SmartGSN : An Online Tool to Semi-automatically Manage Assurance Cases

Oluwafemi Odu, Daniel Mendez Beltran, Emiliano Berrones Gutierrez
Alvine B. Belle, Gerhard Yu, Melika Sherafat

SAFECOMP 2025

Presenter : Gerhard Yu

September 10th, 2025

YORK U

Outline

Background

System Assurance, Assurance Case Pattern, LLM

Motivation

Context, Problem Statement, Objectives

Methodology and Results

Approach, Experiments, Evaluation, Results

Conclusion and Future Work

Conclusion, Next Steps

Questions and Answers

Discussion

System Assurance

Mission-critical systems are increasingly designed to be **autonomous**, **interoperable** and **interconnected**.

Justifying and providing **confidence** in the **essential properties** (e.g., safety, security, reliability) of these systems is crucial to **prevent system failure**.

To ensure **systems comply** with specific **industrial standards** and **relevant laws**.

Assurance Case (AC)

Purpose

- A set of **structured arguments** supported by **evidence** that justifies and demonstrates that a system meets desirable **non-functional requirements** in a given environment.

- **Assurance Cases** support **System Assurance.**

Application

- **Prevents system failures** that could lead to catastrophic consequences like life loss, environmental threats, and financial losses.

- **Certification** in accordance with **industrial standard such as DO-178C** for avionics and **ISO 26262** for automotive.

Representing an Assurance Case

G1: Collision Avoidance Algorithm (CAA) provides correct instructions for avoiding collisions between two UAVS

S1: Strategize over the capabilities of the CAA to give accurate directives to individuals UAVs

G1.1: GPS is accurate within 5 cms

C1: RTK ground station is provided in flying area

S2: Strategize over the GPS accuracy claims

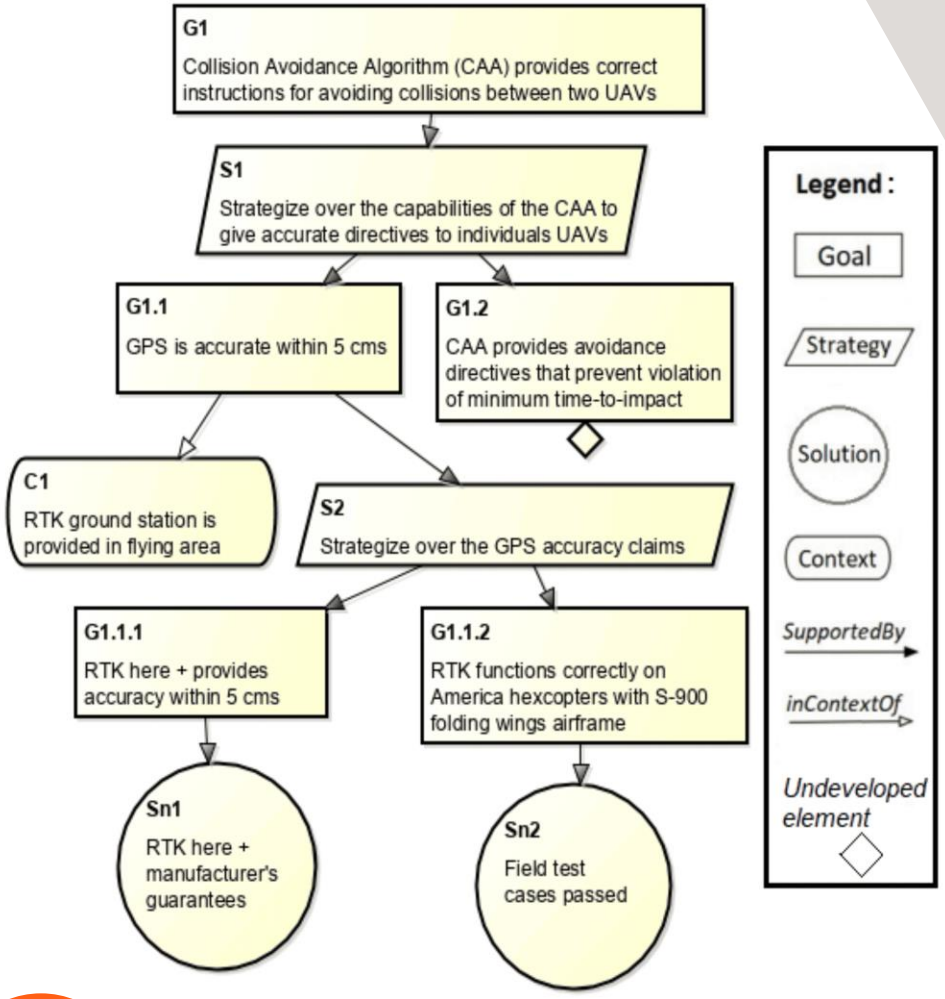
G1.1.1: RTK here + provides accuracy within 5 cms

