

La grammatica inganna

Perché i prompt efficaci non sono temi scritti bene

Indice

Introduzione - La buona scrittura non basta

1. L'errore di partenza: trattare l'AI come un lettore umano
2. Il prompt come specifica operativa
3. Perché la grammatica perfetta può peggiorare un prompt
4. Cosa mostrano le ricerche sul prompt strutturato
5. Grammatica utile e grammatica ornamentale
6. Prompt scarno non significa prompt povero
7. Il metodo: da desiderio a istruzione eseguibile
8. L'italiano come rischio operativo
9. Prompt engineering per agenti e automazioni
10. Esercizi e laboratori comparativi

Toolkit operativo

Schede pratiche di trasformazione

Rubrica di valutazione dei prompt

Errori ricorrenti da correggere

Glossario operativo

Conclusione

Nota bibliografica di lavoro

Introduzione - La buona scrittura non basta

Questo manuale nasce da una tesi semplice, ma spesso controintuitiva: un prompt scritto in italiano elegante, corretto e scorrevole non è automaticamente un buon prompt. Nei contesti scolastici, universitari e professionali siamo abituati a considerare la qualità di un testo attraverso la grammatica, la ricchezza lessicale, la coesione del periodo e la cura dello stile. Questa abitudine diventa però fuorviante quando viene trasferita in modo ingenuo all'interazione con i modelli linguistici generativi.

Un LLM non valuta un prompt come un docente valuta un tema. Non assegna un premio alla bella forma. Non migliora la risposta perché l'utente dimostra padronanza del congiuntivo, usa subordinate raffinate o inserisce formule di cortesia elaborate. Il modello riceve una sequenza di token e cerca di generare una continuazione coerente con istruzioni, vincoli, esempi, formato richiesto e contesto. Se questi elementi sono vaghi, nascosti o distribuiti dentro una prosa elegante ma poco governabile, il risultato può peggiorare.

La tesi corretta, quindi, non è che la grammatica sia inutile o dannosa in se. Sarebbe una caricatura. La grammatica è utile quando rende l'istruzione più chiara, elimina ambiguità, separa le relazioni logiche e rende più leggibile l'input. Diventa invece deprecabile quando viene trattata come ornamento, come dimostrazione di cultura o come sostituto della specifica. Un prompt può essere formalmente impeccabile e operativamente debole. Può suonare bene a un essere umano e funzionare male per un modello.

Il punto decisivo è spostare l'attenzione dalla bella frase alla specifica eseguibile. Il prompt efficace non è un piccolo saggio. È più vicino a una scheda tecnica, a un brief operativo, a una funzione con parametri, a un contratto tra utente e modello. Deve dire che cosa fare, con quali dati, entro quali limiti, secondo quali criteri e in quale forma restituire il risultato.

Questo manuale è pensato per formatori, AI officer, consulenti, docenti, manager e professionisti che vogliono insegnare o usare l'AI generativa in modo più affidabile. L'obiettivo non è sostituire la lingua naturale con codice puro. L'obiettivo è imparare a usare la lingua naturale come interfaccia di controllo: meno retorica, più

struttura; meno implicito, più vincoli; meno desideri generici, più istruzioni verificabili.

1. L'errore di partenza: trattare l'AI come un lettore umano

Il primo errore nel prompting nasce dall'antropomorfizzazione. I modelli linguistici rispondono con frasi naturali, tono conversazionale e apparente comprensione. Questo porta molti utenti a comportarsi come se davanti avessero una persona competente da convincere, impressionare o compiacere. Il prompt viene scritto come una richiesta cortese a un esperto umano: periodi ampi, mitigazioni, premesse, scuse, formule di gentilezza, spiegazioni laterali e dettagli narrativi.

Questa modalità è comprensibile. La conversazione è l'interfaccia più naturale che abbiamo. Ma è anche il motivo per cui molti prompt falliscono. L'AI non ha bisogno di essere rassicurata, sedotta o preparata psicologicamente. Ha bisogno di ricevere segnali chiari. Quando l'utente scrive 'Saresti così gentile da aiutarmi a realizzare, se possibile, una sintesi che tenga conto dei principali elementi emersi nel testo seguente?', sta producendo un testo socialmente educato, ma non necessariamente una buona istruzione.

