Normative Requirements Operationalization with Large Language Models

Nick Feng, **Lina Marsso**,
Sinem Getir Yaman, Isobel Standen, Yesugen Baatartogtokh, Reem Ayad,
Victória Oldemburgo de Mello, Bev Townsend, Hanne Bartels,
Ana Cavalcanti, Radu Calinescu, and Marsha Chechik

RE 2024
June 28, 2024

Context

Systems increasingly interacting with humans in various domains

(transport, environment, health and social care)

ALMI: Assistive-care robotics

Helps with food preparation, dressing, fallen-user alert, etc.

Detect the user has fallen

Alert that the user has fallen

ALMI robot from RoboStar (University of York, UK)

Normative Requirements

- Capture **s**ocial, **l**egal, **e**thical, **e**mpathetic, **c**ultural (**SLEEC**) aspects of systems
  e.g., When another person is in the room, avoid mentioning medications that the patient is on.

- Normative requirements encompass both functional and non-functional aspects and are delineated with respect to time

- Specified by stakeholders with non-technical expertise
  - Lawyer, regulators, ethicists, etc.

- Hard to get right
  - Stakeholders from different fields, different vocabularies
  - Their views are often conflicting or redundant
  - Stakeholders might not have sufficient technical background to reason about requirements
  - Requirements are complex: Allow constraints over data and time

Normative requirement well-formedness verification

Automated techniques based on first-order logic [ASE2024,ICSE2024] and process algebra [GYBJCC23] have been developed to check the well-formedness of normative requirements:

- Situational and vacuous conflicts
- Restrictive or insufficient requirements
- Unnecessary redundant requirements

[GYBJCC23] S. Getir-Yaman, C. Burholt, M. Jones, R. Calinescu, and A. Cavalcanti. "Specification and Validation of Normative Rules for Autonomous Agents", FASE 2023.
[ASE2024] N. Feng, L. Marsso, S. Yaman, B. Townsend, Y. Baatartogtokh, A. Cavalcanti, R. Calinescu, and M. Chechik. “Towards a Formal Framework for Normative Requirements Elicitation”, ASE 2024.
[ICSE2024] N. Feng, L. Marsso, S. Yaman, B. Townsend, Y. Baatartogtokh, R. Ayad, V. Mello, I. Standen, I. Stefanakos, C. Imrie, G. Rodrigues, A. Cavalcanti, R. Calinescu, and M. Chechik.. "*Analyzing and Debugging Normative Requirements via Satisfiability Checking*", ICSE2024 -.

SLEEC DSL [GYBJCC23]

Definitions

definition block

event DressingStarted
event CurtainOpenRqst
event CurtainsOpened

measure userUnderdressed: **Bool**
measure roomTemperature: **numeric**

Events: atomic, instantaneous actions

Measures: readable properties, types: *Boolean, numeric*

Rules

rule block

Rule1 **when** CurtainOpenRequest **then** CurtainsOpened **within** 30 **seconds**

unless userUnderDressed **then** RefuseRequest **within** 30 **seconds**

[GYBJCC23] S. Getir-Yaman, C. Burholt, M. Jones, R. Calinescu, and A. Cavalcanti. "Specification and Validation of Normative Rules for Autonomous Agents", FASE 2023.

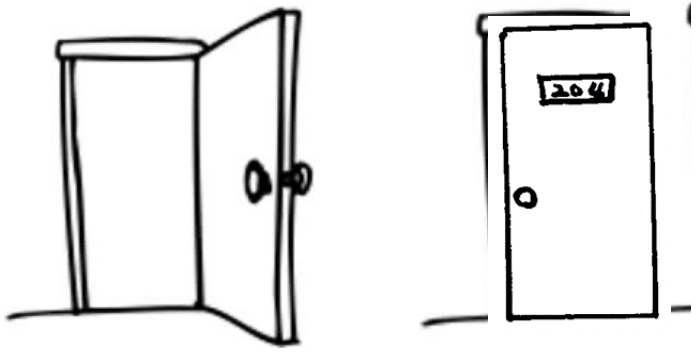
Problem

- Current analysis methods assume that the system capabilities in the SLEEC requirements are independent.

- Relations often assumed by non-technical stakeholders, e.g., based on common sense: If a patient is not allowed to drink liquid, it is also not allowed to drink juice!

