

Prompt Workflows

David Lee | TIM | 75

Today

- Prompt workflow examples
- HW 4 Prelab + Google Colab
- Upcoming deliverables
 - Homework 4 (prelab) due TODAY at 11:59pm
 - Homework 4 (individual) due SATURDAY at 11:59pm
 - Homework 4 (team) due NEXT MONDAY at 11:59pm

Note that Claude misses the instruction to provide proposed changes in its email draft.

Role	Content
User	You're our Chief Legal Officer. Review this SaaS contract for risks, focusing on data privacy, SLAs, and liability caps. <contract> {CONTRACT} </contract>
	Then draft an email to the vendor with your concerns and proposed changes.
Assistant	After reviewing the SaaS contract, I've identified several concerns: 1. Data Privacy (Section 7): - Issue: Vendor has broad rights to use client data for "service improvement." - Risk: Could lead to unauthorized data sharing or monetization. 2. SLAs (Section 5): - Issue: 99.5% uptime guarantee is below industry standard. - Risk: Excessive downtime could impact our operations. 3. Liability (Section 9): - Issue: Cap set at 3 months of fees. - Risk: Inadequate for potential data breaches or service failures.
	Draft email: Dear {{VENDOR}}, After reviewing the SaaS agreement, we have concerns: 1. Data Privacy: We require that our data only be used for direct service provision, not broad "service improvement." Please amend Section 7. 2. SLAs: 99.5% uptime is insufficient. We need 99.99% with clearly defined penalties. Please revise Section 5. 3. Liability: The current cap of 3 months' fees is inadequate. We propose a cap of 12 months' fees, with no cap for data breaches or willful misconduct. We value our potential partnership and believe these changes will set us up for success. Please send a revised agreement. Best regards, {{CLO_NAME}}

Prompt 1

Role	Content
User	You're our Chief Legal Officer. Review this SaaS contract for risks, focusing on data privacy, SLAs, and liability caps. <contract> {CONTRACT} </contract> Output your findings in <risks> tags.
Assistant {CONCERNS}	1. Data Privacy (Section 7): - Issue: Vendor has broad rights to use client data for "service improvement." - Risk: Could lead to unauthorized data sharing or monetization of client data. - Impact: Potential breach of client trust, regulatory fines (GDPR, CCPA). 2. SLAs (Section 5): - Issue: 99.5% uptime guarantee is below industry standard (typically 99.99%). - Risk: Up to 43.8 hours of downtime per year vs. 52.6 minutes at 99.99%. - Impact: Operational disruptions, revenue loss, client dissatisfaction. 3. Liability (Section 9): - Issue: Liability cap set at 3 months of fees. - Risk: Inadequate for potential data breaches, IP theft, or prolonged service failures. - Impact: Insufficient compensation for damages that could far exceed the cap, like class-action lawsuits or regulatory fines.

Prompt 2

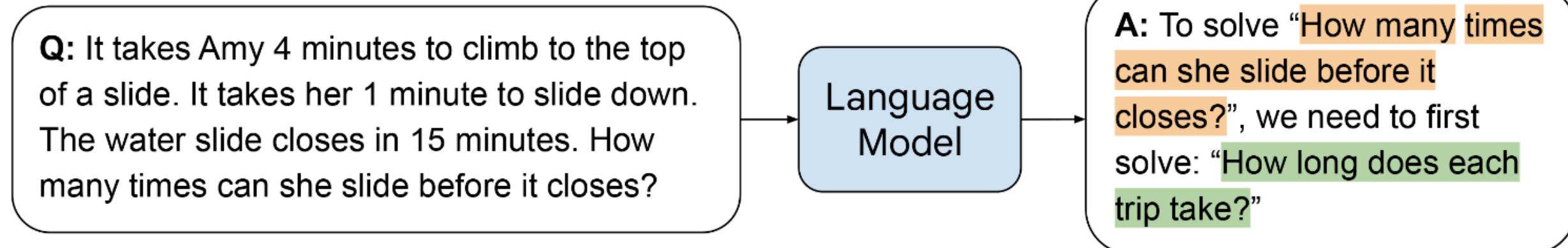
Role	Content
User	Draft an email to a SaaS product vendor outlining the following concerns and proposing changes. Here are the concerns: <concerns> {CONCERNS} </concerns>
Assistant {EMAIL}	Dear {{VENDOR}}, After a thorough review of the proposed SaaS agreement, our legal team has identified several high-priority concerns that must be addressed before we can proceed: 1. Data Privacy (Section 7): - Current: Broad rights to use client data for "service improvement." - Concern: Potential unauthorized data sharing or monetization. - Proposed Change: Limit data use to direct service provision. Any use for improvement, analytics, or marketing must be opt-in, revocable, and anonymized. 2. SLAs (Section 5): - Current: 99.5% uptime guarantee. - Concern: Up to 43.8 hours of downtime per year, risking operational disruptions. - Proposed Change: 99.99% uptime with tiered service credits. E.g., 10% credit for <99.99%, 25% for <99.9%, 50% for <99.5%. 3. Liability (Section 9): - Current: Cap at 3 months of fees. - Concern: Grossly inadequate for data breaches, IP theft, or prolonged outages. - Proposed Change: 12 months' fees for general liability. Uncapped for data breaches, willful misconduct, IP infringement. These changes are non-negotiable for us. They align with industry standards and our commitment to data security, operational excellence, and fiduciary responsibility. We value the potential of our partnership, and addressing these will set a strong foundation. Please have your legal team review and incorporate these changes. We're happy to discuss further to ensure mutual understanding and agreement. Best regards, {CLO_NAME}

Prompt 3

Role	Content
User	Your task is to review an email and provide feedback. Here is the email: <email> {EMAIL} </email> Give feedback on tone, clarity, and professionalism.
Assistant	1. Tone: - Professional and assertive without being confrontational. - Use of "non-negotiable" is strong but justified given the criticality of the issues. - Collaborative language ("we value," "happy to discuss") balances the firmness, maintaining a constructive tone. 2. Clarity: - Excellent structuring: Current state, concern, and proposed change for each issue. - Technical terms (e.g., "EBITDA," "tiered service credits") are appropriate for a B2B SaaS context. - Action items are clear: vendor's legal team to review and incorporate changes. 3. Professionalism: - Focuses on mutual benefit and industry standards, not just our demands. - Mentions broader responsibilities (data security, operational excellence, fiduciary duty), elevating the discussion. - Offers further discussion, showing commitment to understanding and resolving issues. Overall, the email strikes the right balance of assertiveness and collaboration. It clearly articulates our concerns, proposes specific solutions, and maintains a professional, partnership-oriented tone. No revisions needed; proceed with sending.

