

A Prompting

To develop an intuition of the LLM’s task, we show examples of prompts for each configuration used in the experiments of Section 5.1. In particular, we show examples of prompts for both option termination verification and option discovery, and both game screen and cropped observations. Finally, we present prompts where two options terminated (a subgoal is achieved): a key pick-up and a door unlock.

A.1 Cropped vs gamescreen observations

```


Prompt with cropped observations



The environment is MiniHack.



I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:



- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall, wall, top left corner (of a room), top right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.



The task of the agent is to win the game.



First, based on your knowledge of NetHack, break down the task of the agent into subgoals. Then, consider the following game transition, which might or might not contain these subgoals. Determine if any of the subgoals is achieved at Time: 1 or not.



Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:



```
python
{
 <name of goal>: <bool>,
}
```



Observation Sequence:



```
<gameplay>
Time: 0
Current message: Never mind.

 - - - -
 | . . |
 | . . |
 - @ < |
 |
 | . (. . . |
 - - - - -
```



Time: 1  
Current message:



```
 - - - -
 | . . |
 | @ . |
 - . < |
 |
 | . (. . . |
 - - - - -
```



```
</gameplay>
```



I will not consider anything that is not in the dictionary.  
You have only one shot at this, and you cannot ask for clarifications.


```

Prompt 2: Example of a prompt where transitions have cropped observations.

Prompt with gamescreen observations

The environment is MiniHack.

I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:

- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall, wall, top left corner (of a room), top right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.

The task of the agent is to win the game.

First, based on your knowledge of NetHack, break down the task of the agent into subgoals.
Then, consider the following game transition, which might or might not contain these subgoals.
Determine if any of the subgoals is achieved at Time: 1 or not.

Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:

```
```python
{
 <name of goal>: <bool>,
}
```

Observation Sequence:

```
<gameplay>
Time: 0
```

Never mind.

```

 - - - -
 | . . |
 | . . |
 - @ < |
 |
 | . (. . . |
 - - - - -
```

```
Agent the Footpad St:14 Dx:17 Co:17 In:9 Wi:11 Ch:7 Chaotic S:0
Dlvl:1 $:0 HP:12(12) Pw:2(2) AC:7 Xp:1/0
Time: 1
```

```

 - - - -
 | . . |
 | @ . |
 - . < |
 |
 | . (. . . |
 - - - - -
```

```
Agent the Footpad St:14 Dx:17 Co:17 In:9 Wi:11 Ch:7 Chaotic S:0
Dlvl:1 $:0 HP:12(12) Pw:2(2) AC:7 Xp:1/0
```

```
</gameplay>
```

I will not consider anything that is not in the dictionary.  
You have only one shot at this, and you cannot ask for clarifications.

Prompt 3: Example of a prompt where transitions have game screen observations.

## A.2 Option termination vs option discovery

### Prompt with cropped observations

```
The environment is MiniHack.

I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:
- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall,
wall, top left corner (of a room), op right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.

The task of the agent is to win the game.

Consider the following subgoals:
'''python
subgoals = {
 "pick up the key": None,
 "open the door": None,
}
'''

Then, consider the following game transition, which might or might not contain these subgoals.
Determine if any of the subgoals is achieved at Time: 1 or not.

Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:
'''python
{
 <name of goal>: <bool>,
}
'''

Observation Sequence:

<gameplay>
Time: 0
Current message: Never mind.

 - - - -
 | . . |
 | . . |
 - @ < |
 |
 | . (. . . |
 - - - - -

Time: 1
Current message:

 - - - -
 | . . |
 | @ . |
 - . < |
 |
 | . (. . . |
 - - - - -

</gameplay>

I will not consider anything that is not in the dictionary.
You have only one shot at this, and you cannot ask for clarifications.
```

Prompt 4: Example of a prompt for instruction verification. Here, goals are provided externally from a human.

## Prompt with cropped observations

The environment is MiniHack.

I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:

- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall, wall, top left corner (of a room), top right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.

The task of the agent is to win the game.

First, based on your knowledge of NetHack, break down the task of the agent into subgoals. Then, consider the following game transition, which might or might not contain these subgoals. Determine if any of the subgoals is achieved at Time: 1 or not.

Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:

```
python
{
 <name of goal>: <bool>,
}
```

Observation Sequence:

```
<gameplay>
Time: 0
Current message: Never mind.
```

```
 - - - -
 | . . |
 | . . |
 - @ < |
 |
 | . (. . . |
 - - - - -
```

```
Time: 1
Current message:
```

```
 - - - -
 | . . |
 | @ . |
 - . < |
 |
 | . (. . . |
 - - - - -
```

```
</gameplay>
```

I will not consider anything that is not in the dictionary.  
You have only one shot at this, and you cannot ask for clarifications.

Prompt 5: Example of a prompt for options discovery and termination verification.

### A.3 Examples of different subgoals

## Examples of key pickup

The environment is MiniHack.

I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:

- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall, wall, top left corner (of a room), top right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.

The task of the agent is to win the game.

First, based on your knowledge of NetHack, break down the task of the agent into subgoals.

Then, consider the following game transition, which might or might not contain these subgoals.

Determine if any of the subgoals is achieved at Time: 1 or not.

Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:

```
python
```

```
{
 <name of goal>: <bool>,
}
```

Observation Sequence:

```
<gameplay>
```

```
Time: 0
```

```
Current message: It's a wall.
```

```
 | . . |
 | . . |
 - - + - . < |
 | |
 | . @ . . . |
 - - - - - - -
```

```
Time: 1
```

```
Current message: g - a key named The Master Key of Thievery.
```

```
 | . . |
 | . . |
 - - + - . < |
 | |
 | . @ . . . |
 - - - - - - -
```

```
</gameplay>
```

I will not consider anything that is not in the dictionary.

You have only one shot at this, and you cannot ask for clarifications.

Prompt 6: Example of a prompt where the transition shows a successful key pickup subgoal.

## Examples of door unlock

The environment is MiniHack.

I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:

- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall, wall, top left corner (of a room), top right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.

The task of the agent is to win the game.

First, based on your knowledge of NetHack, break down the task of the agent into subgoals.

Then, consider the following game transition, which might or might not contain these subgoals.

Determine if any of the subgoals is achieved at Time: 1 or not.

Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:

```
python
{
 <name of goal>: <bool>,
}
```

Observation Sequence:

```
<gameplay>
Time: 0
Current message:
```

```
 - - - -
 | . . |
 | . . |
- + - . < |
| . . @ . . |
| |
- - - - -
```

```
Time: 1
Current message: Never mind.
```

```
 - - - -
 | > | . . |
 . . | . . |
- - | - . < |
| . . @ . . |
| |
- - - - -
```

```
</gameplay>
```

I will not consider anything that is not in the dictionary.  
You have only one shot at this, and you cannot ask for clarifications.

Prompt 7: Example of a prompt where the transition shows a successful door-unlock subgoal.

## B Responses

To give an intuition of how different models approach the task, here, we show a series of responses to Prompt 1. Notice that, to stimulate the model to reason about subgoals, we ask to consider its knowledge of Nethack, rather than MiniHack.

### c4ai-command-r-v01

```
'''python
{
 'find_item': False,
 'open_door': False,
 'collect_item': False,
 'move_upstairs': False,
 'move_downstairs': True,
 'kill_monster': False,
 'find_key': False,
 'open_lock': False,
 'light_lamp': False,
 'win_game': False
}
'''
```

Response 1: Example response

### gemma-1.1-7b-it

```
Subgoals:
- **Find a way out:** Locate and reach the exit of the map.
- **Gather useful items:** Collect any available items for future use.
- **Explore the environment:** Discover the layout of the map and potential hazards.

Subgoal Achieved at Time: 1

'''python
{
 "Find a way out": False,
 "Gather useful items": True,
 "Explore the environment": True,
}
'''
```

Response 2: Example response

### Mistral-7B-Instruct-v0.2

```
Based on my understanding of the given environment and the task, the subgoals for the agent are as follows:

1. Exploration: Reach all the unvisited parts of the map.
2. Collection: Gather all the useful items (pick-axe, key, lamp, etc.)
3. Door Opening: Open all the doors (locked or not) to progress through the map.

Now, let's analyze the given gameplay sequence and determine if any of the subgoals have been achieved at Time: 1.

'''python
{
 "Exploration": False,
 "Collection": False,
 "Door Opening": False
}
'''

At Time: 1, the agent has moved to a new location and has not interacted with any items or doors. Therefore, none of the subgoals have been achieved at this point.
```

Response 3: Example response

### Mixtral-8x7B-Instruct-v0.1

Based on my knowledge of NetHack, the task of the agent can be broken down into the following subgoals:

1. Movement: The agent needs to move through the game environment to reach the goal.
2. Collection: The agent needs to collect useful items that can help it in achieving the goal.
3. Door manipulation: The agent needs to open doors to explore new areas.
4. Staircase navigation: The agent needs to navigate staircases to move to different levels of the game.
5. Key collection: The agent needs to collect keys to open locked doors.
6. Winning the game: The agent needs to reach the final goal and win the game.

Based on the given game transition, the subgoals achieved at Time: 1 are:

```
```python
{
  "Movement": True,
  "Collection": False,
  "Door manipulation": False,
  "Staircase navigation": False,
  "Key collection": False,
  "Winning the game": False,
}
```
```

Explanation:

- \* Movement: The agent moved from the bottom left corner of the room to the bottom right corner of the room, so this subgoal is achieved.
- \* Collection: The agent did not collect any item in this transition, so this subgoal is not achieved.
- \* Door manipulation: The agent did not encounter any doors in this transition, so this subgoal is not achieved.
- \* Staircase navigation: The agent did not encounter any staircases in this transition, so this subgoal is not achieved.
- \* Key collection: The agent did not collect any keys in this transition, so this subgoal is not achieved.
- \* Winning the game: The agent did not reach the final goal in this transition, so this subgoal is not achieved.

### Response 4: Example response

### Llama-3-8B-Instruct

Based on my knowledge of NetHack, I break down the task of the agent into the following subgoals:

- \* Find a useful item (pick-axe, key, lamp...)
- \* Find a ladder or staircase up
- \* Find a ladder or staircase down
- \* Open a door (if it's closed)
- \* Unlock a door (if it's locked)
- \* Reach the goal (win the game)

Now, considering the given game transition, I determine if any of the subgoals has been achieved at Time: 1 or not:

```
{
 'Find a useful item': False,
 'Find a ladder or staircase up': False,
 'Find a ladder or staircase down': True,
 'Open a door (if it\'s closed)': False,
 'Unlock a door (if it\'s locked)': False,
 'Reach the goal': False
}
```

Note that the agent has moved north and then east, but no new items or doors have been discovered, and no doors have been opened or unlocked. However, the agent is now closer to the ladder or staircase down, which is a step towards reaching the goal.

### Response 5: Example response



### Llama-3-70B-Instruct

Based on my knowledge of NetHack, I break down the task of the agent into the following subgoals:

- \* Reach the staircase/ladder: The agent needs to find a way to move up or down the levels to progress in the game.
- \* Collect useful items: The agent needs to collect items such as pick-axes, keys, and lamps to overcome obstacles and access new areas.
- \* Unlock doors: The agent needs to find keys to unlock closed doors and access new areas.
- \* Avoid obstacles: The agent needs to avoid walls, locked doors, and other obstacles that can hinder its progress.
- \* Reach the goal: The agent needs to reach the final goal, which is not specified in this extract but is typically the treasure or the exit.

Now, considering the given game transition, I determine if any of the subgoals have been achieved at Time: 1 or not:

```
{
 'Reach the staircase/ladder': False,
 'Collect useful items': False,
 'Unlock doors': False,
 'Avoid obstacles': True,
 'Reach the goal': False,
}
```

The agent has avoided obstacles by moving north and not hitting the wall, but none of the other subgoals have been achieved at Time: 1.

Response 6: Example prompt for option discovery, where goals are not provided externally.

## C Scale

In this section we show how the F1 score varies according to the number of parameters of the LLM. For a fair analysis, we only compare models of the same family, where multiple sizes of the model are available. Namely, we consider: *Meta-Llama-3-8B-Instruct* and *Meta-Llama-3-70B-Instruct*; *gemma-1.1-2b-it* and *gemma-1.1-7b-it*; *Mistral-7B-Instruct-v0.2* and *Mixtral-8x7B-Instruct-v0.1*. Since *c4ai-command-r-plus* does not fit an Nvidia A100 80Gb, we exclude *c4ai-command-r-v01* from the comparison.

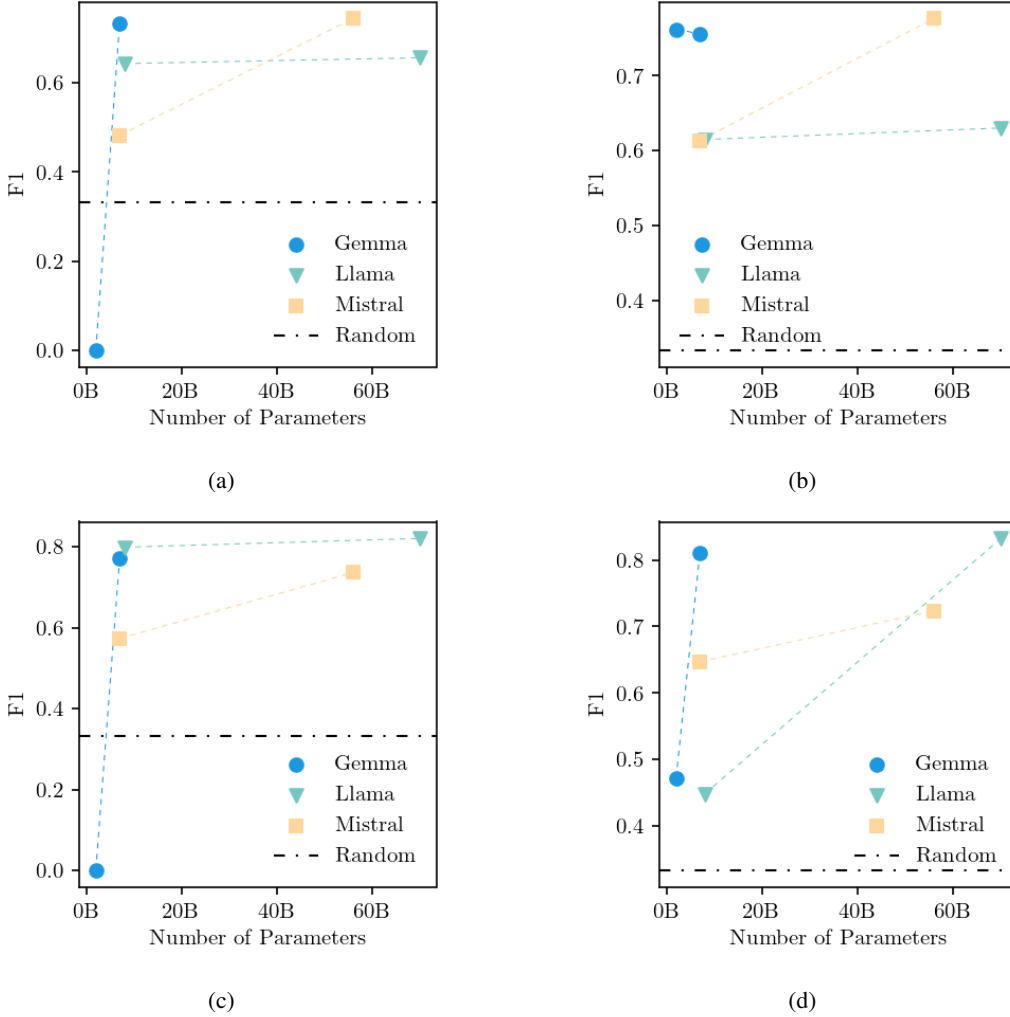


Figure 1: F1 score as a function of the LLM size.

## D Ablations

In this section, we include results of ablations, to understand which part of the prompt affect the performance the most. We present two ablations:

- (i) With and without a token separator to isolate each cell in the grid observation.
- (ii) With and without including actions in the transition.

### D.1 Tokenisation

## Examples of prompt with no token separator

The environment is MiniHack.

I will present you with a short extract of a gameplay. At each timestep, symbols represent the following items:

- "." represents a floor tile.
- "|" can represent either a wall, a vertical wall, an open door.
- "-" can represent either the bottom left corner (of a room), bottom right corner (of a room), wall, horizontal wall, wall, top left corner (of a room), top right corner (of a room).
- "+" represents a closed door. Doors can be locked, and require a key to open.
- "(" represents a useful item (pick-axe, key, lamp...)
- "<" represents a ladder or staircase up.
- ">" represents a ladder or staircase down.

The task of the agent is to win the game.

First, based on your knowledge of NetHack, break down the task of the agent into subgoals. Then, consider the following game transition, which might or might not contain these subgoals. Determine if any of the subgoals is achieved at Time: 1 or not.

Report your response in a dictionary containing the name of the subgoals as keys and booleans as value. For example:

```

{
 'python':
 {
 <name of goal>: <bool>,
 }
}

```

Observation Sequence:

```

<gameplay>
Time: 0
Current message: Never mind.

|. . |
|. . |
-@<|
.....|
. (. . .

Time: 1
Current message:

|. . |
|@. |
-.<|
.....|
. (. . .

</gameplay>

```

I will not consider anything that is not in the dictionary. You have only one shot at this, and you cannot ask for clarifications.

Prompt 8: Example of a prompt where the transition shows a successful door-unlock subgoal.

| Annotator                  | F1 ↑        | Accuracy ↑  | Precision ↑ | Recall ↑    | TP ↑       | TN ↑      | FP ↓     | FN ↓      |
|----------------------------|-------------|-------------|-------------|-------------|------------|-----------|----------|-----------|
| Human                      | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>171</b> | <b>85</b> | <b>0</b> | <b>0</b>  |
| Mixtral-8x7B-Instruct-v0.1 | <b>0.73</b> | <b>0.65</b> | 0.75        | <b>0.71</b> | <b>121</b> | 45        | 40       | <b>50</b> |
| c4ai-command-r-v01         | 0.66        | 0.63        | 0.87        | 0.53        | 90         | 72        | 13       | 81        |
| gemma-1.1-7b-it            | 0.66        | 0.64        | 0.89        | 0.52        | 89         | 74        | 11       | 82        |
| Mistral-7B-Instruct-v0.2   | 0.58        | 0.60        | 0.97        | 0.41        | 70         | 83        | 2        | 101       |
| Meta-Llama-3-8B-Instruct   | 0.54        | 0.56        | 0.90        | 0.39        | 66         | 78        | 7        | 105       |
| gemma-1.1-2b-it            | 0.53        | 0.49        | 0.69        | 0.43        | 74         | 52        | 33       | 97        |
| Meta-Llama-3-70B-Instruct  | 0.16        | 0.39        | <b>1.00</b> | 0.09        | 15         | <b>85</b> | <b>0</b> | 156       |
| c4ai-command-r-plus        | 0.02        | 0.33        | 0.40        | 0.01        | 2          | 82        | 3        | 169       |
| Random                     | 0.33        | 0.33        | 0.33        | 0.33        |            |           |          |           |

Table 5: Performance of LLM annotations with **game screen** observations, subgoals **provided** in the prompt, and **no token separator**.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Mixtral-8x7B-Instruct-v0.1 | <b>0.76</b>   | 0.68                | 0.77                 | <b>0.75</b>       | <b>128</b>    | 47            | 38              | <b>43</b>       |
| gemma-1.1-7b-it            | <b>0.76</b>   | 0.68                | 0.77                 | <b>0.75</b>       | <b>128</b>    | 46            | 39              | <b>43</b>       |
| c4ai-command-r-v01         | 0.74          | <b>0.69</b>         | 0.83                 | 0.67              | 115           | 62            | 23              | 56              |
| Meta-Llama-3-70B-Instruct  | 0.71          | 0.68                | 0.89                 | 0.60              | 102           | 72            | 13              | 69              |
