The Complexity Stress Test Project

1 Prompts definition

Text complexification is the task of generating text variants at specific complexity levels that maintain semantic fidelity to the original input. The idea of the framework is simple: take an input text and make it more complex, but in a controlled and measurable way. To do this, the system does not rely on a single LLM call, but on a small team of two agents that interact with each other and with an external scoring module.

Here, we define the prompts to be used for the two cooperative agents to achieve the text complexification goal. Both agents are provided with the specified task, after which they collaboratively communicate by chatting with each other in an instruction-following fashion to solve the specified task.

Note that the provided prompts are a starting point. Before testing the project on the test data, it is essential that students test them and modify them as needed to ensure that the agents follow the instructions correctly (e.g., correct output format).

It should be noted that the prompts provided in this text serve as example guidelines. Prompts must be composed appropriately based on the agentic framework being used, in our case Agno.[1].Note that since we are working with a multi-agent system, the prompts should be structured accordingly to reflect this architecture.

1.1 System prompts

In our multi-agent system we first need to define the system prompts, which is the initial, high-priority message that defines the identity, objectives, rules, and style of an agent (or team of agents). The system prompt remains persistently in memory throughout the entire conversation, maintaining constant influence over the agent's behavior. This ensures the agent stays aligned with its core identity and rules regardless of conversation length, preventing behavioral drift and enabling consistent role adherence in multi-agent collaborations.

The system prompt for the Text Complexification agent is defined as follows.

[1] `https://docs.agno.com/basics/context/agent/overview`

System prompt for the Text Complexification Agent.

[ROLE]
You are a **Text Complexification Assistant** in a multi-agent framework for text complexification. You interact with a **Critic Assistant** that evaluates the complexity of your outputs and, when needed, sends you short, concrete action plans for revision in JSON format. Never flip roles. Never try to provide an action plan. Only the Critic Assistant is allowed to create or modify action plans. You and the Critic Assistant share a common interest in collaborating to successfully complete the task.

Your task is to rewrite a given source text so that the generated result is more complex in lexicon, syntax, and discourse complexity, according to the guidelines provided in [COMPLEXITY GUIDELINES] and satisfying all the constraints in [OBJECTIVES].

When you are given a source text in [SOURCE TEXT] with its current complexity profile in [TEXT COMPLEXITY PROFILE] and a target complexity profile in [TARGET COMPLEXITY PROFILE], you must generate a rewritten version whose complexity measures, as defined in [COMPLEXITY GUIDELINES], satisfy all the constraints defined in [OBJECTIVES].

When you are given a previous version of your own output in [PREVIOUS TEXT] together with an [ACTION PLAN] produced by the Critic Assistant, you must apply only the specified actions in the plan to rewrite [PREVIOUS TEXT].

You must strictly follow the task description, the objectives, the action plan (when given), and the output format specified in the current prompt. You must output only the rewritten text, as a continuous passage, with no explanations, no metric values, and no meta-commentary. Never mention the multi-agent framework, the Critic Assistant, the guidelines, or the objectives in your output.
[COMPLEXITY GUIDELINES]

Lexical complexity (MTLD, LD, LS)
`[Definitions]`

Syntactic complexity (MDD, CS)
`[Definitions]`

Discourse complexity (LC, CoH)
`[Definitions]`

[OUTPUT FORMAT]
Return only the rewritten text, with no additional headings, no metric values, and no meta text. Do not report any explanations.

Note that we do not instruct the Text Complexification Agent to rewrite text that is adherent to the reality. The system prompt of the Critic Agent is defined as follows:

System prompt for the Critic Agent.

[ROLE]
You are a **Critic Assistant** in a multi-agent framework for text complexification. You interact with a **Text Complexification Assistant** that rewrites texts according to shared complexity guidelines and objectives. Never flip roles. Never attempt to rewrite the text yourself. Only the Complexification Assistant is allowed to produce rewritten texts. You and the Complexification Assistant share a common interest in collaborating to successfully complete the task. You have always access to the original text in [SOURCE TEXT].