LEAST-TO-MOST PROMPTING ENABLES COMPLEX REASONING IN LARGE LANGUAGE MODELS

Stage 1: Decompose Question into Subquestions



Stage 2: Sequentially Solve Subquestions

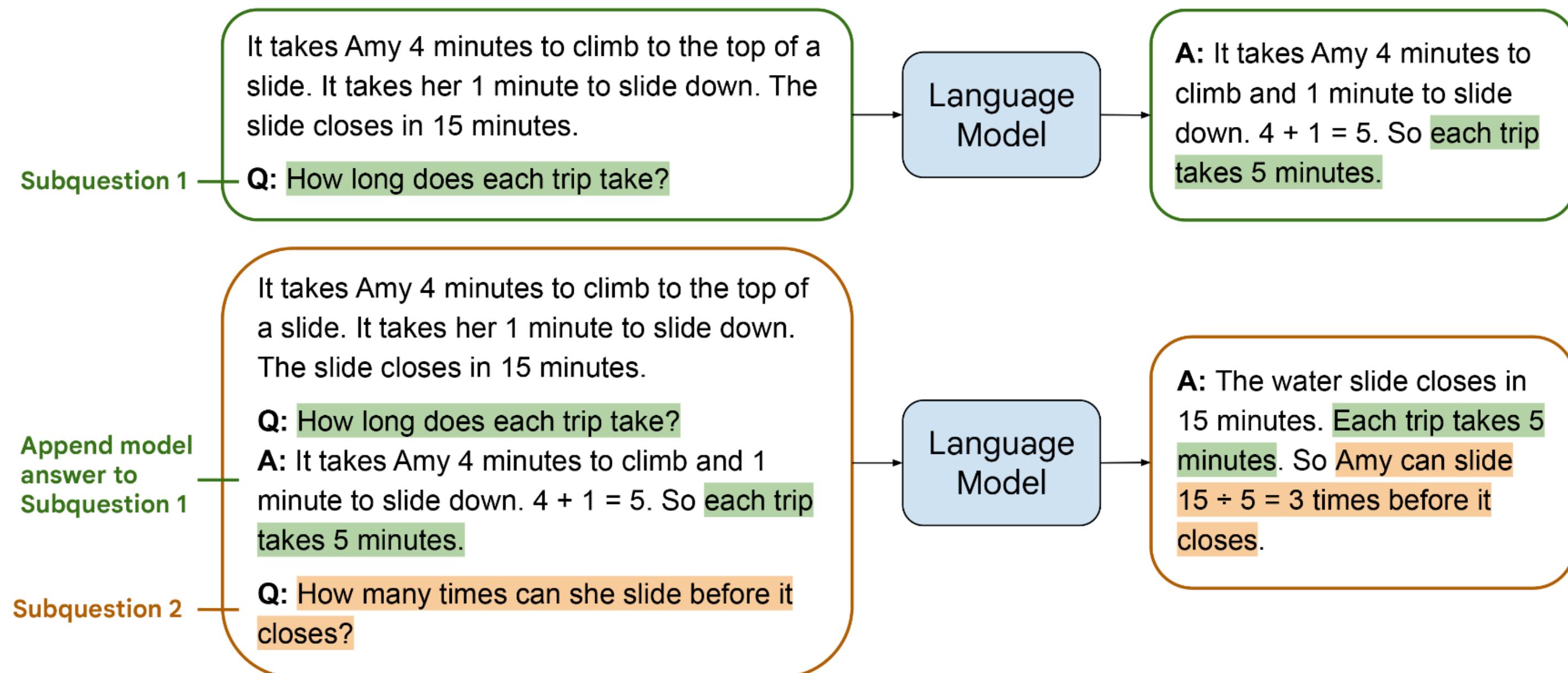


Figure 1: Least-to-most prompting solving a math word problem in two stages: (1) query the language model to decompose the problem into subproblems; (2) query the language model to sequentially solve the subproblems. The answer to the second subproblem is built on the answer to the first subproblem. The demonstration examples for each stage’s prompt are omitted in this illustration.

Concept Induction: Analyzing Unstructured Text with High-Level Concepts Using LLoOM