Our goal

*Capturing the **semantic relationship** between the abstract representations of system capabilities in the SLEEC requirements.*



Capturing Semantic Relations

Use LLM to identify semantic relations between abstract representation of system capabilities (i.e., help with domain modeling)

- ❑ Capture relations between events, measures, and events and measures
- ❑ Restrict ourselves to binary relations
- ❑ Select a few relation types to focus on, in collaboration with a philosopher with expertise in common-sense knowledge

Relations between events: examples

Relation	SLEEC DSL example
hypernym	*water is an instance of liquid* `DrinkWater` **hypernym** `DrinkLiquid`
equal	patient and client are interchangeable, when patients are the only clients `CallPatient` **equal** `CallClients`
contradictory	*impossible to open and close the door simultaneously* `OpeningDoor` **isContradictoryWith** `ClosingDoor`
happen before	*locking the door occurs after closing it* `ClosingDoor` **happensBefore** `LockingDoor`

Relations between measures: examples

Notation	SLEEC DSL example
imply	*door opened can not be locked* `doorOpened` **imply** `not doorLocked`
mutual exclusive	*door opened can not be locked* `doorOpened` **mutuallyExclusive** `doorLocked`
opposite	*the door can be either opened or closed* `doorOpened` **oppositeTo** `doorClosed`
equal	*patient and client are interchangeable, when patients are the only clients* `patientConsented` **equal** `userConsented`

Relations between events and measures: examples

ID	SLEEC DSL example
induce	`CollectConsent induces consentObtained`
forbidden	`inWater forbids CarStartSpeeding`
until	`when CollectConsent then consentObtained until ConsentWithdraw`
for	`when LoginConfirmed then loggedIn for 10 minutes`

Sanitizing SLEEC definitions

Definitions *definition block*

event DressingStarted
event CurtainOpenRqst
event CurtainsOpened

measure userUnderdressed: **Bool**
measure roomTemperature: **numeric**

Rules *rule block*

Rule1 **when** CurtainOpenRequest **then** CurtainsOpened **within** 30 **seconds**

unless userUnderDressed **then** RefuseRequest **within** 30 **seconds**

Sanitizing SLEEC definitions

Enrich the preliminary set of SLEEC requirements with captured semantic relations

1. Use our LLM technique to extract the semantic relation candidates
2. Automatically filter semantic relation candidates to ensure consistency
3. (Stakeholders) Manually review the remaining candidates, add new ones
4. Integrate validated candidates into preliminary SLEEC DSL requirements

Sanitizing SLEEC definitions

Enrich the preliminary set of SLEEC requirements with captured semantic relations

1. **Use our LLM technique to extract the semantic relation candidates**
2. Automatically filter semantic relation candidates to ensure consistency
3. (Stakeholders) Manually review the remaining candidates, add new ones
4. Integrate validated candidates into preliminary SLEEC DSL requirements

Excerpt of prompt for identifying hypernym binary relations

Here are my definitions:

def_start:

event PreparingDeployment
....
measure userOccupied: Boolean

e.g., ALMI definition

def_end

For every pair of events A and B, can you please answer the following questions: Does occurrence of event A implies the occurrence of event B at exactly the same time, if yes, please say A **hypernym** B, if no, please say A NOT hypernym B. For example, if A is a special type of B, then occurrence of A implies the occurrence of B (DrinkingWater **hypernym** Drinking).
...

Sanitizing SLEEC definitions

Enrich the preliminary set of SLEEC requirements with captured semantic relations

1. Use our LLM technique to extract the semantic relation candidates (*Rel*)
2. **Automatically filter semantic relation candidates to ensure consistency**
3. (Stakeholders) Manually review the remaining candidates, add new ones
4. Integrate validated candidates into preliminary SLEEC DSL requirements

Automatic filtering

A. **Identify relations that could lead to logical inconsistencies** (based on inference rules)

B. **Filter the identified relations**

A. Inconsistent semantic relation identification

Identify relations extracted (*Rel*) that could lead to logical inconsistencies based on inference rules

- Use inference rules (Horn clauses) to derive new relations from *Rel*
- Obtain *Rel**: fixpoint set of all relations that are in *Rel* or derivable from *Rel*
- A relation *r* leads to an inconsistency if *Rel** contains both the positive relation *r* and the negative relation not ***r***
- GPT is then tasked to judge the inferred relationship.
- Filter the smallest set of relation from *Rel* to resolve the inconsistency

Horn inference rules for semantic relations

Inference rules for relations between two events.