Sn1: RTK here + manufacturer's guarantees

G1.1.2: RTK functions correctly on America hexcopters with S-900

Sn2: Field test cases passed

G1.2: CAA provides avoidance directives that prevent violation of minimum time-to-impact (Undeveloped)



01 **Structured Prose Representation**

02 **GSN Representation**

Assurance Case Pattern (ACP)

❑ Purpose

- A **template** used to guide and ease the creation of an assurance case.
- It contains **placeholders** with generic information which are replaced with system-specific information during the creation of an assurance case for a given system.
- **Assurance Case Patterns** facilitates the creation of **Assurance Cases.**

❑ Application

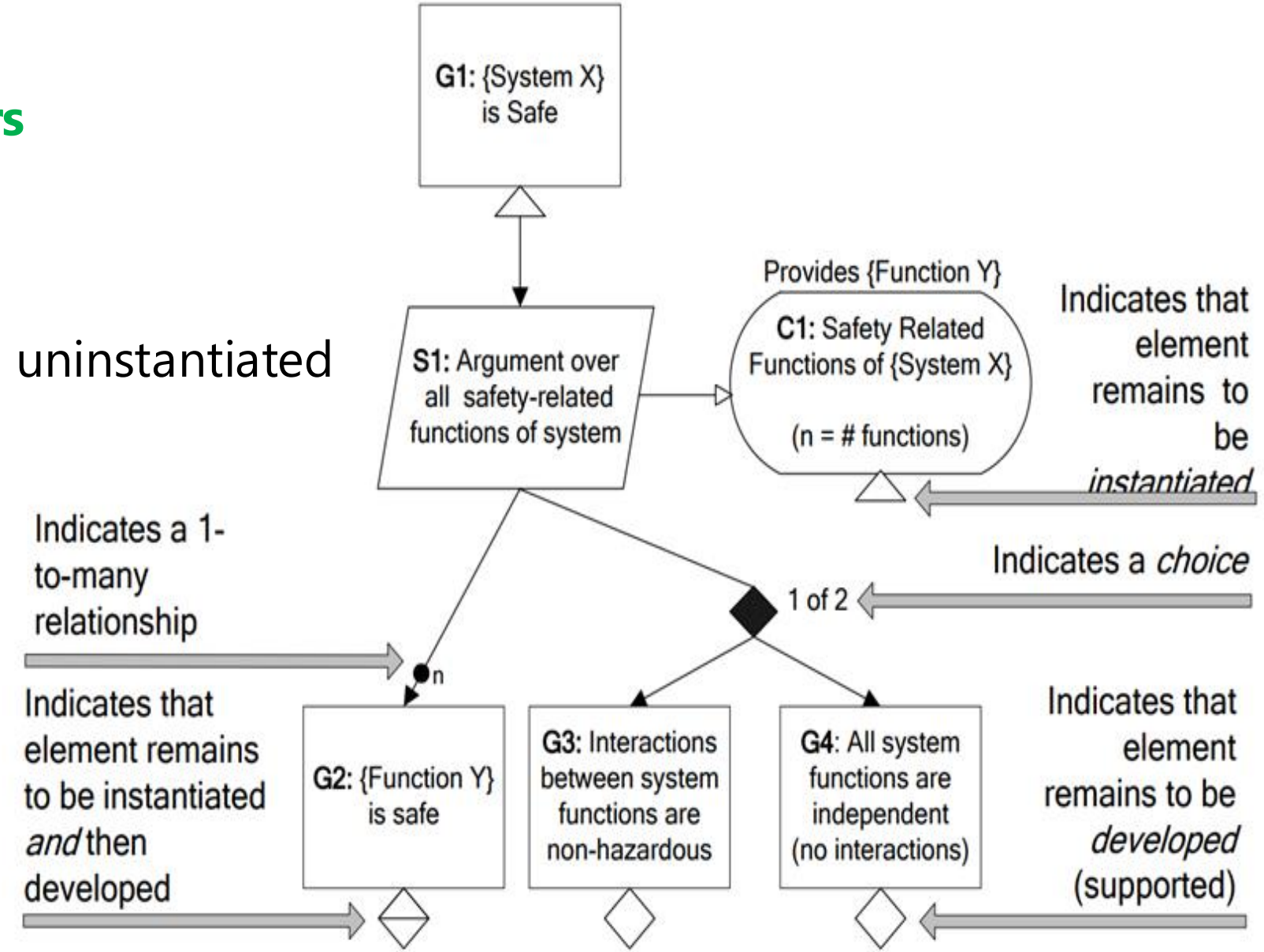
- To facilitate **re-use.**
- To improve the **structure** of an Assurance case.
- To mitigate **assurance deficits.**