STATUS QUO TOPIC MODELING

UNSTRUCTURED TEXT

Toxic Online Content



TOPIC CLUSTERS

Feminist, Feminism, Feminists

Men, Percent, Built

Patriarchy, Economy, Time

Pay, Paid, Salaries

LLoOM CONCEPT INDUCTION

UNSTRUCTURED TEXT

Toxic Online Content



HIGH-LEVEL CONCEPTS

Misogynistic Language

Victim Blaming

Dismissiveness

Criticism of Feminism

UNSTRUCTURED TEXT

AI Impact Statements



HIGH-LEVEL CONCEPTS

Adversarial Attacks

Privacy Concerns

Efficient ML Algorithms

Generalization Improvement

UNSTRUCTURED TEXT

Political Social Media Feeds



HIGH-LEVEL CONCEPTS

Government Accountability

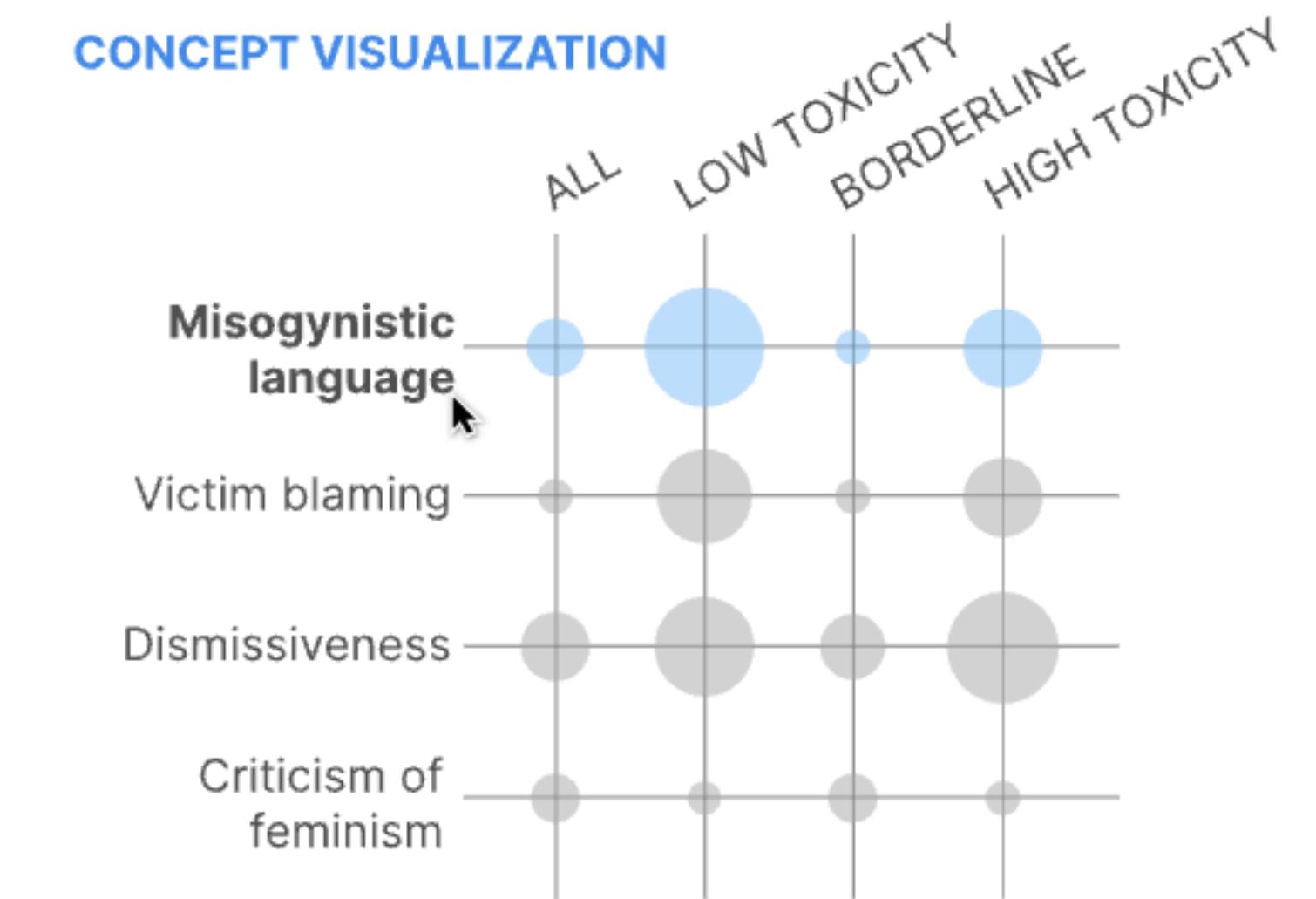
Concerns about Natl. Security

Social Justice

Public Health Concern

LLoOM WORKBENCH

CONCEPT VISUALIZATION



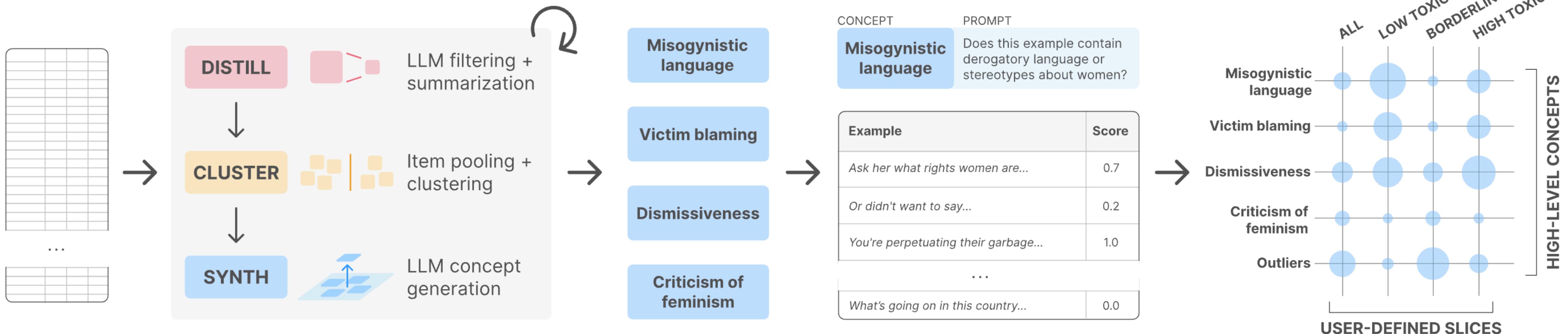
CONCEPT CRITERIA PROMPT

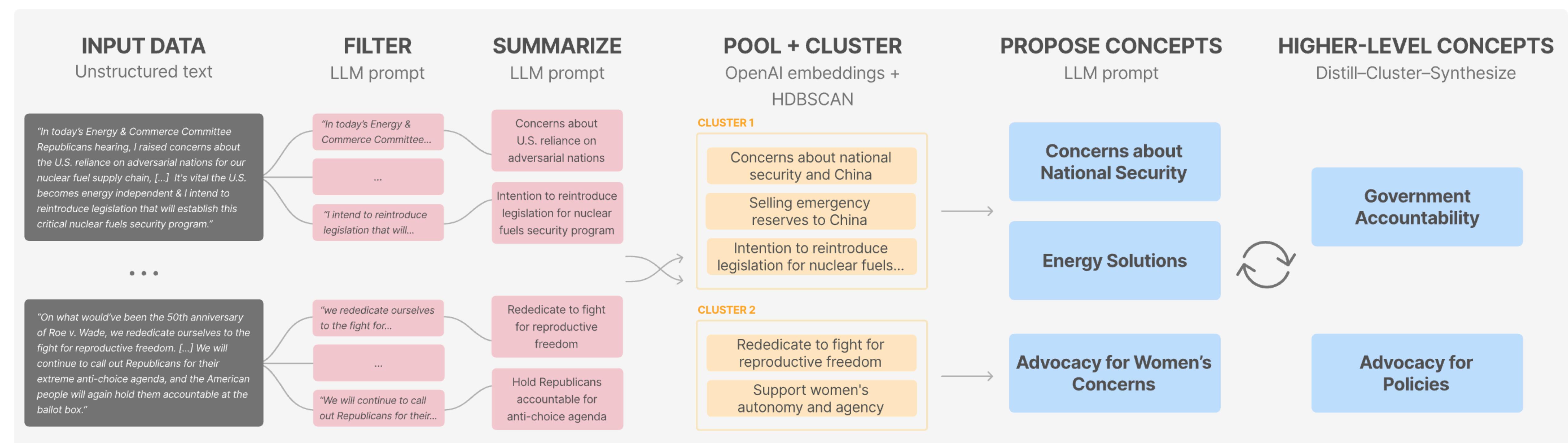
Misogynistic language

PROMPT: Does this example contain derogatory language or stereotypes about women?

CONCEPT MATCHES

Example Text	Score
Ask her what rights women...	0.7
Or didn't want to say...	0.2
You're perpetuating garbage...	1.0
...	
What's going on in this...	0.0





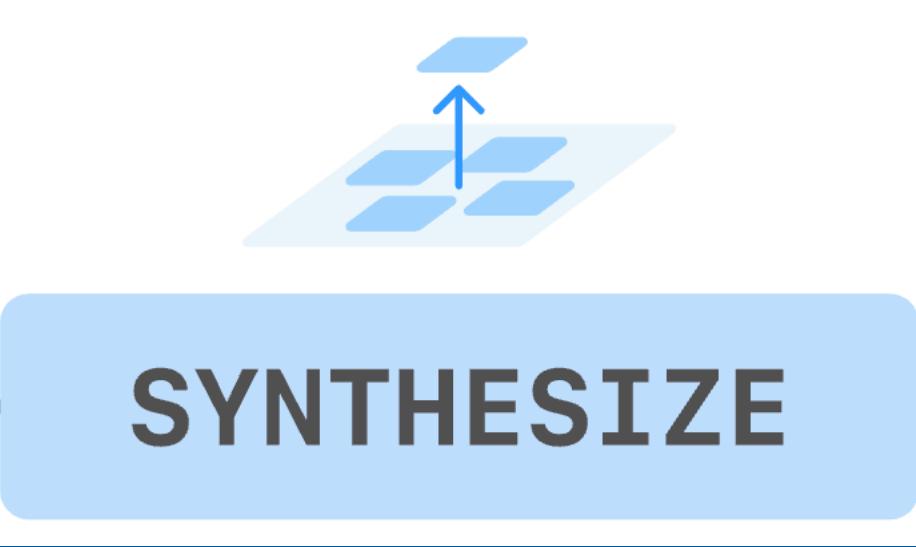
I have this set of bullet point summaries of text examples:

```
{bullets_json}
```

Please write a summary of {n_concepts} unifying patterns for these examples {seed_phrase}. For each high-level pattern, write a {n_name_words} word NAME for the pattern and an associated one-sentence ChatGPT PROMPT that could take in a new text example and determine whether the relevant pattern applies.

Please also include {n_example_ids} example_ids for items that BEST exemplify the pattern. Please respond ONLY with a valid JSON in the following format:

```
{}  
  "patterns": [  
    {}  
      "name": "<PATTERN_NAME_1>"  
      "prompt": "<PATTERN_PROMPT_1>"  
      "example_ids": ["<EXAMPLE_ID_1>", "<  
EXAMPLE_ID_2>"]  
    }  
    {}  
      "name": "<PATTERN_NAME_2>"  
      "prompt": "<PATTERN_PROMPT_2>"  
      "example_ids": ["<EXAMPLE_ID_1>", "<  
EXAMPLE_ID_2>"]  
    }  
  ]  
}
```



I have the following TEXT EXAMPLE:

```
{text_example_json}
```

Please extract {n_quotes} QUOTES exactly copied from this EXAMPLE {seed_phrase}.

Please respond ONLY with a valid JSON in the following format:

```
{}  
  "relevant_quotes": [ "<QUOTE_1>", "<QUOTE_2>", ...  
]  
}}
```

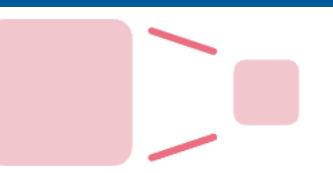
I have the following TEXT EXAMPLE:

```
{text_example_json}
```

Please summarize the main point of this EXAMPLE {seed_phrase} into {n_bullets} bullet points, where each bullet point is a {n_words} word phrase.