| gemma-1.1-2b-it            | 0.68          | 0.57                | 0.67                 | 0.68              | 116           | 29            | 56              | 55              |
| c4ai-command-r-plus        | 0.64          | 0.61                | 0.84                 | 0.51              | 88            | 68            | 17              | 83              |
| Mistral-7B-Instruct-v0.2   | 0.59          | 0.61                | <b>0.99</b>          | 0.42              | 72            | <b>84</b>     | <b>1</b>        | 99              |
| Meta-Llama-3-8B-Instruct   | 0.49          | 0.52                | 0.87                 | 0.34              | 58            | 76            | 9               | 113             |
| Random                     | 0.33          | 0.33                | 0.33                 | 0.33              |               |               |                 |                 |

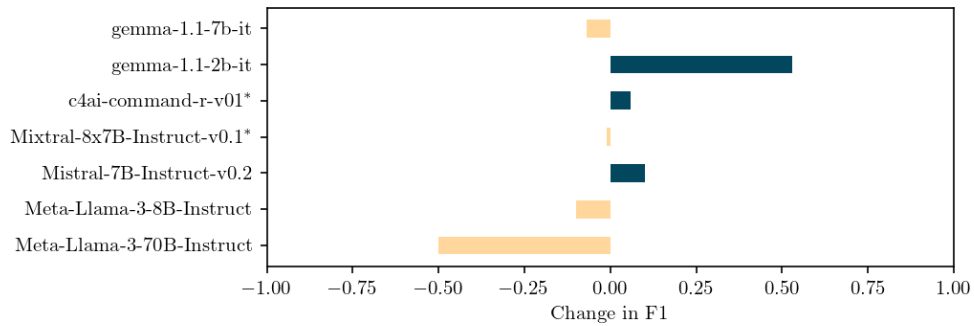
Table 6: Performance of LLM annotations with **cropped** observations, subgoals **provided** in the prompt, and **no token separator**.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Meta-Llama-3-70B-Instruct  | <b>0.81</b>   | 0.71                | 0.72                 | <b>0.92</b>       | <b>157</b>    | 25            | 60              | <b>14</b>       |
| c4ai-command-r-plus        | 0.79          | <b>0.74</b>         | <b>0.86</b>          | 0.73              | 125           | 64            | 21              | 46              |
| gemma-1.1-7b-it            | 0.75          | 0.63                | 0.68                 | 0.84              | 144           | 17            | 68              | 27              |
| Mixtral-8x7B-Instruct-v0.1 | 0.72          | 0.61                | 0.69                 | 0.76              | 130           | 27            | 58              | 41              |
| c4ai-command-r-v01         | 0.63          | 0.58                | 0.76                 | 0.54              | 92            | 56            | 29              | 79              |
| Meta-Llama-3-8B-Instruct   | 0.56          | 0.56                | <b>0.86</b>          | 0.41              | 70            | 74            | 11              | 101             |
| Mistral-7B-Instruct-v0.2   | 0.52          | 0.54                | 0.83                 | 0.38              | 65            | 72            | 13              | 106             |
| gemma-1.1-2b-it            | 0.00          | 0.33                | 0.00                 | 0.00              | 0             | <b>85</b>     | <b>0</b>        | 171             |
| Random                     | 0.33          | 0.33                | 0.33                 | 0.33              |               |               |                 |                 |

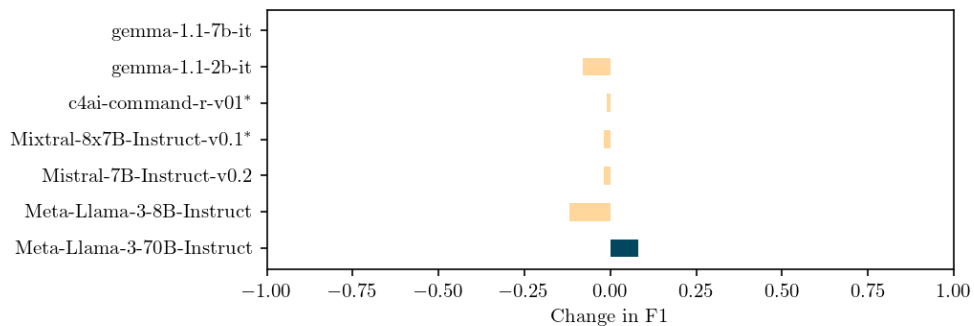
Table 7: Performance of LLM annotations with **game screen** observations, subgoals **suggested** by the LLM, and **no token separator**.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Meta-Llama-3-70B-Instruct  | <b>0.83</b>   | <b>0.74</b>         | 0.74                 | 0.94              | 161           | 29            | 56              | 10              |
