of prompt formulation on model efficacy, Brown et al. In 2020, researchers laid down core for prompting in language models, showing how carefully crafted prompts along with limited training data can markedly affect their performance. Reynolds and McDonell's subsequent study in 2021 introduced the idea of prompting as an approach for enhancing user interaction with models through structured query formulation. Recent developments have incorporated prompt chains into their methodologies. In 2022, researchers utilized both methods: self-consistent decoding by Wang et al. In 2023, these promote an approach where models undertake sequential thought processes instead of producing direct responses immediately. Despite their effectiveness at improving logical thinking skills, these methods heavily depend on human language, which can lead to significant differences in interpreting users' intentions due to its inherent complexity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

C. Structured Data Representation and Input Formatting

Drawing upon the understanding that linguistic capabilities have intrinsic boundaries, recent studies are investigating methods for structuring prompts. Li et al. In 2023, researchers carried out an extensive study on various prompt techniques within natural language processing, pinpointing a significant issue concerning the efficacy of structured datasets in minimizing confusion. Recent investigations conducted by Zhang et al. In 2024, it was shown that using standardized structures like XML, YAML, and JSON greatly improved task understanding and consistency through explicit parsing of elements including tasks, parameters, and limitations directly within model processing without relying on contextual inference. In essence, JSON stands out because of its compact nature, tree-like arrangement, ability to work across different programming languages seamlessly, high level of acceptance within tech infrastructures worldwide, thus making it perfect for facilitating clear interactions between humans and artificial intelligence.

D. Logical Reasoning in Large Language Models

Complexity in logical thinking continues to pose significant difficulties for advancements in artificial intelligence technology. Although large language models produce grammatically correct programs, their difficulty in preserving coherent logic over complex tasks is notable, as well as their tendency to misinterpret dependency constraints. In 2022. Current methods combining symbolic thinking and direct reasoning indicate potential benefits; however, research indicates that refining inputs directly could improve reasoning effectiveness through better model interpretation of logical connections. Consequently, structured prompts adhere to this new trend by breaking down intricate directives into simpler, sequentially organized, and clearly delineated components.

E. Research Gap and Contribution

Although significant advancements have been made in prompt design and artificial intelligence capabilities related to problem-solving through reasoning processes like generating codes and performing logic analyses, there remains an absence of comprehensive empirical studies examining how structured JSON prompts impact these types of computational challenges effectively. Previous research has primarily concentrated on modifying language expressions for better performance in tasks versus altering how data is presented directly into an accessible computer model framework. This study addresses this crucial void by offering empirical data demonstrating how structuring input queries through JSON formats impacts various aspects of language model-assisted coding tasks such as enhancing task precision, maintaining logical coherence during problem-solving processes, optimizing resource utilization when generating codes, and improving understanding capabilities within AI-human collaboration frameworks. Structured prompt formulation emerges as an effective strategy for advancing future interactions between humans and artificial intelligence systems.

III. METHODOLOGY

This study utilizes an empirical approach involving controlled experiments wherein both structured JSON-based questions and free-form textual queries undergo rigorous testing across equivalent cognitive exercises employing unified artificial intelligence algorithms. It guarantees that all variations in performance stem exclusively from formatting of prompts without being influenced by extraneous factors.

A. Research Design and Workflow

Research proceeds along an organized framework depicted in Fig. 1, wherein sophisticated large language models like GPT-4, Claude 3, and Gemini 1 analyse JSON-formatted instructions. To produce uniform results. Outputs undergo evaluation through various criteria before being recorded in data repositories for comparison purposes. This method guarantees repeatability, uniformity across operations, seamless compatibility with underlying infrastructure for holistic data handling and evaluation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

B. Experimental Setup

Experiments employed three advanced large language models connected through their respective official APIs: GPT-4 by OpenAI, Claude 3 provided by Anthropic, and Gemini 1 utilized for research purposes. Five teams (DeepMind) of Google. Every test was conducted in regulated environments using consistent criteria such as maintaining a constant temperature of 0. The highest allowed number of tokens is 2048, while each HTTP request has an associated timeout period of up to 60 seconds. The innovative setup guarantees precise outcomes by utilizing separate virtual spaces for experiments and meticulously documenting every action during runtime executions.