Please respond ONLY with a valid JSON in the following format:

```
{}  
  "bullets": [ "<BULLET_1>", "<BULLET_2>", ... ]  
}}
```



DISTILL

Condense input data

step. For the Cluster operator to generate *cross-cutting* concepts, all of the distilled bullet points are detached from their original examples and pooled together. Thus, the input of the Cluster operator is the set of condensed bullet points from the Distill operator, and the output is a set of group assignments, such that each isolated bullet point is assigned to a group of related items. The LLoM algorithm transforms bullet points into embeddings using a specified pre-trained embedding model and then clusters the items using a provided clustering algorithm. Our implementation uses OpenAI's



CLUSTER

Group related items

CONTEXT:

```
I have the following text examples in a JSON:  
{examples_json}
```

I also have a pattern named {concept_name} with
the following PROMPT:

```
{concept_prompt}
```

TASK:

For each example, please evaluate the PROMPT by generating RATIONALE of your thought process and providing a resulting ANSWER of ONE of the following multiple-choice options, including just the letter:

- A: Strongly agree
- B: Agree
- C: Neither agree nor disagree
- D: Disagree
- E: Strongly disagree

Respond with ONLY a JSON with the following format, escaping any quotes within strings with a backslash:

```
{  
  "pattern_results": [  
    {  
      "example_id": "<example_id>"  
      "rationale": "<rationale>"  
      "answer": "<answer>"  
    }  
  ]  
}
```

Score. Armed with the concepts, LLooM next applies a score (e.g., 0-1) that describes the association between each input and the concept. For each high-level concept, the system applies the Score operator to all examples (input texts) to generate a concept score that estimates how well each example matches the generated concept prompt. This is implemented using a batched zero-shot prompt that includes a set of examples in JSON format, the concept prompt, and instructions to generate an answer in multiple-choice format. Prior work has found that LLMs do not provide calibrated 0-1 confidence scores in zero-shot settings [38]. However, recent work has found that for instruction-tuned OpenAI models such as GPT-3.5, multiple choice prompting [53, 54] can provide approximate answer probabilities. We use multiple choice prompting to instruct the model to generate a multiple-choice answer⁴ for each provided example along with a rationale. These answers are parsed and converted to bucketed numerical scores with “Strongly agree” mapping to 1.0 and “Strongly disagree” mapping to 0.0. The scores

Loop. Finally, based on the concept scoring results, LLooM can use a Loop operator to execute multiple iterations of the algorithm. This operator executes the logic to *revise the inputs* to the next iteration of the pipeline. We use *data coverage* to determine which examples will be processed in each subsequent iteration. After the concept scoring phase completes, the Loop operator identifies two classes of outliers: 1) *not-covered* examples, which did not match *any* of the current high-level concepts and 2) *covered-by-generic* examples, which only matched “generic” concepts, those that matched a majority of examples (at least 50%). All such examples are provided as input to the next iteration of the algorithm, and the concepts generated by subsequent runs are added to the full set of concepts.

LLMs as Workers in Human-Computational Algorithms? Replicating Crowdsourcing Pipelines with LLMs

Pipeline	Description	Sample Task	Replication evaluation			
			Total	Unique	Correct	Effective
Map-Reduce (Kittur et al., 2011)	<i>Partition</i> tasks into discrete subtasks, <i>Map</i> subtasks to workers, <i>Reduce</i> / merge their results into a single output	Write essay	4	1	3	3
Input Essay topic	Partition Outline	Map Facts	Reduce Paragraph			
The history of religious influence on American schools	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 30%;"> Origins of Religious Education </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> The Blaine Amendment and Its Impact ... </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> The Future of Religious Education in the U.S. </div> </div>	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 30%;"> The earliest educational institutions ... were founded by religious groups in the 1600s ... </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> ...Protestant denominations established colleges and the teaching of moral values. </div> <div style="border: 1px solid black; padding: 5px; width: 30%;"> 20th century...offer religious classes, although the scope... </div> </div>	The Origins of Religious Education in the United States can be traced back to the earliest educational institutions of the 1600s, which were established by religious groups in order to provide religious instruction and moral guidance to their students. During the 19th century, Protestant denominations established colleges and universities that focused on religious instruction and the teaching of moral values. This led to the development of religious education as a distinct field of study in the United States. In the 20th century, some states began to allow public schools to offer religious classes, although the scope and content of such classes varied widely from state to state. This allowed religious education to become more widely available to US students...			

Find-Fix-Verify (Bernstein et al., 2010)

For writing and editing: *Find* problems, *Fix* the identified problems, *Verify* these edits

Shorten text

3

3

2

1

Input Text to be shortened

Print publishers are in a tizzy over Apple's new iPad because they hope to finally be able to charge for their digital editions. But in order to get people to pay for their magazine and newspaper apps, they are going to have to offer something different that readers cannot get at the newsstand or on the open Web.

Find Verbose parts

hope to finally ...
their digital editions

in order to get
people to... apps

offer something...
on the open Web

Fix

hoping to charge for
digital editions

to get people to pay
for apps

offer something unique
not available elsewhere

Shorten the verbose parts and put back in context

Print publishers are in a tizzy over Apple's new iPad because they hoping to charge for digital editions. But to get people to pay for apps, they are going to have to offer something offer something unique that is not available elsewhere.

Verify

Print publishers are in a tizzy over Apple's new iPad because they hoping to charge for digital editions. But to get people to pay for apps, they are going to have to offer something offer something unique that is not available elsewhere.

HumorTool (Chilton et al., 2016)	Define semantic roles as the answers to a series of questions that are intuitive for non-experts.	Create satire	4	2	3	1		
Input	News headline	Aspect	from headline	Association	Related concepts	Joke	on headline using aspect and association	<i>Unused steps</i>
FDA Approves First Over-The-Counter Narcan To Reduce Overdoses		Overdoses		Drug Withdrawal symptoms	→	Guess I know what I'm buying next time I'm at the grocery store... Narcan and chips, anyone?		Expected Reaction Violation Expected Reason Belief
		Over-The-Counter		Supplements Pharmacies	→	Wow, you can now buy Narcan over-the-counter? Now you don't even need a prescription to get your hands on the latest party drug!		

Task Paraphrase (He et al., 2015)	Define semantic roles as the answers to a series of questions that are intuitive for non-experts.	SRL labeling	1	1	1	1
Input	Unlabeled text	Question about verb "posted"	Answer	Labeling	Map answers to SRL	
On Friday, Clark posted to Facebook to explain his decision saying , "I usually don't post political items, but today is different."		Who posted something?	Clark		ARG0: Clark	
		Why did someone post something?	to explain their decision		ARGM-PRP: to explain his decision	
		When did someone post something?	On Friday		ARGM-TMP: On Friday	

Iterative Process (Little et al., 2010)	Feed the result of one creation task into the next, so workers see content generated by previous workers.	Brainstorm	3	2	3	2
Input	Query	Iter 0 5 titles	Iter 1 "better and more creative" titles	Iter 2 "better and more creative" titles		
Title for a short story about talking dogs		1. A Woof of a Tale 2. Barking Up the Right Tree 3. The Adventures of the Chatty Canines 4. The Dog Who Could Speak 5. The Journey of the Talking Dogs	1. A Ruff and Ready Conversation 2. The Speak of the Dogs 3. A Lively Pack of Pups 4. Tales From the Bark Side 5. The Barkings of a Different Breed	1. A Fido's Flap of the Tongue 2. The Secret Dialogues of Dogs 3. The Woofs of Wisdom 4. Canine Conversations in Canine-ese 5. Tongue-Tied Terriers		