| gemma-1.1-7b-it            | <b>0.83</b>   | 0.72                | 0.71                 | <b>0.98</b>       | <b>168</b>    | 17            | 68              | <b>3</b>        |
| Mixtral-8x7B-Instruct-v0.1 | 0.78          | 0.68                | 0.72                 | 0.85              | 146           | 27            | 58              | 25              |
| c4ai-command-r-plus        | 0.75          | 0.69                | <b>0.81</b>          | 0.71              | 121           | 56            | 29              | 50              |
| Meta-Llama-3-8B-Instruct   | 0.69          | 0.59                | 0.70                 | 0.68              | 116           | 35            | 50              | 55              |
| c4ai-command-r-v01         | 0.66          | 0.57                | 0.70                 | 0.63              | 107           | 39            | 46              | 64              |
| Mistral-7B-Instruct-v0.2   | 0.54          | 0.47                | 0.65                 | 0.47              | 80            | 41            | 44              | 91              |
| gemma-1.1-2b-it            | 0.00          | 0.33                | 0.00                 | 0.00              | 0             | <b>85</b>     | <b>0</b>        | 171             |
| Random                     | 0.33          | 0.33                | 0.33                 | 0.33              |               |               |                 |                 |

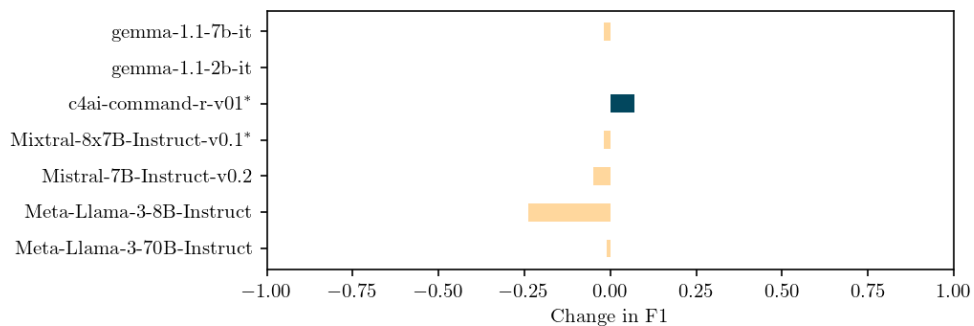
Table 8: Performance of LLM annotations with **cropped** observations, subgoals **suggested** by the LLM, and **no token separator**.



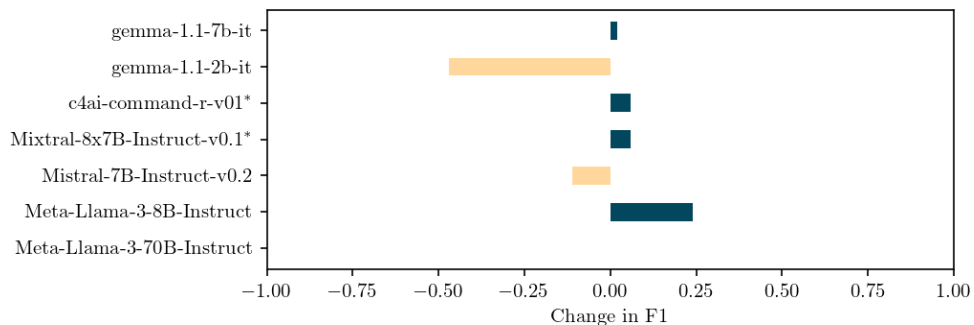
(a)



(b)

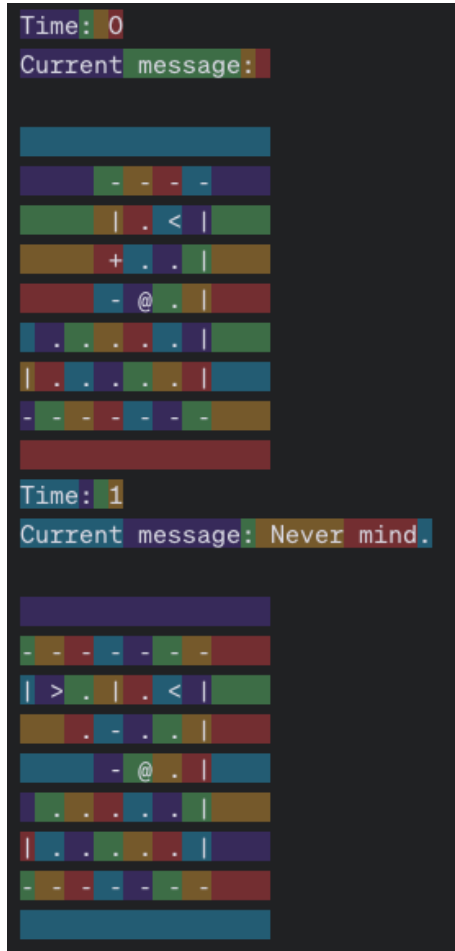


(c)

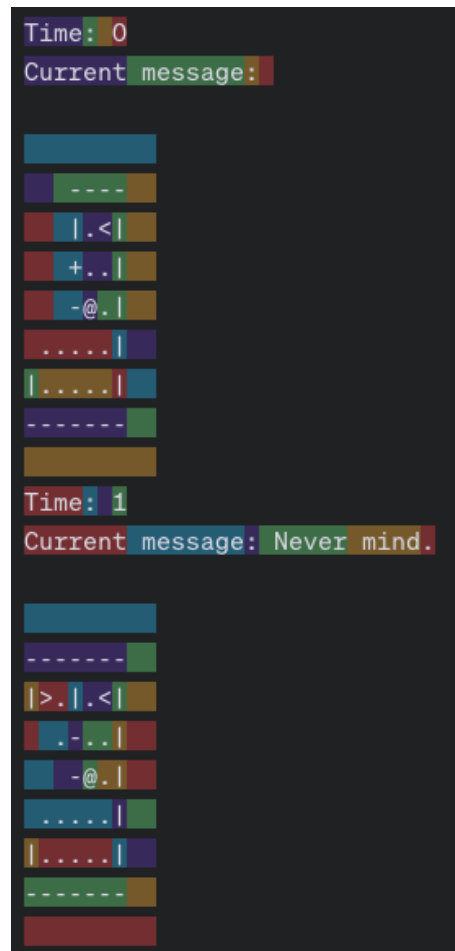


(d)

Figure 2: Variation in F1 score between the baseline results presented in Tables 1-4 and the results without a token separator in Tables 5-8. Yellow bars indicate worse performance without a separator, and blue otherwise.



(a)



(b)

Figure 3: Tokenisation of the same prompt, with (a) and without (b) a token separator (whitespace).

## D.2 Actions

In this section we investigate the impacts of explicitly adding the action  $a_t$  to the transition  $(s_t, a_t, s_{t+1})$ , which was left implicit in the main experiments in Section 5.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Mixtral-8x7B-Instruct-v0.1 | <b>0.83</b>   | <b>0.79</b>         | 0.89                 | <b>0.77</b>       | <b>132</b>    | 69            | 16              | <b>39</b>       |
| Mistral-7B-Instruct-v0.2   | 0.67          | 0.66                | 0.99                 | 0.50              | 86            | 84            | 1               | 85              |
| gemma-1.1-7b-it            | 0.66          | 0.66                | 0.97                 | 0.50              | 86            | 82            | 3               | 85              |