$$\frac{e_a \textbf{ isContradictoryWith } e_b}{\textbf{not } (e_a \textbf{ hypernym } e_b)}\ (IP1^-) \quad \frac{e_a \textbf{ happenBefore } e_b}{\textbf{not } (e_a \textbf{ hypernym } e_b)}\ (IP2^-) \quad \frac{e_a \textbf{ hypernym } e_b \wedge e_b \textbf{ hypernym } e_c}{e_a \textbf{ hypernym } e_c}\ (IPtrans^+) \quad \frac{e_a \textbf{ equal } e_b}{e_a \textbf{ hypernym } e_b}\ (IPEQ^+)$$

$$\frac{e_a \textbf{ hypernym } e_b}{\textbf{not } (e_a \textbf{ isContradictoryWith } e_b)}\ (ME1^-) \quad \frac{e_b \textbf{ isContradictoryWith } e_a}{e_a \textbf{ isContradictoryWith } e_b}\ (MEcomm^+) \quad \frac{e_a \textbf{ hypernym } e_b \wedge e_b \textbf{ isContradictoryWith } e_c}{e_a \textbf{ isContradictoryWith } e_c}\ (METrans^+)$$

$$\frac{e_a \textbf{ hypernym } e_b \wedge e_b \textbf{ hypernym } e_a}{e_a \textbf{ equal } e_b}\ (EQIP^+) \quad \frac{e_a \textbf{ equal } e_b}{e_b \textbf{ equal } e_a}\ (EQcom^+)$$

$$\frac{e_a \textbf{ happenBefore } e_b \wedge e_b \textbf{ happenBefore } e_c}{e_a \textbf{ happenBefore } e_c}\ (HBtrans1^+) \quad \frac{e_a \textbf{ happenBefore } e_b \wedge e_c \textbf{ hypernym } e_b}{e_a \textbf{ happenBefore } e_c}\ (HBtrans2^+)$$

$$\frac{e_b \textbf{ hypernym } e_a \wedge e_b \textbf{ happenBefore } e_c}{e_a \textbf{ happenBefore } e_c}\ (HBtrans3^+)$$

Inference rules for relations between two propostions over measures.

$$\frac{p_a \textbf{ mutualExc } p_b}{\textbf{not } (p_a \textbf{ imply } p_b)}\ (MIP1^-) \quad \frac{p_a \textbf{ imply } p_b \wedge p_b \textbf{ imply } e_c}{p_a \textbf{ imply } p_c}\ (MIPtrans^+) \quad \frac{p_a \textbf{ equal } p_b}{p_a \textbf{ imply } p_b}\ (IPEQ1^+) \quad \frac{p_a \textbf{ equal } p_b}{\neg p_a \textbf{ imply } \neg p_b}\ (IPEQ2^+)$$

$$\frac{p_a \textbf{ imply } p_b}{\textbf{not } (p_a \textbf{ mutualExc } p_b)}\ (MME1^+) \quad \frac{p_b \textbf{ mutualExc } p_a}{p_a \textbf{ mutualExc } p_b}\ (MMEcomm^+) \quad \frac{p_a \textbf{ imply } p_b \wedge p_b \textbf{ mutualExc } p_c}{p_a \textbf{ mutualExc } p_c}\ (MMEtrans^+)$$

$$\frac{p_a \textbf{ imply } p_b \wedge p_b \textbf{ imply } p_a}{p_a \textbf{ equal } p_b}\ (MEQIP^+) \quad \frac{p_a \textbf{ opposite } \neg p_b \wedge p_b \textbf{ imply } p_a}{p_a \textbf{ equal } p_b}\ (MEQOP^+) \quad \frac{p_a \textbf{ equal } p_b}{p_b \textbf{ equal } p_a}\ (MEQcom^+)$$

$$\frac{p_a \textbf{ equal } \neg p_b}{p_a \textbf{ opposite } p_c}\ (MOPEQ^+) \quad \frac{p_b \textbf{ opposite } p_s}{p_a \textbf{ opposite } p_b}\ (MOPcoms^+) \quad \frac{p_a \textbf{ mutualExc } p_b \wedge \neg p_a \textbf{ mutualExc } \neg p_b}{p_a \textbf{ opposite } p_b}\ (MOPME^+)$$

`+' : rules deriving positive relations, used to derive new relations
`-': rules deriving negative relations, used for identifying inconsistency witnesses

Rule derived example

Rel = { r1= e1 **hypernym** e3,
r2 = e3 **hypernym** e2 }

$$\frac{e_a \textbf{ hypernym } e_b \wedge e_b \textbf{ hypernym } e_c}{e_a \textbf{ hypernym } e_c} \; (IPtrans^{+})$$