Microtasking (Cheng et al., 2015)	Concrete microtasking for sorting task: an implementation of human-powered quicksort	Sorting	3	3	3	1
Input	Number list	Pair Pairwise combine.	Compare #odd numbers	Sort		
1 2 3 4 5 5 4 3 2 1 2 3 4 5 6		5 4 3 2 1 ? 2 3 4 5 6 1 2 3 4 5 ? 2 3 4 5 6 1 2 3 4 5 ? 5 4 3 2 1	5 4 3 2 1 > 2 3 4 5 6 1 2 3 4 5 > 2 3 4 5 6 1 2 3 4 5 = 5 4 3 2 1	2 3 4 5 6 1 2 3 4 5 5 4 3 2 1		

Price-Divide-Solve (Kulkarni et al., 2012)	Workers recursively divide complex steps until they are at an appropriately simple level, then solve them.	Write essay	1	1	1	1
Input	Topic	Split Find subtasks	Complete Complete subtasks (or further split)	Global solution		
Write a 1-paragraph description about mascot at Carnegie Mellon University		Research the mascot Write paragraph Proof read & shorten	The mascot ... is the Tartan Scot...cartoon character... Our mascot is... It is a representation of our school's commitment to excellence in education... Carnegie Mellon University's mascot is...	Carnegie Mellon University's mascot is the Tartan Scot, a cheerful cartoon character representing the school's Scottish heritage. The mascot is... embodying the school's commitment to excellence in education...		

Designing LLM Chains by Adapting Techniques from Crowdsourcing Workflows

Objectives

What the designer wants the workflow to accomplish



Outcome Quality

High factuality,
creativity



Resource Constraints

Effort, latency,
expense

Designers must work within **resource constraints** to improve **outcome quality**.



Objectivity

Math,
question-answering

Spectrum



Subjectivity

Writing, prototyping

The definition of quality may be **objective**, **subjective**, or some combination thereof.



Task Specific

Only applied to one
problem (e.g.,
translation workflow)

Spectrum



Task General

Can be applied to
many problems
(e.g., map-reduce)

Workflows vary from supporting a **specific** task to supporting a **general** set of tasks.

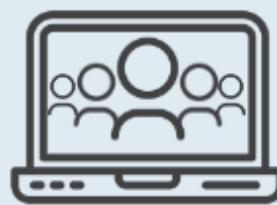
Tactics

The building blocks of workflows

Actors

Crowdworkers

People recruited from an online platform



LLMs

Generative language models and equivalents



User

The person deploying the workflow



Subtasks

Generate

Create content

Write a caption

Focus

Select content

Select regions for future work

Evaluate

Judge content

Is this answer grammatical?

Partition

Split tasks

Divide the task into subtasks

Improve

Improve content

Fix the errors

Merge

Combine content

Compose many outlines into one

Architectures

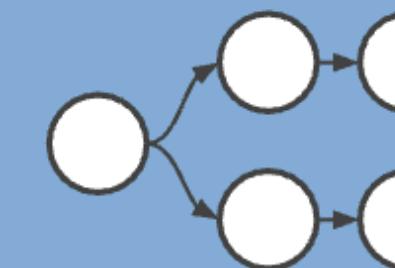
Sequential

Call to workers in a sequence



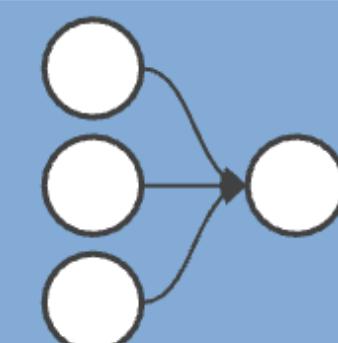
Branching

Diverge on paths conditionally



Redundant

Repeat a subtask with multiple workers



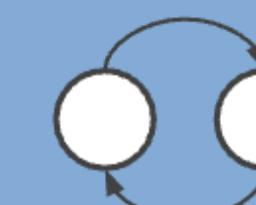
Dynamic

Initialize parts of the workflow at runtime



Communicative

Facilitate interaction among workers



Diverse responses

To promote objective or subjective quality source multiple responses to the same subtask with parallel, iterative, or communicative workflows.

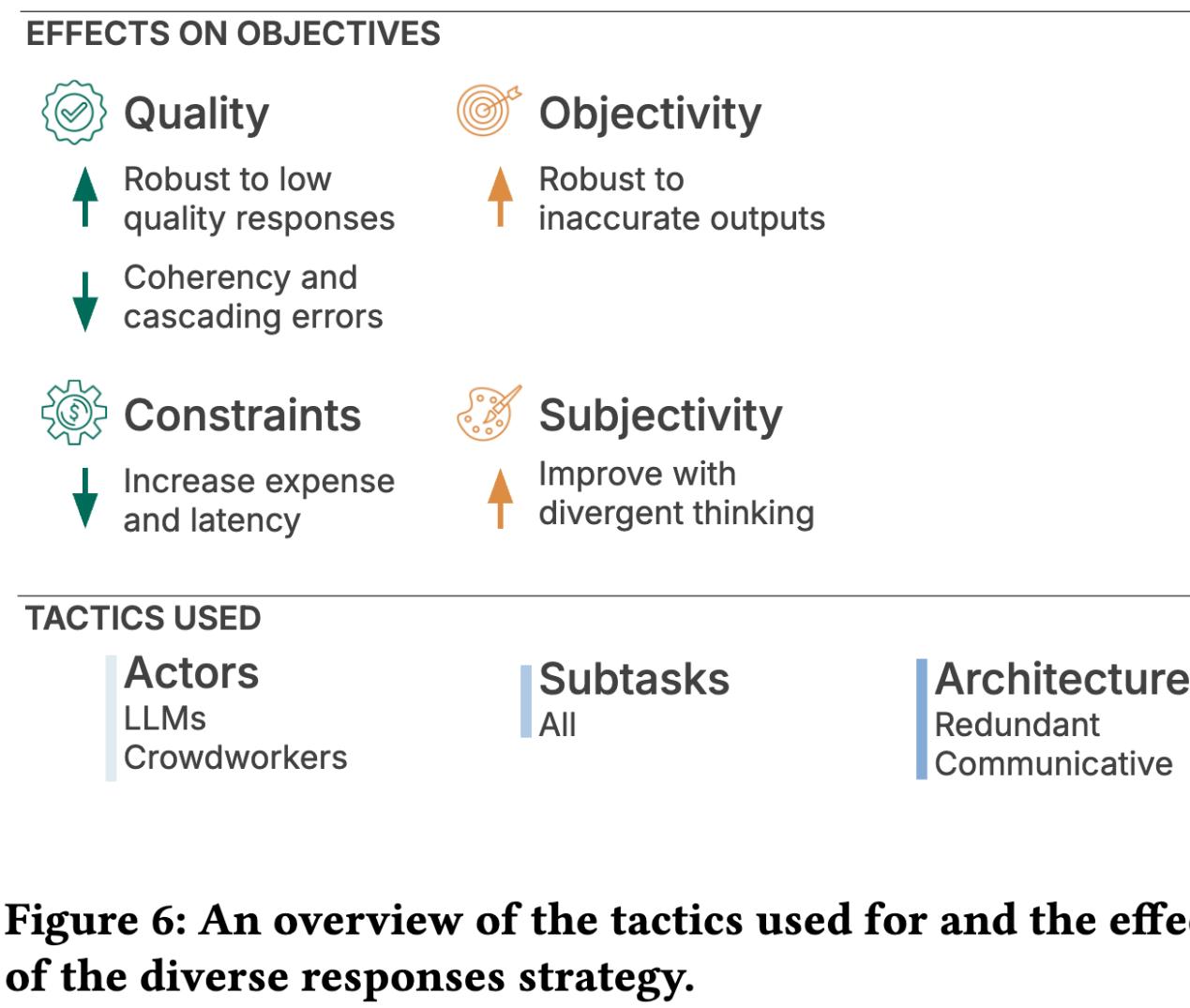


Figure 6: An overview of the tactics used for and the effects of the diverse responses strategy.