Assurance deficit: **Gap in knowledge** hindering complete confidence in an assurance case.

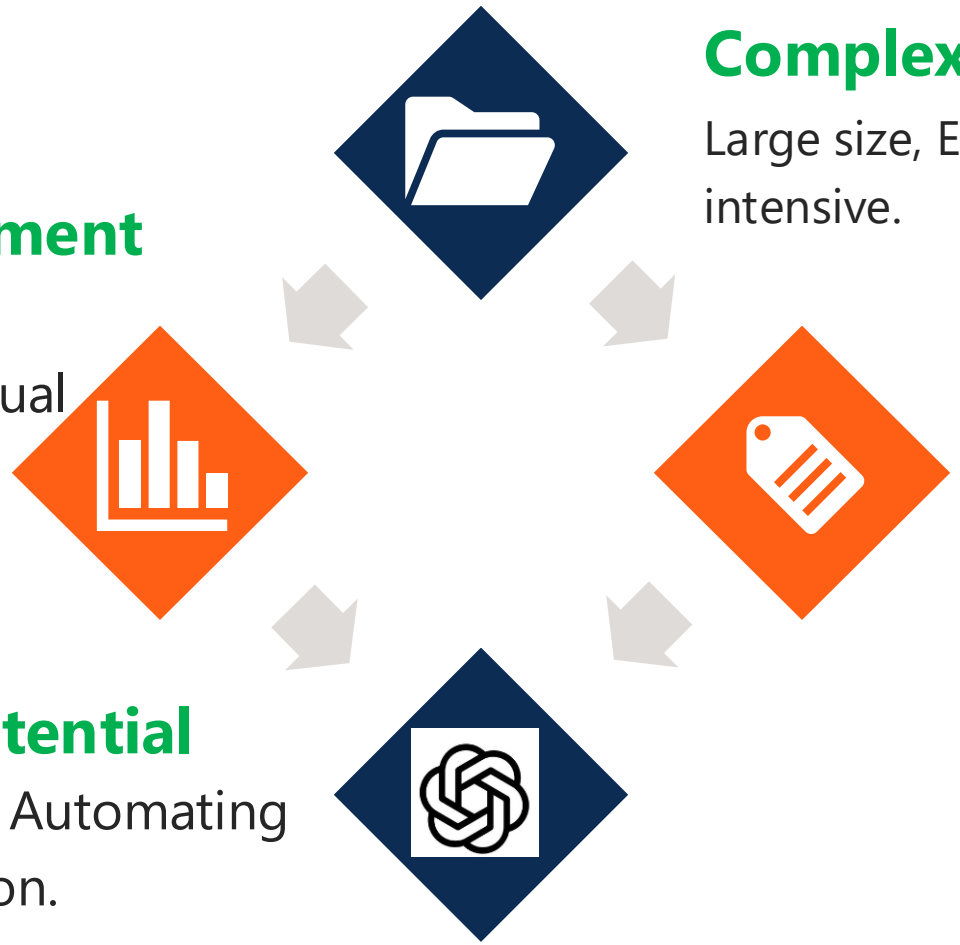
Representing an Assurance Case Pattern

❑ Additional Decorators

- Uninstantiated
- Undeveloped and uninstantiated
- Placeholders
- Multiplicity
- Optionality
- Choice



Motivation



Complexity of Assurance Cases

