ToMi Data Issues Analysis

The issues in the ToMi dataset are categorized in **structural** (**incorrect ground truth**), **semantic** (**incorrect story type**) and **pragmatic flaws** (**incorrect agent number**). The data and the metadata are synthetically generated, and to understand the problems in them we analyzed the code in the original repository. We found out problems with incorrect story type and incorrect agent number, but it is possible that these are related to the ground truth issue too.

Incorrect Story Type

The following is an example of incorrect story type, where the code is set as type *true_belief*, but it should be *false_belief* instead:

1 Jayden entered the lounge.
2 Isla entered the lounge.
3 The cherry is in the pantry.
4 Isla exited the lounge.
5 Isla entered the hall.
6 Jayden moved the cherry to the treasure_chest.
Code generated story type: true_belief

(Isla exited before Jayden moved the object to a new location and she entered a different room, so the story type should be *false_belief*)

Diving deep into the original code, we noticed that the reason for these issues is a weak logic on how the story type is assigned.
Specifically, for every story a list of actions is randomly generated (always containing one *move* action and up to two *loc_change* actions), and then the story is labeled as *false_belief* just if the *move* action is in the **second position** of the list.
The previously presented story is wrongly classified as *true_belief*, since *move* is in the third position, after the two *loc_change* (4 and 5).

Incorrect Agent Number

The code authors define as *agent 0* the agent moving the object from a location to another, and *agent 1* the remaining one. The problem is that in the generated metadata, many agents are assigned an incorrect number, as in the following example where Logan is incorrectly labeled as *agent 0* and James as *agent 1*.

1 Logan entered the hallway.
2 James entered the hallway.
3 The jeans is in the bucket.
4 William entered the hallway.
5 Logan exited the hallway.
6 James moved the jeans to the basket.

7 Where will Logan look for the jeans?
Code generated agent number for Logan: agent_0

(James moved the object to a new location, so, according to the official definition, it's James that should be labeled as *agent_0*)

We found that the code initially generates two agents and then shuffles them, just to have a random order in which they enter the room. The problem is that, later, the original order is the one used to generate the rest of the story, and the shuffled order is used to define the agent label in the metadata. This implies a mismatch between what the agents are doing in the story (based on original order) and their labels (based on the shuffled order).
As you can see in the story, Logan is the first character to enter the room and that's why the code assigned the agent number 0 to him, even if he is not the one moving the object.