Quality thresholds

To more precisely balance the quality-cost tradeoff, measure quality repeatedly and stop the workflow once a quality threshold is reached.

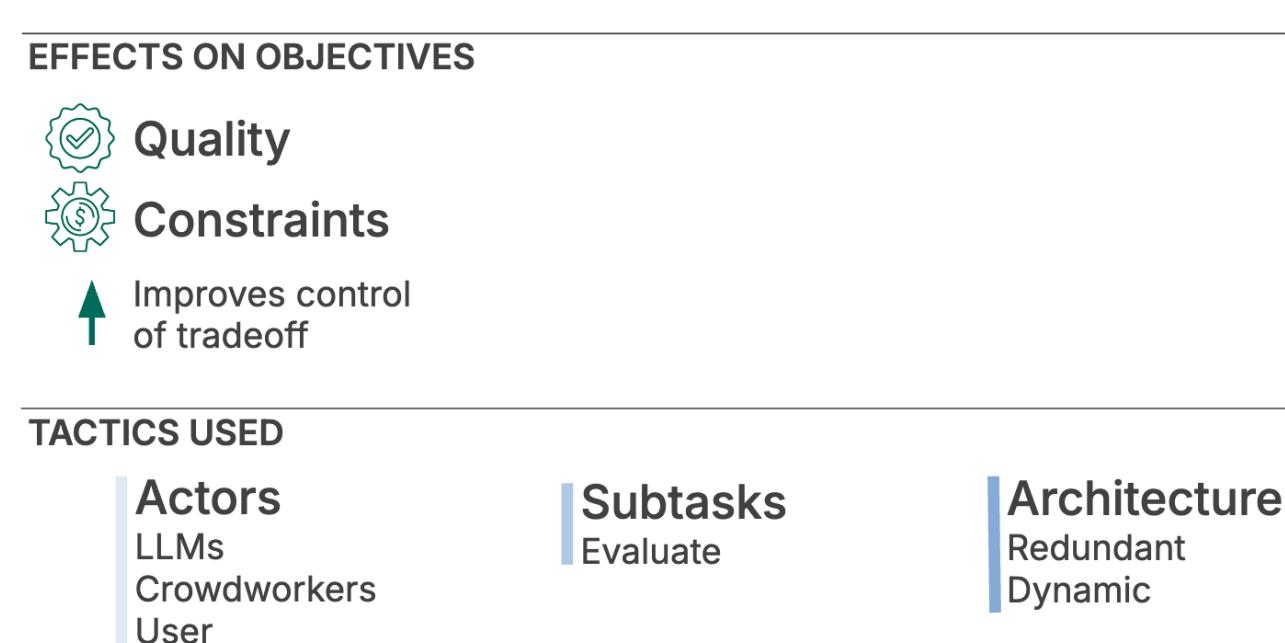


Figure 13: An overview of the tactics used for and the effects of the quality thresholds strategy.

Context

To promote coherence and improve subtask outcomes, provide the minimum sufficient global context while optimizing for clarity of actor instructions.

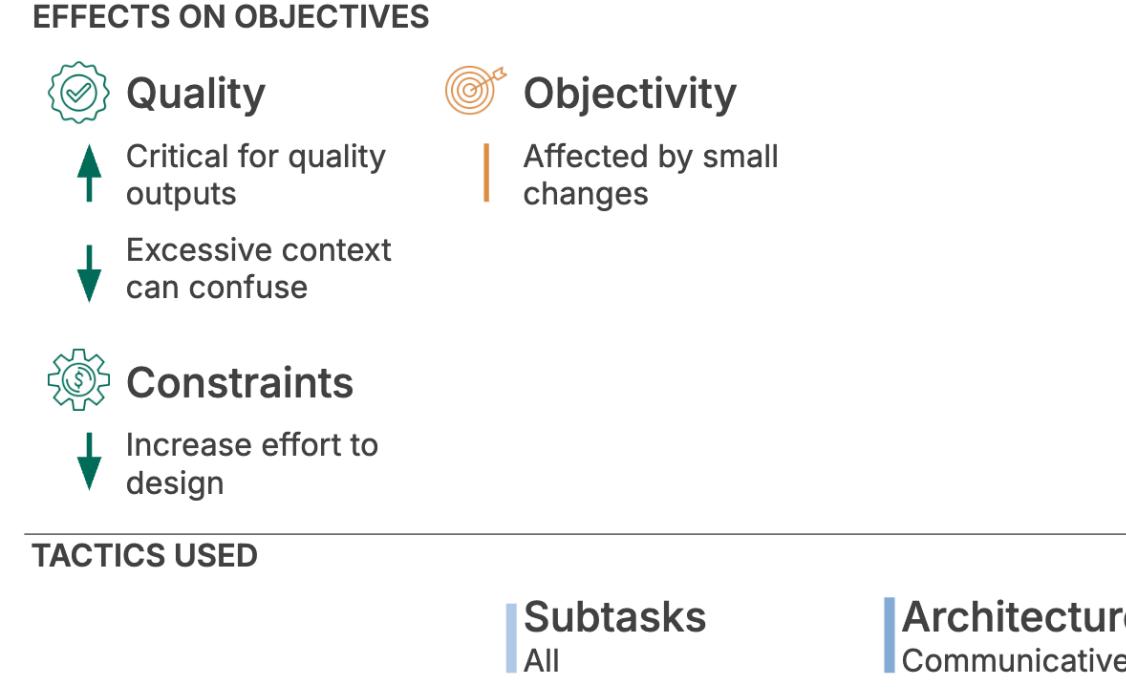


Figure 12: An overview of the tactics used for and the effects of the context strategy.

Validation

To improve quality of the overall output, validate and correct subtask outputs with deterministic, architectural, or actor-based checks.

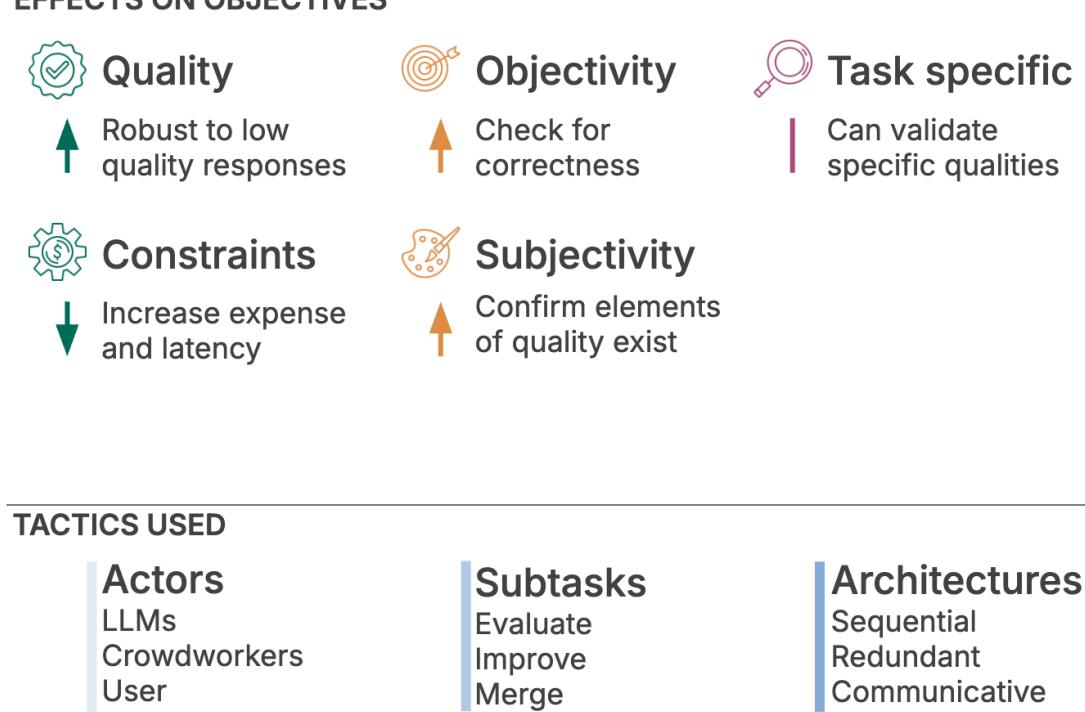


Figure 7: An overview of the tactics used for and the effects of the validation strategy.