Large size, Error-prone and Labor-intensive.

Regulatory Review Challenges

Difficulty in detecting Patterns, Assurance Deficits, and areas of non-compliance.

Exploring LLM Potential

Pattern Detection and Automating Assurance Case creation.

Improve the Management of Assurance Cases

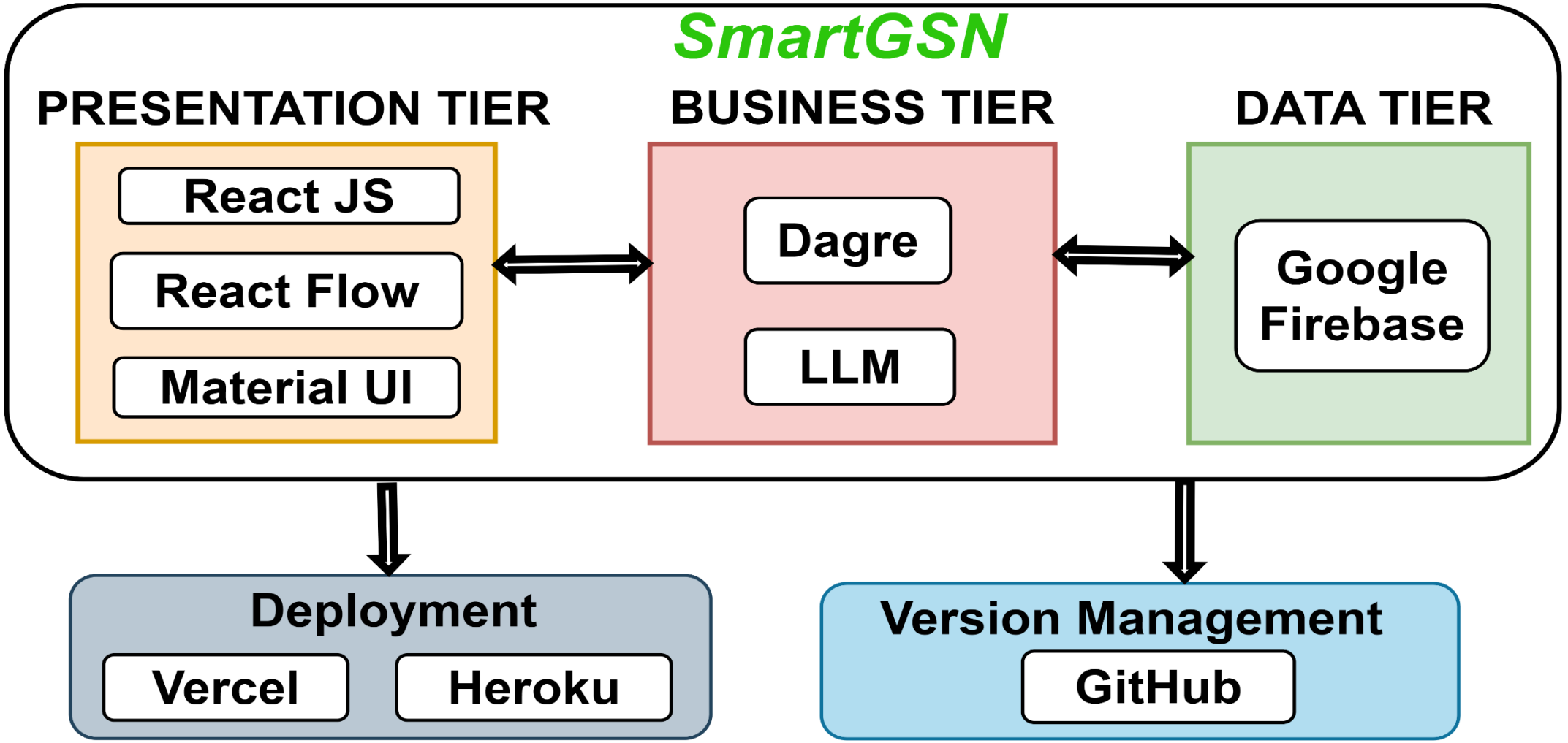
To avoid the pitfalls of manual methods.