From r1 and r2 by rule $(IPtrans^{+})$ we derive r3:
r3 = e1 **hypernym** e2

We ask LLM if r3 holds:

- if yes we add in Rel
- otherwise we add r3 = **not** (e1 **hypernym** e2)

Inconsistent semantic relation example

Rel = {r1= e1 **hypernym** e3,
r2 = e3 **hypernym** e2 ,
r3 = **not** (e1 **hypernym** e2)}

$$\frac{e_a \textbf{ hypernym } e_b \wedge e_b \textbf{ hypernym } e_c}{e_a \textbf{ hypernym } e_c} \; (IPtrans^{+})$$

From r1 and r2 by rule $(IPtrans^{+})$ we derive r5:
r4 = e1 **hypernym** e2

The derived relation r4 is inconsistent with r3 (conflicts)

B. Automatic filtering

A. Identify relations that could lead to logical inconsistencies based on inference rules

B. Filter the identified relations

a) In case the judgement from GPT is inconsistent with the inference result, we prune the relationship that causes inconsistency in *Rel*.

b) Filtering until a fixpoint

Sanitizing SLEEC definitions

Enrich the preliminary set of SLEEC requirements with captured semantic relations

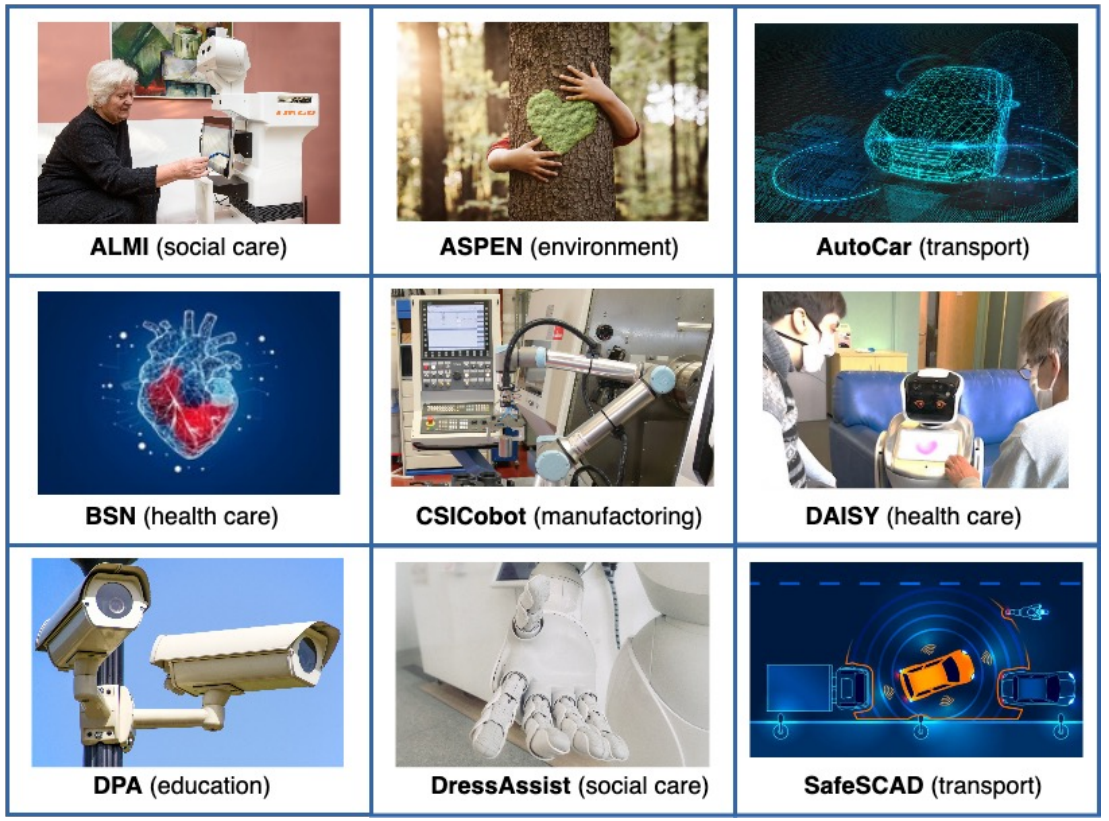
1. Use our LLM technique to extract the semantic relation candidates
2. Automatically filter semantic relation candidates to ensure consistency
3. **(Stakeholders) Manually review the remaining candidates, add new ones**
4. **Integrate validated candidates into preliminary SLEEC DSL requirements**

Evaluation

We first would like to study the effectiveness of identifying semantic relations in improving requirements analysis.