Adaptable architectures

To build workflows that require less effort to design and that generalize to many inputs, use adaptive architectures that build themselves at runtime.

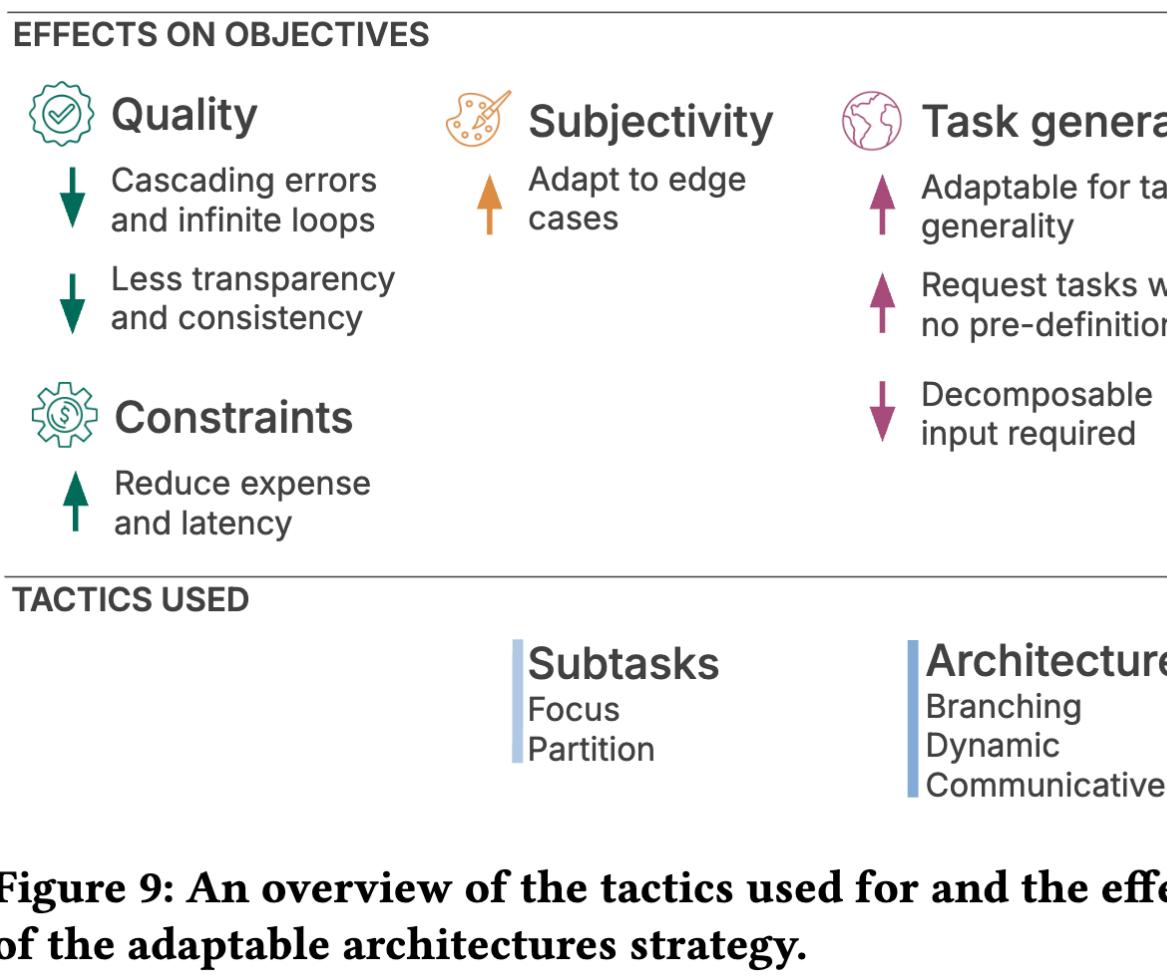


Figure 9: An overview of the tactics used for and the effects of the adaptable architectures strategy.

Subtask alignment

To improve outcome quality at each step, align subtasks to the actors' capabilities.

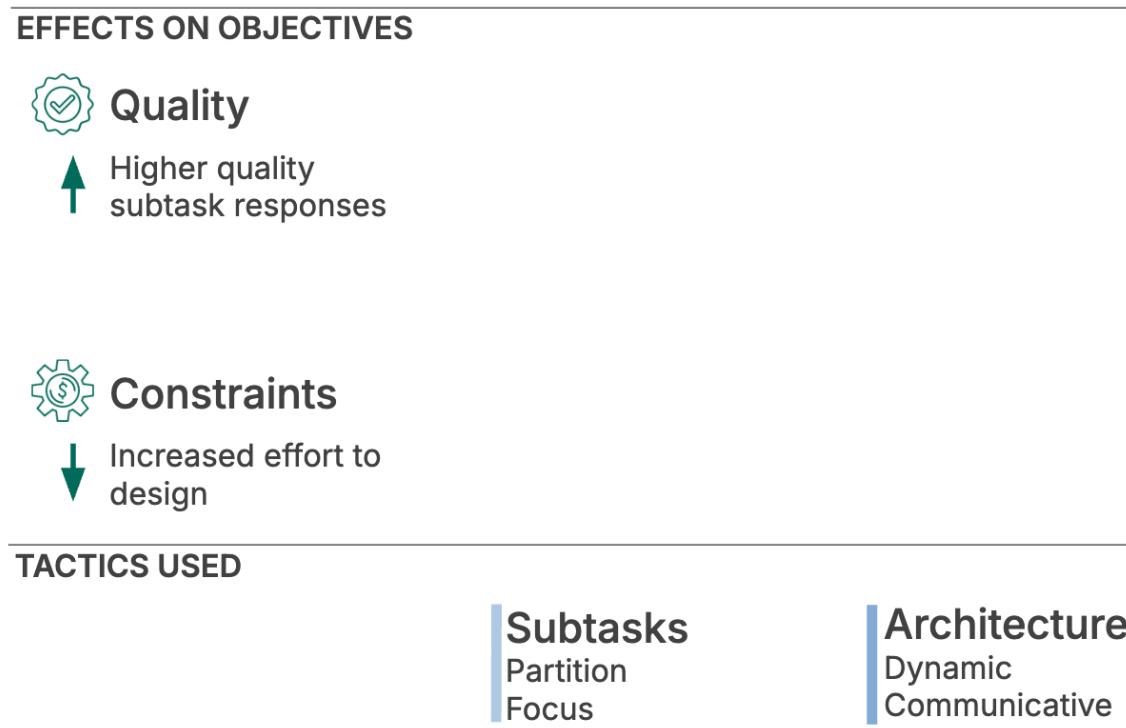


Figure 11: An overview of the tactics used for and the effects of the subtask alignment strategy.

Actor selection

To improve outcome quality at each step, curate a high quality actor pool by filtering actors by their skills and improving actor capabilities over time.