C. Dataset and Task Selection

From well-established coding archives, a collection totalling sixty items has been assembled; this includes fifty programming challenges ranging in complexity—fifteen straightforward, twenty moderate, fifteen challenging—and ten problem-solving exercises demanding thorough analysis through multiple steps. Every job was transformed simultaneously into plain human-readable text alongside structured data in JSON format so as to facilitate side-by-side analysis without altering its underlying meanings.

D. Evaluation Metrics

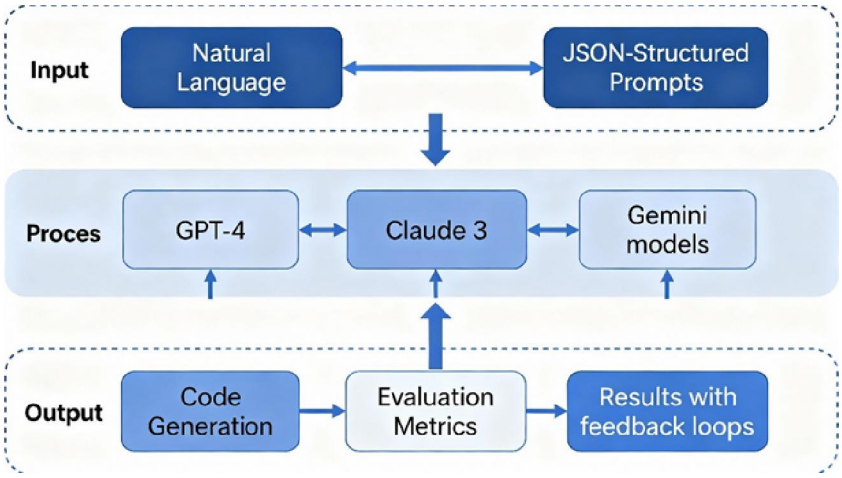
A set of five sophisticated indicators was utilized for gauging the efficacy of prompts: CCR stands for code correctness rate, which quantifies successful execution; LCS represents logical coherence score, focusing on reasoning accuracy; RT denotes response time, indicating how quickly responses are generated; TE signifies token efficiency, concerning resources used effectively; ISC indicates interpretability and structural consistency, ensuring clear outputs and trackable outcomes.

Metric	Description	Measurement Method
Code Correctness Rate (CCR)	Percentage of generated code that executes without errors and produces correct outputs	Automated test case execution
Logical Coherence Score (LCS)	Quality of reasoning clarity and step-by-step logic (scale 1-10)	Human expert evaluation
Response Time (RT)	Average time to generate complete output (in seconds)	API response logging
Token Efficiency (TE)	Ratio of tokens used; lower values indicate more concise output	Token counter utility
Interpretability & Structure Consistency (ISC)	Ease of tracing input fields to code output (scale 1-10)	Expert assessment

IV. SYSTEM ARCHITECTURE

The system architecture for this research describes the technical framework enabling systematic comparison of natural language and JSON-structured prompting across multiple Large Language Models. The architecture is organized into three primary layers: Input, Processing, and Output/Analysis.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