Description of SmartGSN

- SmartGSN leverages LLMs to semi-automate the management of assurance cases
- SmartGSN has Four (4) core features
  - **Detection of Assurance case patterns within Assurance Cases.**
  - **Instantiation of Assurance Cases from Patterns.**
  - **Conversion of Assurance Cases from Textual format to Graphical format.**
  - **Creation (Editing) of Assurance Cases.**

Core Technologies Powering SmartGSN

- A **3-tier client/server architecture.**



Research Methodology

Research Questions

We aim to answer the following Research Questions (RQs):

1. Can SmartGSN correctly detect assurance case patterns in assurance cases?

2. How does the choice of metric thresholds impact the ability of SmartGSN to detect assurance case patterns?

Dataset Description

System	Domain	Assurance Case Patterns (ACPs)			Assurance Cases (ACs)	
		Decorators	Placeholders	Elements	Elements	Relationships
ACAS XU	Aviation	11	10	22	24	23
BLUEROV2	Automotive	17	8	18	24	21
GPCA	Medical	6	21	23	27	26
IM SOFTWARE	Computing	1	9	15	24	23
DEEPMIND	Medical	16	26	17	23	23

Experiment Set-up

- **LLM Set-up**
  - GPT-4o
  - GPT-4 turbo

- **OpenAI API Parameters**
  - Temperature: 1
  - Token Length: 4096

- **Prompting Technique**
  - Zero-Shot + CoT (Chain-Of-Thought)

- **Types of Prompt**
  - System Prompt
  - User Prompt

Experiment Set-up

Prompt Structure

- System Prompt

You are an assistant tasked with detecting an assurance case pattern within an assurance case both represented in an advanced structured prose format. Your responsibility is to evaluate the similarity between an assurance case pattern and an assurance case using predefined metrics. Your role is to utilize the contextual information, predicate-based rules and domain information provided to compute the similarity between an assurance case pattern and an assurance case. The metrics include the BLEU score and Semantic Similarity.

The rule for detecting the assurance case pattern within an assurance case is as follows: "If the BLEU score is superior or equal to 'X' AND the semantic similarity score is superior or equal to 'X', conclude that the pattern has been detected in the assurance case. Otherwise, conclude that the pattern has not been detected in the assurance case."

Follow these steps to determine if the assurance case pattern is detected within the assurance case:

Series of intermediate steps on how to determine if an assurance case pattern is detected within an assurance case

@Context_Assurance_Case

Sample context information on assurance case

@End_Context_Assurance_Case

@Context_Assurance_Case_Pattern

Sample context information on assurance case pattern

@End_Context_Assurance_Case_Pattern

@Assurance_Case_Predicate

Sample predicate-based rules for elements and decorators used in an assurance case

@End_Assurance_Case_Predicate

@Assurance_Case_Pattern_Predicate

Sample predicate-based rules to support assurance case pattern

@End_Assurance_Case_Patten_Predicate

@Structural_Predicate

Sample predicate-based rules to support the structure of assurance case and assurance case patterns

@End_Structural_Predicate

@Domain_Information

Sample domain information of the given system for which an assurance case pattern is to be detected.