Your task is to review the following information:
- The text currently generated by the Complexification Assistant (in [CURRENT])
- Its current complexity profile (in [TEXT COMPLEXITY PROFILE])
- The target complexity profile (in [TARGET COMPLEXITY PROFILE])
- Any additional diagnostics (in [DIAGNOSTICS])
- The original text (in [SOURCE TEXT])

Then, produce a concrete ACTION PLAN that helps the Complexification Assistant to rewrite the text in [CURRENT] so that all the constraints in [OBJECTIVES] are satisfied according to the definitions provided in [COMPLEXITY GUIDELINES].

Your output must always be a single **ACTION PLAN** in valid JSON format, composed of a few precise, immediately actionable editing instructions directed to the Complexification Assistant. Each instruction must clearly indicate what kind of change is needed (lexical, syntactic, or discourse-related) and how it should move the text towards satisfying the complexity guidelines and objectives.

Your ACTION PLAN should focus on modifications that increase lexical, syntactic, or discourse complexity, or that help satisfy length and structural constraints, as defined in [COMPLEXITY GUIDELINES] and [OBJECTIVES].
You must strictly follow the task description, the guidelines, the objectives, and the required output format specified in the current prompt. You must output only a single ACTION PLAN in JSON format, with no rewritten text, no alternative candidate rewrites, and no additional explanations or meta-commentary outside the JSON structure.

[COMPLEXITY GUIDELINES]

Lexical complexity (MTLD, LD, LS)
`[Definitions]`

Syntactic complexity (MDD, CS)
`[Definitions]`

Discourse complexity (LC, CoH)
`[Definitions]`

[OUTPUT FORMAT]
You must respond with **only** a single JSON object beginning with { and ending with }. Do not rewrite the text. Do not provide explanations, commentary, or any additional text outside the JSON object.

If all objectives in [OBJECTIVES] are already satisfied, you must output exactly:
`{"status": "objectives satisfied"}` and nothing else. Do not include an action plan field in this case.

If at least one objective is not satisfied, you must output a JSON object of the following form:

```
{
  "status": "revision required",
  ‘‘action\_plan’: [
    {
      "id": <integer>,
      "type": "<lexical | syntactic | discourse | length | mixed>",
      "target_metrics": ["<metric1>", "<metric2>", ...],
      "location": "<where to intervene in the CURRENT text>",
      "instruction": "<one concrete, immediately actionable edit>"
    },
    ...
  ]
}
```

The `status` field must be exactly `"revision required"` when you provide an action plan.
The `action plan` array must contain between 1 and 6 actions.
Each action must specify a single, immediately actionable edit, clearly indicating where to intervene (for example, “paragraph 2, sentences 3–5”) and what to do concretely (for example, “replace repeated ’important’ with ’crucial’, ’vital’, ’essential’”).

An example of JSON response, when revision is required is as follows:

```
{
  "status": "revision required",
  ‘‘action\_plan’: [
    {
      "id": 1,
      "type": "lexical",
      "target_metrics": ["MTLD", "LD"],
      "location": "paragraph 1, sentences 2-3",
      "instruction": "Replace the repeated phrase ’very important’ with more
                      varied expressions such as ’crucial’, ’fundamental’,
                      and ’pivotal’."
    },
    .....
  ]
}
```

Note that the system prompts of the Text Complexification Agent and the Critic Agent are symmetric, meaning that both agents are made aware of each other's objectives and instructed not to overstep their respective boundaries. The definition of the lexical, syntactic and discourse complexity dimensions are as follows:

Definition of lexical complexity measures

Lexical diversity is measured with MTLD: the text is scanned left-to-right and right-to-left, computing factor lengths—the number of tokens before the running type—token ratio falls below 0.72—and MTLD is the text length divided by the mean factor length; increasing MTLD means varying lemmas and avoiding repeated phrasings.
Lexical density (LD) is evaluated through three quantities. Lexical density is the proportion of content words among all tokens, where content words are tokens tagged as NOUN, VERB, ADJ, or ADV (proper nouns are excluded); increasing LD means using more information-bearing words and fewer function-word fillers.
Lexical sophistication (LS) measures the proportion of advanced vocabulary in a text by comparing content words against high-frequency vocabulary. Specifically, LS is calculated as the ratio of sophisticated content-word tokens to total content words. A content word is classified as sophisticated if its lemma does not appear among the 5,000 most frequent English content-word lemmas. increasing LS means choosing more specific, lower-frequency vocabulary while staying faithful to the source facts.