| c4ai-command-r-v01         | 0.66          | 0.64                | 0.93                 | 0.51              | 87            | 78            | 7               | 84              |
| Meta-Llama-3-8B-Instruct   | 0.61          | 0.62                | 0.97                 | 0.44              | 76            | 83            | 2               | 95              |
| Meta-Llama-3-70B-Instruct  | 0.46          | 0.53                | 0.98                 | 0.30              | 51            | 84            | 1               | 120             |
| gemma-1.1-2b-it            | 0.00          | 0.33                | 0.00                 | 0.00              | 0             | <b>85</b>     | <b>0</b>        | 171             |

Table 9: Performance of LLM annotations with **game screen** observations, subgoals **provided** in the prompt, and the transition includes actions.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Mixtral-8x7B-Instruct-v0.1 | <b>0.80</b>   | <b>0.75</b>         | 0.86                 | <b>0.75</b>       | <b>128</b>    | 64            | 21              | <b>43</b>       |
| gemma-1.1-7b-it            | 0.75          | 0.68                | 0.79                 | 0.72              | 123           | 52            | 33              | 48              |
| c4ai-command-r-v01         | 0.72          | 0.68                | 0.86                 | 0.62              | 106           | 68            | 17              | 65              |
| gemma-1.1-2b-it            | 0.71          | 0.62                | 0.73                 | 0.69              | 118           | 42            | 43              | 53              |
| Mistral-7B-Instruct-v0.2   | 0.66          | 0.66                | 0.98                 | 0.50              | 86            | 83            | 2               | 85              |
| Meta-Llama-3-70B-Instruct  | 0.64          | 0.59                | 0.78                 | 0.54              | 92            | 59            | 26              | 79              |
| Meta-Llama-3-8B-Instruct   | 0.50          | 0.54                | 0.90                 | 0.35              | 60            | 78            | 7               | 111             |

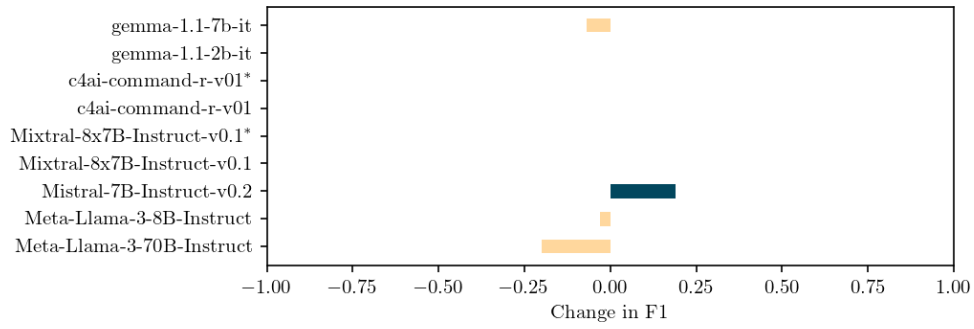
Table 10: Performance of LLM annotations with **cropped** observations, subgoals **provided** in the prompt, and the transition includes actions.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Meta-Llama-3-70B-Instruct  | <b>0.85</b>   | <b>0.79</b>         | 0.80                 | <b>0.91</b>       | <b>155</b>    | 46            | 39              | <b>16</b>       |
| Mixtral-8x7B-Instruct-v0.1 | 0.79          | 0.69                | 0.72                 | 0.86              | 147           | 29            | 56              | 24              |
| Meta-Llama-3-8B-Instruct   | 0.78          | 0.66                | 0.68                 | <b>0.91</b>       | <b>155</b>    | 13            | 72              | <b>16</b>       |
| gemma-1.1-7b-it            | 0.75          | 0.64                | 0.69                 | 0.82              | 141           | 22            | 63              | 30              |