RESERVE:
repository of 9 real-world case studies [ICSE2024]

- **Domains:** transport, environment, manufacturing. health and social care.
- **Different stages:** ranging from the design phase to deployed systems
- **Non-technical stakeholders:** an ethicist, a lawyer, a philosopher, and a psychologist
- **Technical stakeholders:** a safety analyst, and 3 engineers
- **Normative requirements**: 233 N-NFRs in total which are **well-formed-free**



[ICSE2024] N. Feng, L. Marsso, S. Yaman, B. Townsend, Y. Baatartogtokh, R. Ayad, V. Mello, I. Standen, I. Stefanakos, C. Imrie, G. Rodrigues, A. Cavalcanti, R. Calinescu, and M. Chechik.. "*Analyzing and Debugging Normative Requirements via Satisfiability Checking*", ICSE2024 -.

Effectiveness of adding LLM-discovered relationships w.r.t. the number of relevant and spurious WFIs identified

- **103** semantic-relations, **53** of them classified as relevant by N-TS

- Considering this 53 relations enabled us to find **13 new WFIs** (including **6 conflicts**)

case studies	rules (evnt.– msr.)	relations (TP - FP)	new WFI v-conf.	s-conf.	redund.	insuffi.	restrict.
ALMI	39 (41 – 15)	5 - 2	0	0	0	0	0
ASPEN	15 (25 – 18)	19 - 8	1	0	1	0	0
AutoCar	19 (36-26)	6 - 19	0	0	0	0	0
BSN	29 (33-31)	1 - 3	0	0	2	0	0
CSI-Cobot	20 (23-11)	3 - 3	0	0	0	0	0
DAISY	26 (45-31)	15 - 5	5	0	4	0	0
DPA	26 (28-25)	3 - 2	0	0	0	0	0
DressAssist	31 (54-42)	0 - 0	0	0	0	0	0
SafeSCAD	28 (29-20)	1 - 8	0	0	0	0	0

ALMI relation added examples:

kitchenSafe **implies** (riskLevel = low)
alarmOn **mutualExclusive** alarmRestarts

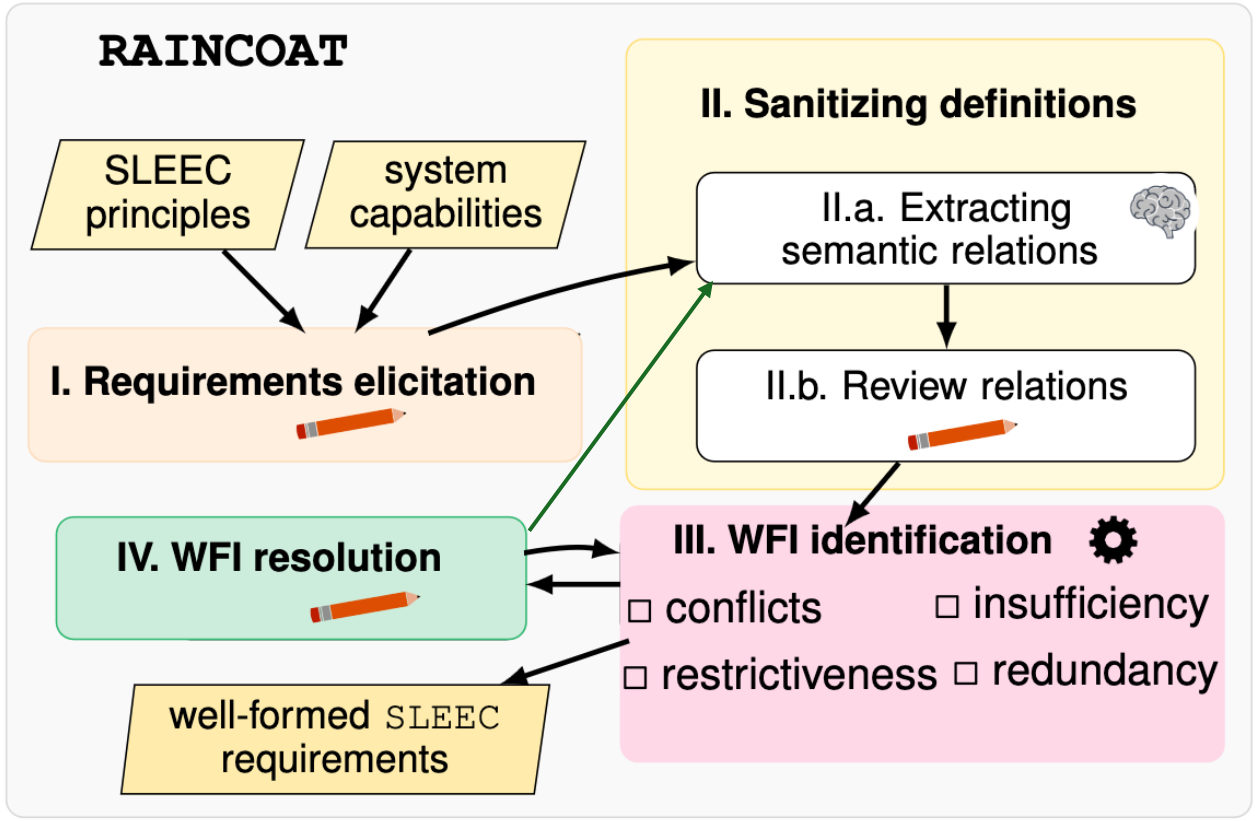
We integrated the definition sanitization step into the normative requirement iterative elicitation process [ASE2024] *to preliminary study:*

