

Nel testo **non è presente alcun riferimento** a costrutti architetturali quali *system prompt* (intesi come istruzioni di sistema persistenti a livello di piattaforma o API), *Personas* stabili, *Gems* di Gemini, *Custom GPTs* o *Skill/agenti* estendibili.

Il volume assolutizza in modo rigido la modalità della **chat lineare in interfaccia web** (con modelli commerciali quali Claude, ChatGPT, Gemini, Copilot). L'assegnazione di un "ruolo" o la definizione di un contesto non vengono concepite come configurazioni di sistema, ma unicamente come clausole aggiuntive inserite dall'utente all'interno del singolo testo di richiesta (Appendice 3, "Versione C — prompt con ruolo e istruzioni" e Tecnica 7).

La gestione della complessità e della memoria nell'interazione non avviene tramite agenti o personalizzazioni dell'ambiente, bensì attraverso un **workflow artigianale di concatenazione manuale ("usa e getta")**:

- L'utente sottomette un prompt singolo nella finestra di dialogo.
- Ottiene un testo di output intermedio (denominato $\backslash(D_1, D_2 \dots D_7\backslash)$ o $\backslash(T_1 \dots T_6\backslash)$).
- Copia, salva o riutilizza manualmente tale output intermedio come materiale di partenza (input) per il prompt successivo.

L'autore teorizza esplicitamente questa limitazione, evidenziando come tutti i parametri di controllo e configurazione dell'infrastruttura (temperatura, *top-p*, *beam search*, istruzioni di sistema) rimangano invisibili e inaccessibili all'utente comune nella versione standard delle chat, riducendo l'intera operatività alla sola formulazione del testo di input.

Nota filologica: Si riscontrano nel testo analizzato ulteriori errori di scansione/digitazione, tra cui la corruzione di termini con inserimento di maiuscole improprie (es. "*gennIAo*" per "*gennaio*"; "*IAuta*" per "*aiuta*") e sviste sintattiche nell'appendice metodologica ("*oridurre*" in luogo di "*produrre*").

Note Critiche & Pattern Ricorrenti

- **Arretratezza Architetturale del Modello di Prompting**: L'approccio del testo soffre di un marcato **riduzionismo operativo**: trattando l'interazione come una sequenza di scambi manuali in chat singola, ignora le architetture moderne di *prompt engineering* (system instructions, memoria a lungo termine, RAG personalizzati, agenti autonomi). Questo genera una sovrastruttura procedurale pesante (fino a 7 passaggi manuali di copy-paste di documenti intermedi $\backslash(D_n\backslash)$) che vanifica l'efficienza dell'automazione.
- **Cliché dell'Utente-Progettista**: Si individua il topos della saggistica divulgativa che scarica interamente sull'utente la responsabilità del fallimento del modello. Se l'IA allucina o risponde in modo viziato, la colpa viene attribuita alla mancanza di rigore nella domanda dell'utente, occultando i limiti strutturali e le opacità degli algoritmi proprietari.
- **Semplificazione dell'Interfaccia Utente**: L'ipotesi che l'utente debba vicariare la mancanza di parametri di sistema (temperatura, top-p) aggiungendo clausole testuali come "*rispondi in modo letterale*" costituisce un'illusione di controllo, poiché tali indicazioni nel prompt utente non modificano i campionamenti stocastici del modello alla radice.

Glossario Plain Language

Termine Tecnico / Burocratico	Spiegazione in Linguaggio Piano
System Prompt	Le istruzioni di fondo date allo strumento prima che l'utente inizi a scrivere, per definirne comportamento e limiti costanti.
Persona / Gem / Custom GPT	Versioni personalizzate e salvate della macchina, create per svolgere sempre un ruolo specifico senza doverglielo rispiegare ogni volta.
Chat "Usa e Getta"	L'uso della finestra di dialogo standard in cui ogni conversazione riparte da zero e non conserva memoria delle sessioni precedenti.
Testo Intermedio (\D_n\)/\T_n\)	La risposta ottenuta da un passaggio che viene usata dall'utente come testo di partenza per la domanda successiva.
Temperatura / Top-p	I comandi tecnici interni che regolano quanto la macchina debba essere precisa e prevedibile oppure vaga e creativa.