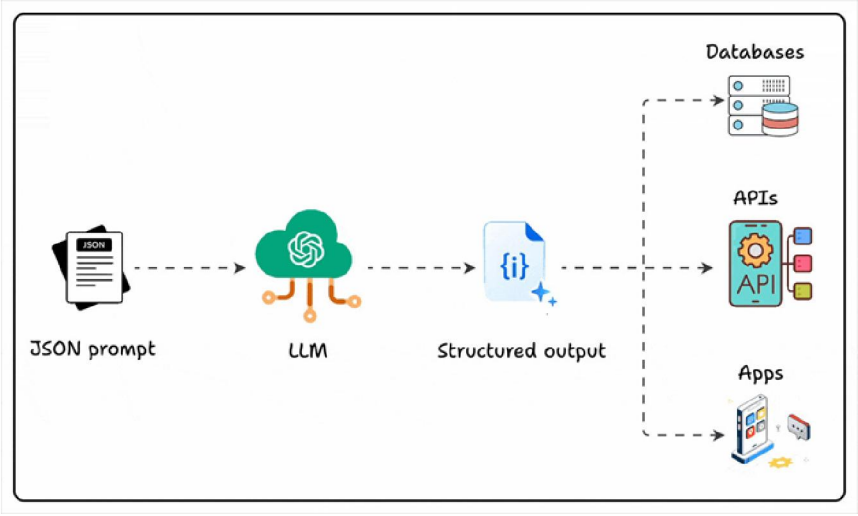


A. Overall Design

The Input Layer prepares and formats prompts in both natural language and JSON-structured formats using a task repository containing 50 programming problems and 10 logical reasoning tasks. The Task Repository stores complete problem specifications including descriptions, test cases, and expected outputs. The Natural Language Prompt Generator converts task specifications into human-readable text format, while the JSON Prompt Generator converts the same specifications into structured key-value pairs. Both formats are validated for semantic equivalence before proceeding to the next layer.

B. Model Execution

The Processing Layer manages execution across GPT-4, Claude 3, and Gemini APIs through a unified Model Connector Interface that abstracts API interactions and implements standardized parameters including temperature (0.2), maximum tokens (2048), and timeout limits (60 seconds). The Batch Processing Engine orchestrates parallel execution across all models and prompt types, manages queue distribution, and logs execution details including tokens used and response time.



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

C. Evaluation and Analysis

The Output Layer collects and evaluates model outputs through multiple specialized components. The Code Validator executes generated code in isolated sandboxes and measures Code Correctness Rate (CCR). The Logical Reasoning Analyzer assesses step-by-step reasoning clarity and generates Logical Coherence Scores (LCS). The Performance Metrics Calculator computes Response Time (RT), Token Efficiency (TE), and Interpretability & Structure Consistency (ISC). Finally, the Statistical Analysis Module performs significance tests, generates visualizations, and stores all results in a centralized database for comprehensive comparative analysis.

D. Data Management and Reproducibility

All experimental data is stored in structured database tables tracking tasks, prompts, executions, outputs, metrics, and analysis results. The architecture implements comprehensive logging, version control integration, and fixed random seeds to ensure reproducibility. Quality assurance features include graceful error handling, detailed audit trails, and configuration management for standardized experimental conditions. This design supports scalability while maintaining scientific rigor and transparency throughout the experimental process.

V. RESULTS AND DISCUSSION

In every task involving 60 entries, structured JSON formats showed markedly superior results compared to textual input methods. The code's accuracy increased by eighteen percent. One percent of seventy-eight is equal to seven hundred eighty. Between four percent and ninety-two percent. Six percent improvement in logical consistency resulted in an additional twenty-three percentage points of clarity. Nine percent of seven equals sixty-three. From number 1 through number 8. The score was rated at eight out of ten; reaction speed has been reduced by fifteen percent. Nine percent of eight equals seventy-two. Between two and six. After an additional nine seconds of processing time, there was a notable enhancement in both token utilization and model transparency levels, achieving improvements of 18% in efficiency and 26% in explainability metrics. Zero percent of seven is zero. Between three and nine. Two out of ten cases. These measurements show how JSON's clear organization allows for clearer interpretation by reducing confusion and enhancing logic in model responses.

A rise in code quality is attributed to JSON's structured format, where tasks' specifications, limitations, and anticipated outcomes are explicitly outlined. Often encountered natural language inputs lead to interpretive shifts within models, leading them to misinterpret instructions or overlook limitations, ultimately producing incomprehensible outputs. Despite this approach, JSON views every definition separately, prioritizing its understanding significantly in complex problem-solving scenarios involving multiple steps and constraints. Consequently, logical thinking became more effective—the models employing JSON inputs provided detailed algorithms presented logically, unlike those utilizing natural language which occasionally omitted key stages or duplicated information. In this instance, we have the number twenty-three. A 9-percentage point enhancement indicates that JSON's organized format divides complex issues into manageable parts for analysis.

