

TOHOKU UNIVERSITY

TESTING TRUSTWORTHY IN-CONTEXT LEARNING WITH MMT4NL



IN-CONTEXT LEARNING

the ability for LLMs to **adapt their responses** based on the context provided in the input without requiring explicit retraining.

They are **highly sensitive to minor input variations**.

- Slight perturbations; small character or word-level changes may lead to significant performance degradation.
 - Real world prompts are by no means "**Perfect**"; sometimes they contain mistakes.
- Minor (unintended) changes may alter the model outputs in unpredictable ways.
 - Justifies the need for model evaluation... but how?

“TRADITIONAL METHODS”

Imagine being tasked with testing a function that adds 2 numbers together \Rightarrow **Easy!** There can be only one definitive answer. (e.g. $1+3 = 4$, $5+6 = 11$)

- Not applicable to LLM Tasks – Summarization, Question Answering, and Sentiment Analysis
- There isn't a single list of a “universally agreed” correct answer – test oracle problem

METAMORPHIC TESTING

An alternative approach to address this problem is to apply a software design principle known **as "Metamorphic testing"**. Key questions change from:

Is this output correct?

to

Does the change in the output makes sense, given the changes in the input? Does the **consistency** of output changes hold, for a **controlled set of input changes**?

MMT4NL FRAMEWORK




- a software testing-inspired framework designed for evaluating the trustworthiness of LLMs, particularly their ICL capabilities.
- It aims to verify relationship between outputs that **should hold** when inputs are modified.

METAMORPHIC RELATION

$$MR(x, f(x), P(x), P(f(x))) \implies g(P(x)) = P(f(x))$$

where:

- $x \Rightarrow$ the original input
- $f(x) \Rightarrow$ the transformed input, f is the function that perform the input transformations.
- $P(x) \Rightarrow$ the output of the original input
- $P(f(x)) \Rightarrow$ the output of the transformed input

A **Metamorphic Relation** implies that: there exists a relation g that specifies the expected relationship between the output of the transformed input $P(f(x))$ and the output of the original input $P(x)$ 

TYPE OF PERTUBATIONS

- **Taxonomy:** Testing consistency when a word is replaced by a synonym word. The expected output relation is typically that the output should remain the same ($P(x) = P(f(x))$)

"I am so tired" \Leftrightarrow "I'm so exhausted"

- **NER (Named Entity Recognition):** Replacing "pronouns" with fictitious "proper nouns," expecting the output to remain unchanged ($P(x) = P(f(x))$)

"I am so tired" \Leftrightarrow "Jane is so tired"

- **Negation Handling:** This transforms the input by adding negation cues to ideally reverse or appropriately adjust the original sentiment or meaning. The expected output relation is approximately the negation of the original output ($\neg P(x) \approx P(f(x))$)

"I am so tired" \Leftrightarrow "I am so not tired"

- **Vocab (Vocabulary-based tests):** These check robustness when the input contains new or potentially unknown words. The expected relationship is that the model should either ignore the new word or make an inference from the context ($g(P(x)) \approx P(f(x))$)

"I am so tired" \Leftrightarrow "I am so really tired"

- **Fairness:** This involves changing demographic attributes in the input, such as gender or race. If these demographic changes should not affect the task, the expected output should remain the same ($P(x) = P(f(x))$)

"I am so tired" \Leftrightarrow "She is so tired"

- **Temporal:** This checks consistency when the input has time-based information that is modified. The expected output should remain consistent ($P(f(x)) = P(x)$)

"I am so tired" \Leftrightarrow "Not sure how it was like before but now I'm so tired"

- **SRL (Semantic Role Labeling):** This tests consistency when the input is rephrased while preserving meaning, ensuring predicates and semantic roles remain unchanged. The expected output is the same as the original output ($P(f(x)) = P(x)$)

"would it be common to find a penguin in Miami?" \Leftrightarrow "Is a penguin in Miami commonly found?"

- **Coreference:** It involves restructuring questions to include explicit pronoun references, which can create referential distance.

"would it be common to find a penguin in Miami?" \Leftrightarrow "Considering penguins, would it be common to find one in Miami?"

TOHOKU UNIVERSITY

ACCEPTABILITY OF ARGUMENTS & LOGIC PROGRAMMING



PURPOSE



- Study the fundamental mechanism used in **human argumentation**.
- Explore ways to implement this mechanism onto computers and systems.

This paper introduces an abstract theory that model human's argumentative reasoning with the central notion being "Acceptability of Arguments" based on the **Argumentation Framework**.

ARGUMENTATION FRAMEWORK (AF)

A map of debates, highlighting the attacks that occurs between arguments.

To get more technical it is officially defined as a pair of

- AR : A set of arguments.
- $attacks$: A binary relation on AR . $\Rightarrow attacks \subseteq AR \times AR$.

$$AF = \langle AR, attacks \rangle$$

It is essentially a directed graph, with a list of arguments as "nodes", and an attack from A to B is a directed edge from $A \Rightarrow B$.