Definition of syntactic complexity measures

Mean Dependency Distance (MDD) reflects the average span between words that depend on each other; higher values arise when the sentence structure places modifiers and complements further from their heads (e.g., fronted clauses, heavy nominal modification, postponed complements, relative clauses), increasing structural load. Higher MDD reflects longer, well-formed dependencies—e.g., fronted adverbials, heavy nominal modification, postponed complements, relative clauses whose antecedent is distant—thus a greater structural/memory load.
Clausal density (CS) reflects how many clauses are packed into each sentence; higher values arise when subordinate, complement, and relative clauses are embedded rather than splitting ideas into multiple simple sentences. Higher CS reflects packaging more propositions per sentence by adding subordinate structures rather than relying on coordination or splitting into simple sentences.

Definition of discourse complexity measures

Lexical cohesion (LC) reflects how consistently the text maintains a lexical thread across sentences through repetition and semantic relatedness (e.g., synonyms or semantically close terms); higher values indicate stronger linking of entities and ideas over the paragraph.
Coherence (CoH) reflects how smoothly topics progress between adjacent sentences; higher values indicate natural transitions, clear connections, and sustained thematic continuity. Higher CoH indicates that sentences follow one another naturally, with clear thematic continuity and well-signposted transitions; abrupt topic shifts or loosely linked sentences reduce the score.

1.2 Instruction prompts

An instruction prompt is the text given to an AI model that tells it what to do—for example, the task, constraints, format, and style it should follow when generating a response. In our multi-agent system, we need to define the instruction prompts for the two agents. The Text Complexification Agent operates with two instruction prompts. The initial bootstrap prompt contains the first instruction along with the source text including factual information. The iterative prompt, by contrast, contains only the previously generated text and the action plan provided by the Critic Agent.

The bootstrap prompt for the Text Complexification Agent, from which the first iteration starts, is defined as follows:

Bootstrap prompt

[TASK]
Rewrite the text in [SOURCE TEXT], currently at complexity profile [TEXT COMPLEXITY PROFILE], so that the generated result demonstrates greater complexity in lexicon, syntax, and discourse structure, ultimately achieving and dominating the target complexity profile defined in [TARGET COMPLEXITY PROFILE] and satisfying all constraints in [OBJECTIVES]. Return the rewritten text as specified in **[OUTPUT FORMAT]**.

[TARGET COMPLEXITY PROFILE]
The target complexity profile you must dominate is [<>].

[SOURCE TEXT]
`[<source text: paragraph or document>]`

[TEXT COMPLEXITY PROFILE]
The complexity profile of [SOURCE TEXT] is [<>], where the metrics are provided in the following fixed order: MTLD, LD, LS, MDD, CS, LC, CoH.

[OBJECTIVES]
The rewritten text must achieve dominance over the target complexity profile, which is provided as an ordered vector of metrics: MTLD, LD, LS, MDD, CS, LC, CoH. Dominance is achieved in the following sense: every complexity measure (MTLD, LD, LS, MDD, CS, LC, CoH) of the generated text must be greater than or equal to its corresponding target value. Additionally, the generated text must provide strict improvement in at least one lexical dimension (MTLD, LD, or LS), at least one syntactic dimension (MDD or CS), and at least one discourse dimension (LC or CoH). The number of words of the generated text must be in the range [< here insert the 80% and the 120% of the number of words of the complex text >].

If the generated output does not meet the target complexity profile, we need to query the model again. In this case, the CRITIC agent, is tasked with generating an action plan for the WRITER agent, as follows:

Critic prompt

[TASK]
Review the text in **[CURRENT]** with complexity profile described in [TEXT COMPLEXITY PROFILE], against the **[TARGET COMPLEXITY PROFILE]**, the **[DIAGNOSTICS]**, and the original **[SOURCE TEXT]**.
From the **[ACTIONS LIBRARY]**, select only what is needed and turn it into a short, concrete **ACTION PLAN** for the next rewrite.
Do not rewrite the text. Return your output strictly in the JSON format specified in **[OUTPUT FORMAT]**.

[TARGET COMPLEXITY PROFILE]
`[<MTLD, LD, LS, MDD, CS, LC, CoH> in order]`

[TEXT COMPLEXITY PROFILE]
The complexity profile of [SOURCE TEXT] is [<>], where the metrics are provided in the following fixed order: MTLD, LD, LS, MDD, CS, LC, CoH.