@End_Domain_Information

Experiment Set-up

Prompt Structure

- User Prompt

I need a comparative analysis of an assurance case and an assurance case pattern. This involves assessing their similarity using the established metrics: BLEU score and Semantic similarity. Apply the following measure-driven rule to determine if the assurance case pattern has been detected within the assurance case:

- If the BLEU score is superior or equal to X AND the semantic similarity score is superior or equal to X, conclude that the pattern has been detected in the assurance case.

- Otherwise, conclude that the pattern has not been detected in the assurance case.

@Assurance_Case_Pattern

Formalized Assurance Case Pattern

@End_Assurance_Case_Pattern

@Assurance_Case

Formalized Assurance Case

@End_Assurance_Case

Experiment Set-up

Pattern Detection Metric Rule

> If the value of *metric_1* is superior or equal to *threshold_metric_1*, **AND** if the value of *metric_2* is superior or equal to *threshold_metric_2*, . . . , **AND** if the value of *metric_n* is superior or equal to *threshold_metric_n*, then conclude that the formalized assurance case pattern has been detected in the formalized assurance case. Otherwise, conclude that the formalized assurance case pattern has not been detected in the formalized assurance case.

Metrics used in Pattern Detection Rule

- BLEU Score
- Cosine Similarity

Metric Thresholds

- 0.2 | 0.4 | 0.6 | 0.8 | 1.0

Evaluation Metrics

- **Precision**
  - Number of patterns correctly detected by SmartGSN over the total number of patterns detected by SmartGSN.

- **Recall**
  - Number of patterns correctly detected by SmartGSN over the total number of patterns manually used to create that assurance case.

- **F-Measure**
  - The harmonic mean of the precision and the recall.
  - F-Measure : (2 × Precision × Recall) / (Precision + Recall).

RQ1 Results

Can SmartGSN correctly detect assurance case patterns in assurance cases?

RQ1 Results

Recall (R), Precision (P), and F-Measure (FM) Result

System	Model	Metric Threshold														
		0.2			0.4			0.6			0.8			1		
		R	P	FM	R	P	FM	R	P	FM	R	P	FM	R	P	FM
ACAS XU	GPT-4o	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
BLUEROV2	GPT-4o	0.5	1	0.67	0.5	1	0.67	0.5	1	0.67	0	0	0	0	0	0
	GPT-4 Turbo	0.5	1	0.67	0.5	1	0.67	0.5	1	0.67	0	0	0	0	0	0
GPCA	GPT-4o	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
IM Software	GPT-4o	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
DeepMind	GPT-4o	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

RQ2 Results

How does the choice of metric thresholds impact the ability of SmartGSN to detect assurance case patterns?

RQ2 Results

Recall (R), Precision (P), and F-Measure (FM) Result

System	Model	Metric Threshold														
		0.2			0.4			0.6			0.8			1		
		R	P	FM	R	P	FM	R	P	FM	R	P	FM	R	P	FM
ACAS XU	GPT-4o	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
BLUEROV2	GPT-4o	0.5	1	0.67	0.5	1	0.67	0.5	1	0.67	0	0	0	0	0	0
	GPT-4 Turbo	0.5	1	0.67	0.5	1	0.67	0.5	1	0.67	0	0	0	0	0	0
GPCA	GPT-4o	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
IM Software	GPT-4o	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
DeepMind	GPT-4o	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	GPT-4 Turbo	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

Key Takeaways

At a threshold of 0.2, SmartGSN achieve perfect metric scores except for BLUEROV2.

At thresholds of 0.8 and 1.0, SmartGSN did not detect patterns at these higher thresholds.

Lower thresholds (e.g., 0.2) enable better pattern detection. Optimal threshold range is likely [0.2, 0.6].

SmartGSN performs lower under BLUEROV2 because its assurance case features multiple patterns. Currently, SmartGSN can detect only one pattern at a time.

Conclusion

- Introduced SmartGSN, a novel tool utilizing LLMs for semi-automatic management of assurance cases.
- Evaluated the pattern detection feature of SmartGSN.
- Future work will enhance pattern detection using advanced rules and pattern-matching algorithms.
- Plans to support assurance case refactoring and enable collaborative editing features.

Q&A

Any Questions?

THANK YOU FOR YOUR TIME