a) Facilitate the elicitation of a comprehensive set of requirements: encompass
   - **all existing system capabilities**
   - **social, legal, ethical, empathetic, cultural considerations**

b) Impact on supporting non-technical stakeholders in achieving a **well-formed set of requirements**.

[ASE2024] N. Feng, L. Marsso, S. Yaman, B. Townsend, Y. Baatartogtokh, A. Cavalcanti, R. Calinescu, and M. Chechik. “Towards a Formal Framework for Normative Requirements Elicitation”, ASE 2024.

Overview of the new approach



How well does RAINCOAT support non-technical stake-holders in guiding the elicitation and analysis process?

i. **adhoc elicitation**:
Manual elicitation without guidelines
(to provide baseline)

ii. **systematic-elicitation:**
Following RAINCOAT stage 1 guidance
(impact of structured approach to requirements elicitation)

iii. **RAINCOAT-elicitation-validation:**
Following the overall RAINCOAT approach
(impact of structured approach to requirements elicitation)

How well does RAINCOAT support non-technical stake-holders in guiding the elicitation and analysis process?

experiments	cases	rules (evnt.– msr.)	relations (TP - FP)	WFI					Number of iterations				
				v-conf.	s-conf.	redund.	insuffi.	restrict.	v-conf.	s-conf.	insuffi.	restrict.	redund.
adhoc-elicitation	Tabiat	19 (27 – 13)	13 - 20	0	0	0	not applicable		not applicable				
systematic-elicitation	Casper	57 (59 – 24)	11 - 18	22	2	12	not applicable		not applicable				
RAINCOAT-	Tabiat	28 (37 – 18)	17 - 21	0	0	0	0	0	1	3	3	0	0
elicitation-validation	Casper	26 (38 – 14)	23 - 22	0	0	0	0	0	0	0	1	0	1

- Systematic vs adhoc : cover more capabilities, but causes more WFIs
- Systematic vs RAINCOAT: * more rules, but 12 redundant ones!
  * confidence of having elicited enough (insufficiency and restrictiveness)

Summary

Goal: support non-technical stakeholders in eliciting well-formed normative requirements

Our contributions:

Leverage LLMs to capture common sense to bridge gap between manually and automatically analyzing normative requirements

- **Extracting semantic relations** between the abstract representations **of system capabilities** in the requirements
- **Enrich automated reasoning techniques** for **eliciting and analyzing** the consistency and coherence of the requirements

Outcome: An effective use of LLM for bridging the gap between formal reasoning tool and non-technical stakeholders!

Future research directions (SLEEC)

a) How to best support non-technical stakeholders in validating SLEEC requirements?
b) How to capture user preferences and uncertainty in SLEEC requirements?
c) How to handle system adaptation, environment adaptation, SLEEC requirements evolution?
d) How to assure that systems satisfy their SLEEC constraints:
e) How to elicit, analyze, and monitor SLEEC requirements with respect to uncertainty?
f) How to handle systems with a less clear interface (e.g., LLM-based)?

Thank you

The iimplementation and all our evaluation artifacts available in:
https://github.com/NickF0211/sleecvalDef