Il modello può comunque rispondere, perché gli LLM sono robusti e tollerano richieste imperfette. Ma tollerare non significa ottimizzare. Il modello dovrà dedurre da solo il pubblico, la lunghezza, il grado di dettaglio, il formato, il tipo di sintesi, la presenza o assenza di esempi, il livello tecnico e il criterio con cui distinguere il rilevante dal marginale. Una frase gentile nasconde molte decisioni operative.

La differenza è evidente se confrontiamo due richieste sullo stesso compito.

Prompt letterario:

Vorrei che tu mi aiutassi a rielaborare il testo seguente in modo efficace, mantenendo un buon livello stilistico e cercando di renderlo più chiaro e più adatto a un pubblico non specialista, senza tuttavia snaturarne il contenuto complessivo.

Prompt operativo:

Obiettivo: riscrivere il testo per un pubblico non specialista.

Vincoli:

- mantieni la tesi originale;
 - elimina gergo tecnico non necessario;
 - frasi massimo 22 parole;
 - tono diretto, non promozionale;
 - massimo 1.500 caratteri.
- Output: restituisci solo la versione riscritta.

Il primo prompt è più vicino alla buona prosa. Il secondo è più vicino a una specifica. Nel secondo caso il modello riceve un obiettivo, una serie di vincoli e un formato di ritorno. Non deve indovinare cosa significhi 'efficace', 'buon livello stilistico' o 'adatto'. Gli elementi qualitativi vengono tradotti in regole operative.

2. Il prompt come specifica operativa

Un prompt efficace non è la domanda più bella. È la descrizione più controllabile del risultato desiderato. Per questo il prompting professionale assomiglia sempre meno alla composizione di un testo e sempre più alla progettazione di una specifica. Una specifica non cerca di essere elegante; cerca di essere eseguibile.

Pensare al prompt come specifica significa separare i componenti. Ogni richiesta complessa contiene almeno sei elementi: ruolo, obiettivo, contesto, input, vincoli e output. Quando questi elementi vengono fusi in un unico paragrafo, il modello deve ricostruire la struttura nascosta. Quando vengono etichettati, la struttura è già disponibile.

Struttura base:

Ruolo: chi deve essere il modello in questa interazione.

Task: quale operazione deve eseguire.

Contesto: quali informazioni deve considerare.

Vincoli: cosa deve rispettare o evitare.

Processo: quali passaggi deve seguire, se necessari.

Output: in quale formato deve restituire il risultato.

Questa struttura non impoverisce la lingua. La rende funzionale. La prosa resta utile per spiegare il contesto, ma non deve nascondere l'azione. In un buon prompt ogni blocco ha una responsabilità. Il ruolo orienta il dominio. Il task definisce l'azione. Il

contesto delimita il campo. I vincoli riducono lo spazio delle soluzioni. Il formato rende il risultato utilizzabile.

Il valore di questo approccio emerge soprattutto nei processi ripetibili: analisi di documenti, generazione di report, revisione di testi, classificazione di dati, creazione di contenuti editoriali, supporto a decisioni aziendali, agenti e automazioni. In questi casi non basta ottenere una risposta piacevole una volta. Serve produrre output coerenti, confrontabili e verificabili nel tempo.

Il prompt come specifica permette anche di fare qualcosa che la buona prosa non consente: testare. Se un output non rispetta il formato, si può correggere il blocco 'Output'. Se il tono è sbagliato, si può correggere il blocco 'Vincoli'. Se il modello interpreta male il compito, si può correggere il blocco 'Task'. Un prompt monolitico, invece, è più difficile da diagnosticare: non si capisce quale parte abbia generato l'errore.

3. Perché la grammatica perfetta può peggiorare un prompt

La grammatica perfetta peggiora un prompt quando aumenta la distanza tra intenzione e istruzione. Questo accade in almeno cinque modi: prolissità, ambiguità, dispersione dei vincoli, rumore conversazionale e falsa completezza.