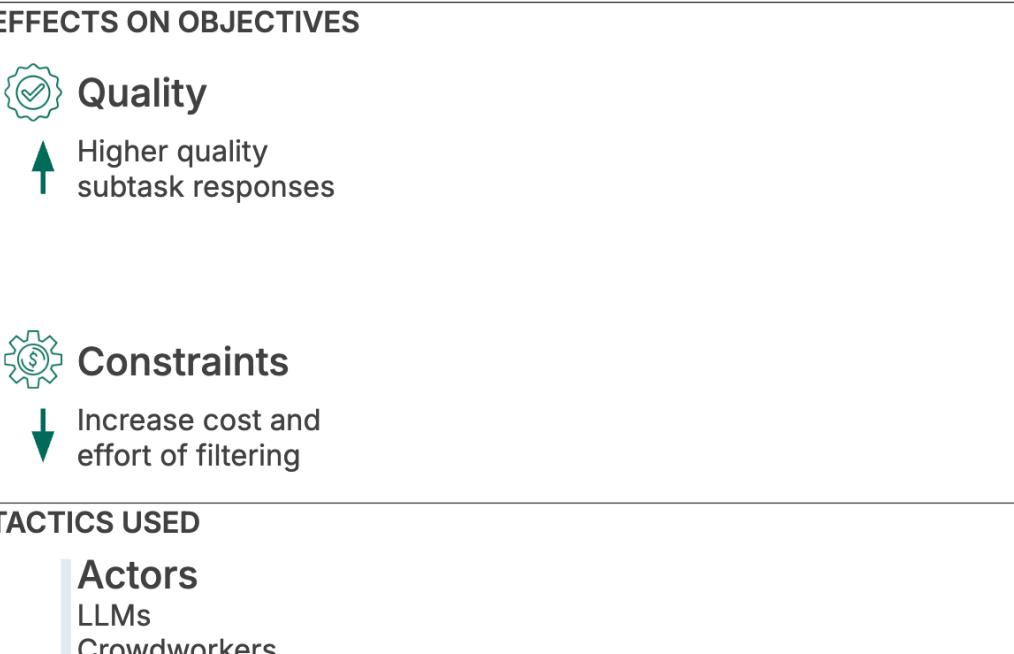


Figure 10: An overview of the tactics used for and the effects of the actor selection strategy.

User guidance

To help the user explore the design space and align subjective outcomes to their vision, incorporate their input into the workflow at runtime.

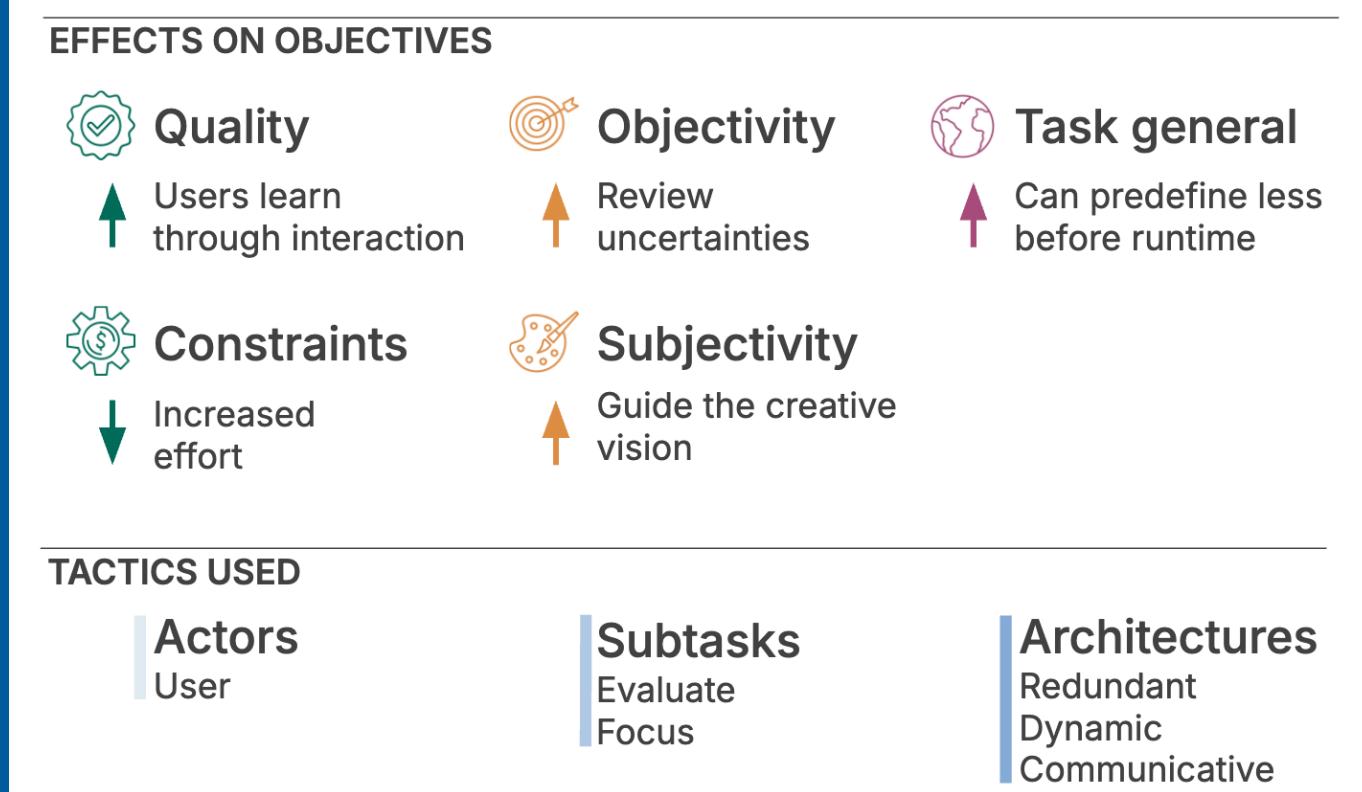


Figure 8: An overview of the tactics used for and the effects of the user guidance strategy.

Prelab overview

Templating & Prompt Workflows

TIM 175 WEEK 4 PRELAB

Brief Task Overview

Complete the following activities to understand methods for evaluating prompts and outputs:

1. Activity 1: Templating and Workflows in LastMileAI
2. Activity 2: Templating and Workflows in Google Colab

Activity 2: Templating and Workflows in Google Colab

Now that we have played around with templating and workflows on LastMileAI, let's try to implement it in code!

Set Up Google Colab

- Create a copy of [Gemini Workflow Template.ipynb](#)
- Follow the detailed steps [here](#) to make an API key and add it as an access token

Go through the Gemini Workflow Template Walkthrough.

Before diving into the activity, go through the walkthrough defined within the Google Colab notebook, which will guide you in the following steps:

1. Run the cell to import necessary packages.
2. Run the cell to define the API call.
3. Run the cell for setting the Google API Key.
4. Run the cell for defining the input variables, the prompt, and for printing the prompt
5. Run the cell for calling the Gemini API and printing out the response
6. Run the cell for defining the follow-up prompt.
7. Run the cell for calling the Gemini API and printing out the response for the follow-up prompt.

Task:

Use the above notebook to automate meal planning based on individual dietary needs for a bunch of people (Data given below as json). You need to figure out how to dynamically run the workflow with the different inputs, without having to copy paste manually or alter the prompt for every person. The workflow should dynamically generate recipes, grocery lists, and estimated costs for each individual person and store it in a table/CSV file.

Steps to Follow

1. Read Input Data

- Load a **CSV or JSON file** containing details for each person (name, diet, allergens, dislikes, favorite foods).

2. Generate a Recipe

- Use a **templated prompt** to create a recipe based on dietary preferences.

3. Create a Grocery List

- Write a prompt that takes the above recipe and provides a grocery list from the generated recipe ingredients.

4. Estimate Grocery Costs

- Write a prompt that takes the generated grocery list and estimates the total cost for all grocery items.

5. Run the Workflow for All Individuals

- Loop through each person's data and apply the workflow dynamically.
- You should run the cell once and get output for each person.