| Mistral-7B-Instruct-v0.2   | 0.71          | 0.61                | 0.70                 | 0.71              | 122           | 33            | 52              | 49              |
| c4ai-command-r-v01         | 0.71          | 0.67                | <b>0.86</b>          | 0.60              | 102           | 69            | 16              | 69              |
| gemma-1.1-2b-it            | 0.00          | 0.33                | 0.00                 | 0.00              | 0             | <b>85</b>     | <b>0</b>        | 171             |

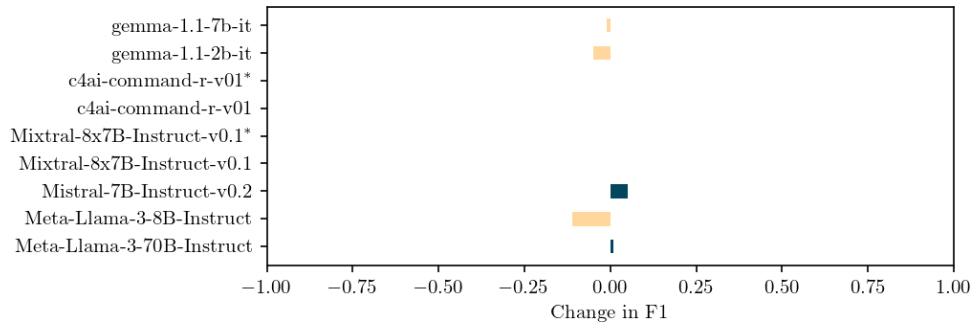
Table 11: Performance of LLM annotations with **game screen** observations, subgoals **suggested** by the LLM, and the transition includes actions.

| Annotator                  | F1 $\uparrow$ | Accuracy $\uparrow$ | Precision $\uparrow$ | Recall $\uparrow$ | TP $\uparrow$ | TN $\uparrow$ | FP $\downarrow$ | FN $\downarrow$ |
|----------------------------|---------------|---------------------|----------------------|-------------------|---------------|---------------|-----------------|-----------------|
| Human                      | <b>1.00</b>   | <b>1.00</b>         | <b>1.00</b>          | <b>1.00</b>       | <b>171</b>    | <b>85</b>     | <b>0</b>        | <b>0</b>        |
| Meta-Llama-3-70B-Instruct  | <b>0.86</b>   | <b>0.80</b>         | 0.82                 | 0.91              | 155           | 50            | 35              | 16              |
| Mixtral-8x7B-Instruct-v0.1 | 0.83          | 0.75                | 0.76                 | 0.91              | 156           | 35            | 50              | 15              |
| gemma-1.1-7b-it            | 0.82          | 0.72                | 0.73                 | <b>0.92</b>       | <b>158</b>    | 27            | 58              | <b>13</b>       |
| c4ai-command-r-v01         | 0.74          | 0.64                | 0.72                 | 0.76              | 130           | 35            | 50              | 41              |
| Mistral-7B-Instruct-v0.2   | 0.72          | 0.63                | 0.74                 | 0.70              | 120           | 42            | 43              | 51              |
| Meta-Llama-3-8B-Instruct   | 0.53          | 0.46                | 0.63                 | 0.46              | 79            | 39            | 46              | 92              |
| gemma-1.1-2b-it            | 0.42          | 0.48                | 0.83                 | 0.28              | 48            | <b>75</b>     | <b>10</b>       | 123             |

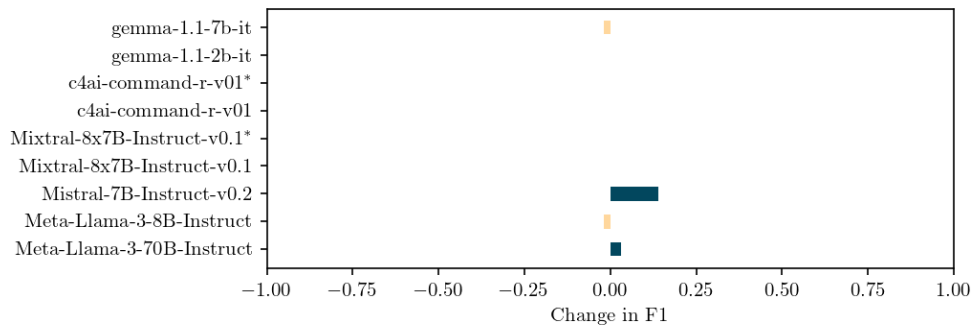
Table 12: Performance of LLM annotations with **cropped** observations, subgoals **suggested** by the LLM, and the transition includes actions.



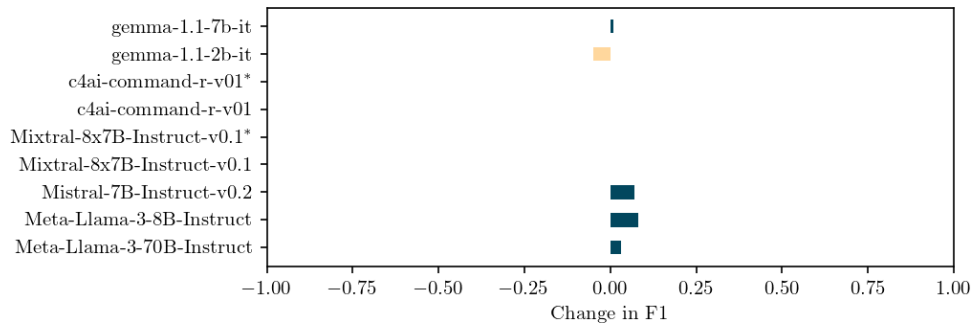
(a)



(b)



(c)



(d)

Figure 4: Variation in F1 score between the baseline results presented in Tables 1-4 and the results where prompts also include the action in Tables 9-12.