EXAMPLE

Example 1 (Mock Argument):

- i_1 : I's first argument (A's gov't doesn't recognize I's).
- a : A's counterargument (I's gov't doesn't recognize A's).
- i_2 : I's second argument (A's gov't is terrorist, justifying non-recognition by I).

$$AR = i_1, a, i_2$$

$attacks = (i_1, a), (a, i_1), (i_2, a)$ (since i_2 undermines A's justification for a , it effectively attacks a).

THE NOTION OF "ACCEPTABILITY"

- AF is a basic structure for representing argument and their attack relationships.
- How can we determine which arguments within this framework are **ultimately acceptable**?
- This paper starts by defining foundational properties a collection of arguments must possess such as being "**conflict-free**" and "**admissible**"
- These serve as components for various acceptability semantics discussed later.

CONFLICT-FREE SETS

- A conflict-free set (Internal Coherence) \leftrightarrow A set $S \subseteq AR$ is conflict-free if there are no arguments A, B in S such that $attacks(A, B)$. In other words, it is a collection of statements or propositions that do not contradict each other.

ADMISSIBLE SETS

- **Admissible Sets (Robustness through self-defense)** \leftrightarrow A conflict-free set of arguments S is admissible if every argument in S is acceptable with respect to S .
 - "An argument $A \in AR$ is acceptable with respect to a set S of arguments **iff** for each argument $B \in AR$: if B attacks A then B is attacked by S ."
 - "A conflict-free set of arguments S is admissible iff each argument in S is acceptable with respect to S ."
 - This means S defends all its own members against any external attacks, using only arguments from within S . To put it simply, for each member A in conflict-free set S that is being attacked externally by B , there must be at least one argument in S that attacks B .

ACCEPTABILITY SEMANTICS



- This paper later it introduces various semantics – specific criteria or methods for selecting which sets of arguments are considered "justified" or "collectively acceptable"
- Preferred Extension (Open-Minded)
- Stable Extension (Assertive / Dominating / Opinionated)
- Grounded Extension (Skeptical / Cautious / Solid Foundation / Undisputed Core)
- Complete Extension (Self-Contained / Fully Reasoned)

PREFERRED EXTENSION



- The largest possible set of admissible arguments.
- The largest collection of arguments that is **conflict-free** and can defend against any external attacks.
- You can't add any more arguments to it and still keep it admissible.

STABLE EXTENSION



- A stable extension is a conflict-free set
- It must attack every single argument that is not in the set.
- Every stable extension is also a preferred extension, hence admissible

GROUNDING EXTENSION



- The set of arguments that are considered absolutely foundational, super safe and **uncontroversial**. (not attacked at all, or defended by other arguments that are themselves “solid ground” – arguments that are not attacked.)
- Officially defined as the least fixed point of the characteristic function (F_{AF}). And there can be **only one for** each framework.
- We can use the Characteristic Function to compute the Grounded Extension

GROUNDING EXTENSION



- **Characteristics Function:** The characteristic function $F_{AF}(S)$ calculates a *new* set of arguments. This new set consists of every single argument (from the entire debate) that can be successfully defended if the arguments in your set S are the ones doing the defending.
 - A "**fixed point**" of F_{AF} is a set of arguments X such that if you put X into the Characteristic Function, you get X back out (i.e., $F_{AF}(X) = X$). It's a stable set where the arguments defended by the set are **precisely** the arguments *in* the set. It means that X defends its own members (and only X)
- We can calculate the grounded extension using the Characteristic Function

GROUNDING EXTENSION

- We can try to find out the Grounded Extension by iterating over the Characteristic Function, starting at $S_0 = \emptyset$
- $S_1 = F_{AF}(S_0)$: The first iteration is gonna return as all the arguments that are not attacked at all (because they don't need any defense) Then we use this as the ground for "acceptable" arguments.
- $S_2 = F_{AF}(S_1)$: We feed the ground of unattacked arguments into the Characteristic Function, to uncover the arguments that can be defended by S
- Repeat: $S_{i+1} = F_{AF}(S_i)$: and keep feeding the results of the previous iterations. The Grounded Extension (GE_{AF}) is the set you reach when $S_i + 1 = S_i$ (the fixed point—according to the definition above)—that is, when applying F_{AF} doesn't add any new arguments.

COMPLETE EXTENSION

- **Complete Extension** \leftrightarrow similar to the Grounded Extension, but instead of building S from the ground up, S can be any set that satisfy the following conditions.
 - It is **conflict-free**
 - It is a "**fixed point**" of F_{AF} ($F_{AF}(S) = S$); self-contained and only defend itself
- There can be more than one Complete Extensions



THANK YOU

F O R T H E A T T E N T I O N