La prolissità aumenta il numero di token senza aumentare l'informazione utile. Un prompt lungo non è automaticamente più ricco. Spesso è solo più costoso e più difficile da seguire. Se molte parole servono soltanto a rendere il periodo più elegante, il rapporto tra informazione e rumore peggiora.

L'ambiguità nasce quando parole apparentemente raffinate sostituiscono criteri concreti. Espressioni come 'migliora lo stile', 'rendilo più professionale', 'valorizza il contenuto', 'mantieni un registro adeguato' possono essere corrette in italiano, ma restano deboli come istruzioni. Che cosa significa professionale? Più tecnico, più sobrio, più sintetico, più formale, più commerciale? Senza criteri, il modello sceglie.

La dispersione dei vincoli si verifica quando le regole importanti sono sepolte dentro subordinate o incisi. Un vincolo deve essere visibile. Se l'utente scrive una frase lunga

in cui il limite di lunghezza, il pubblico e il formato sono inseriti a meta periodo, il modello può trattarli come segnali secondari. Un vincolo isolato in elenco puntato e più forte di un vincolo nascosto in una bella frase.

Il rumore conversazionale e composto da formule di cortesia, meta-commenti e attenuazioni: 'se non ti dispiace', 'ti chiedo cortesemente', 'sarebbe opportuno', 'nel caso fosse possibile', 'mi piacerebbe che'. Queste espressioni hanno valore sociale tra umani, ma di solito non aggiungono informazione operativa per il modello. In alcuni compiti possono spingere l'output verso preamboli, cautela eccessiva o tono discorsivo.

La falsa completezza e il problema più insidioso. Un prompt può sembrare completo perché è lungo e ben scritto, ma mancare dei dati essenziali: pubblico, scopo, formato, criteri di successo, vincoli negativi. L'utente ha l'impressione di aver spiegato molto; il modello riceve invece un testo narrativo con poche istruzioni controllabili.

Per questo una grammatica impeccabile non salva un prompt sottospecificato. Può perfino mascherarne la debolezza. La frase suona bene, quindi sembra buona. Ma un prompt non va giudicato da come suona. Va giudicato da quanto riduce le decisioni che il modello deve prendere al posto dell'utente.

4. Cosa mostrano le ricerche sul prompt strutturato

Le ricerche fornite convergono su una distinzione essenziale: la qualità del prompt dipende più dalla struttura informativa che dalla raffinatezza grammaticale. Nei materiali analizzati ricorrono alcuni filoni: prompt strutturati con checklist, formati semi-strutturati, input parametrici, pseudocodice, XML, Markdown, studi sulla cortesia, studi sulla sensibilità lessicale e analisi del rapporto tra densità semantica e accuratezza.

Il dato più importante non è che esista un unico formato magico. JSON, YAML, Markdown, XML, liste puntate, tabelle e pseudocodice funzionano in modi diversi a seconda del compito. Il punto è che tutti questi approcci obbligano l'utente a separare le parti del problema. Rendono espliciti campi, regole, output, priorità e

criteri. La struttura batte la vaghezza non perché sia più rigida in astratto, ma perché riduce l'ambiguità operativa.

Le ricerche sui prompt per dati strutturati mostrano che la forma logica dell'input può incidere su accuratezza, costo e latenza. A parità di contenuto informativo, cambiare il modo in cui i campi vengono presentati al modello può produrre risposte più stabili o più economiche. Questa osservazione è decisiva: la variabile non è la bellezza della frase, ma l'architettura dell'informazione.

Le ricerche su semi-structured natural language indicano una direzione pragmatica: non serve trasformare ogni prompt in codice puro. Spesso il miglior compromesso è un linguaggio naturale breve, ma organizzato in campi. Il contesto può essere spiegato in frasi semplici; i vincoli e l'output devono invece essere dichiarati con precisione. Questo approccio è particolarmente adatto alla formazione, perché non richiede competenze di programmazione, ma insegna la mentalità della specifica.

Gli studi sulla sottospecificazione aggiungono una cautela importante: non bisogna confondere struttura con accumulo indiscriminato di dettagli. Un prompt iper-regolato può creare vincoli concorrenti, appesantire il compito o introdurre rumore. L'obiettivo non è scrivere prompt lunghissimi. L'obiettivo è massimizzare la densità informativa: ogni parola deve aggiungere contesto, vincolo, esempio o criterio.