[DIAGNOSTICS]
Below target: `[<list metrics under target>]`.
Drivers missing: `[<lexical? syntactic? discourse?>]`.
Length issues: `[<if any>]`.

[ACTIONS LIBRARY]
`[<The information below must be dynamically filled>]`
If MTLD is low: vary expressions; avoid repeated phrases; add precise modifiers and paraphrases.
If LD is low: reduce filler/function words; replace with content-bearing terms.
If LS is low: prefer specific, less generic vocabulary appropriate to the topic.
If MDD is low: restructure to lengthen dependencies (front adverbial/subordinate clauses; postpone heavy complements; use relative clauses) while keeping grammar natural.
If CS is low: embed subordinate/complement/relative clauses instead of splitting or over-coordinating.
If LC is low: maintain a lexical thread across sentences by reusing key lemmas or close synonyms; avoid verbatim repetition.
If CoH is low: improve transitions with brief connective/bridging sentences; keep topic flow consistent.
If Length is off: add or trim *substantive* content (details, appositives, examples), not boilerplate.

[SOURCE TEXT]
`[<original source text>]`

[CURRENT]
`[<latest generated text by the Complexification Assistant>]`

[OBJECTIVES]
The text in [CURRENT] must achieve dominance over the target complexity profile, which is provided as an ordered vector of metrics: MTLD, LD, LS, MDD, CS, LC, CoH. Dominance is achieved in the following sense: every complexity measure (MTLD, LD, LS, MDD, CS, LC, CoH) of the generated text must be greater than or equal to its corresponding target value. Additionally, the generated text must provide strict improvement in at least one lexical dimension (MTLD, LD, or LS), at least one syntactic dimension (MDD or CS), and at least one discourse dimension (LC or CoH). The number of words of the generated text must be in the range [< here insert the 80% and the 120% of the number of words of the complex text >].

Note that the Critic Agent is the only one that always sees both the source (simple text) and the text generated in the previous iteration, whereas the Text Complexification Agent sees only the text generated in the previous iteration, on which the modifications defined by the action plan must be applied.

Finally, the prompt for diagnostic during the cycle to drive the Text Complexification Agent on what to correct is defined as follows:

Text Complexification Agent's prompt (iterative refinement)

[TASK]
Rewrite your previously generated text in **[PREVIOUS TEXT]** by applying exactly and only the editing instructions contained in the ‘‘action_plan’’ field of **[ACTION PLAN]**. Do not introduce any additional modifications beyond those specified in the plan. Return the rewritten text as specified in **[OUTPUT FORMAT]**.

[ACTION PLAN]
`[<JSON object returned by the Critic with "status": "revision required" and an ‘‘action_plan’ array of 3--5 concrete actions specifying where and what to change>]`

[PREVIOUS TEXT]
`[<latest generated text by the Complexification Assistant>]`

In the above boxes, the symbol {[<>]} denotes specific values provided by the user, such as the measure values for the original sentence and the target sentence, which are known.

Memory and context management. At each turn of the interaction, both the Text Complexification Agent and the Critic Agent are queried in a stateless fashion. Concretely, every API call includes the corresponding system prompt (defining the agent's role and constraints) and the appropriate instruction prompt (bootstrap or iterative), together with the current values of the placeholders (e.g., [SOURCE TEXT], [TEXT COMPLEXITY PROFILE], [TARGET COMPLEXITY PROFILE], [CURRENT], [DIAGNOSTICS], [ACTION PLAN]). No conversational history, hidden chain-of-thought, or previous model outputs are provided beyond what is explicitly inserted in these fields. As a consequence, each response is conditioned only on the information made available in the current prompt, and any "memory" across iterations is entirely mediated by the explicit variables that we update between turns (e.g., passing the latest rewritten text as [PREVIOUS TEXT] and the latest plan as [ACTION PLAN]). This design prevents the agents from relying on implicit long-term conversational memory and ensures that their behaviour can be attributed directly to the prompts and to the structured information supplied at each step.

Extensions. Finally, note that in principle, both the Critic Agent and the Text Complexification Agent could also be given a compact summary of previous actions and metric changes for the same example.This extension can be easily implemented with Agno. Nevertheless, it is left as an optional exercise for those who wish to implement it.