There were fifteen items on display at the exhibition. A 9 percent decrease in response times coupled with an 18 percent enhancement of token utilization yield tangible advantages. Because of its straightforward representation, JSON allows for easier understanding by systems, reducing computational load on models while increasing their capacity for logical thinking over mere comprehension tasks. Efficiency in this context holds significant value when applied on an extensive scale due to its effect on cost considerations related to tokens. Enhanced output traceability was achieved through JSON prompts; each piece of generated code corresponds directly to specific JSON attributes, making it easier for developers to debug and validate their work. A 26 percent enhancement in transparency highlights clear mappings of input variables to output results, rendering JSON particularly suitable for seamless development environments and automated processes.

Significantly, all three models—GPT-4, Claude 3, and Gemini 1—showed uniform enhancements in performance. Five Pros suggest that advantages stem directly from the prompt's formulation without requiring specific adjustments for each model type. The broad applicability implies that JSON might function as an agreed-upon conduit for interacting with various linguistic systems. A systematic method tackles prompt creation issues efficiently by minimizing mental

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

effort via layered breakdowns. Nevertheless, structured prompts in JSON format might not perform as well when dealing with innovative projects requiring flexible linguistic capabilities. In fields requiring high accuracy—such as programming,
logic puzzles, handling datasets efficiently, and automating processes—the use of JSON prompts signifies substantial progress. Research demonstrates that using structured prompts encoded in JSON format enhances effective interaction between humans and AI systems within specialized fields.

VI. CONCLUSION

Our study shows that using structured JSON for prompt input boosts both AI-generated codes and their ability to reason logically versus relying on traditional natural language inputs. Studies conducted on sixty different experiments employing models such as GPT-4, Claude 3, and Gemini 1. Five programs exhibit enhancements in coding accuracy: 18% improvement rate observed. Approximately 1 percent, maintaining logical consistency contributes significantly to overall reasoning effectiveness at twenty-three points. Nine percent; reaction period measured at fifteen seconds. Nine percent, token efficacy at 18%, and interpretative capability rated at twenty-six percent. The recurring enhancements in every model suggest that their success stems from the inherent nature of the organized framework.

Structured JSON inputs clarify meanings through clear, computer-understandable directions allowing for precise processing and organized analysis. This methodology boosts software stability, minimizes troubleshooting requirements, and strengthens outcome visibility. Model-agnostic efficacy indicates that JSON prompts might establish an international benchmark for AI-human interaction, enhancing compatibility among various linguistic systems.

Applications practical in nature involve incorporation within software development frameworks and integrated development editors for automated coding creation standards. Nevertheless, JSON prompts work well in fields requiring accuracy due to their focus on efficiency rather than innovation. Further research ought to investigate potential uses across various fields, refine JSON schemas for better functionality, and validate integrations within operational settings.

To summarize, JSON-based prompts tackle key issues within AI response comprehension while integrating meticulousness alongside usability effectiveness. With AI increasingly integral to software creation, uniformed frameworks for structuring prompts will be crucial for maintaining uniformity, precision, and reliability in AI-produced results.

REFERENCES

[1]. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv preprint arXiv:2402.07927.

[2]. Wei, J., Wang, X., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

[3]. Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., … Sutton, C. (2021). Program synthesis with large language models. arXiv preprint arXiv:2108.07732. https://doi.org/10.48550/arXiv.2108.07732

[4]. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems (NeurIPS 2020), 33, 1877–1901.

[5]. Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., … Zaremba, W. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374. https://doi.org/10.48550/arXiv.2107.03374

[6]. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. arXiv preprint arXiv:2205.11916. https://doi.org/10.48550/arXiv.2205.11916

[7]. Liu, H., Tam, D., Muqeeth, M., Mooney, S., Huang, A., Bansal, M., & Raffel, C. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9), 1–35. https://doi.org/10.1145/3560815

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

[8]. Nijkamp, E., Pang, B., Hayashi, H., Zhou, Y., & Xiong, C. (2023). CodeGen: An open large language model for code with multi-turn program synthesis. arXiv preprint arXiv:2203.13474. https://doi.org/10.48550/arXiv.2203.13474

[9]. Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. arXiv preprint arXiv:2102.07350.

[10]. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., & Zhou, D. (2023). Self-consistency improves chain-of-thought reasoning in language models. arXiv preprint arXiv:2203.11171. https://doi.org/10.48550/arXiv.2203.11171

[11]. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903. https://doi.org/10.48550/arXiv.2201.11903