Le ricerche sulla cortesia e sul tono sono utili per la didattica. Non invitano a essere aggressivi con il modello; invitano a distinguere la cortesia umana dalla precisione macchina. Dire 'per favore' non è un problema. Il problema è costruire un'intera richiesta intorno a formule sociali invece che intorno a istruzioni. Una comunicazione rispettosa può essere anche asciutta e diretta.

Infine, le ricerche su pseudocodice, Markdown e tag XML mostrano perché i modelli rispondano bene a strutture vicine al codice: durante l'addestramento hanno assorbito enormi quantità di testo tecnico, repository, markup, documentazione, esempi strutturati e codice sorgente. I modelli riconoscono pattern gerarchici, campi, delimitatori e formati. Usare questi segnali non significa parlare male; significa parlare in modo più leggibile per il sistema.

5. Grammatica utile e grammatica ornamentale

Per evitare equivoci, bisogna distinguere grammatica utile e grammatica ornamentale. La grammatica utile rende l'istruzione più chiara. La grammatica ornamentale rende il testo più elegante senza migliorare il controllo del risultato.

La grammatica utile serve a separare soggetto, azione, oggetto e condizione. È utile quando evita doppie negazioni, chiarisce relazioni temporali, distingue obblighi da preferenze, riduce pronomi ambigui e rende esplicito a che cosa si riferisce un vincolo. In questo senso, scrivere bene aiuta. Un prompt sgrammaticato, contraddittorio o confuso può certamente peggiorare l'output.

La grammatica ornamentale, invece, aggiunge variazione sintattica, perifrasi, subordinate e sfumature che non servono al compito. In un saggio, queste risorse possono migliorare ritmo e stile. In un prompt, spesso aumentano il rumore. La macchina non ha bisogno di un periodo elegante; ha bisogno di sapere cosa deve fare.

Da evitare:

Mi piacerebbe che tu potessi, mantenendo un certo equilibrio tra completezza informativa e leggibilità generale, produrre un'analisi che sappia valorizzare gli elementi principali del testo senza perdersi in aspetti secondari.

Da preferire:

Analizza il testo.

Output:

1. Tesi principale, massimo 3 righe.
2. Cinque elementi chiave.
3. Tre aspetti secondari da ignorare.
4. Valutazione finale, massimo 120 parole.

La seconda versione non è più povera. È più esplicita. Trasforma 'equilibrio', 'leggibilità', 'elementi principali' e 'aspetti secondari' in componenti osservabili. Il modello sa cosa restituire e l'utente può controllare se lo ha fatto.

Una regola pratica: se una parola non cambia il comportamento atteso del modello, probabilmente è decorativa. Se invece specifica un criterio, un limite, una procedura, un destinatario, una priorità o un formato, allora è utile.

6. Prompt scarno non significa prompt povero

Il prompt scarno è spesso frainteso. Scarno non significa approssimativo. Significa privo di ornamento. Un prompt scarno può essere altamente sofisticato se contiene gli elementi giusti. La qualità non sta nella lunghezza, ma nella precisione della selezione.

Un prompt povero dice: 'Fammi un post interessante su questo tema'. Un prompt scarno ma forte dice: 'Scrivi un post LinkedIn per imprenditori. Tesi nelle prime due righe. Tono critico, non motivazionale. Massimo 1.800 caratteri. Chiudi con una domanda operativa. Non usare emoji. Restituisci solo il post'. Il secondo è breve, ma contiene pubblico, canale, struttura, tono, lunghezza, vincoli negativi e formato.

La differenza tra povertà e densità è cruciale. Il prompt povero lascia tutto al modello. Il prompt denso lascia al modello lo spazio creativo necessario, ma controlla le decisioni che contano. Non dice ogni singola parola da usare, ma stabilisce il perimetro entro cui generare.

In molti casi, il prompt migliore ha una forma quasi asciutta:

Ruolo: editor tecnico.

Task: rendi il testo più chiaro senza cambiarne la tesi.

Pubblico: manager non tecnici.

Vincoli: frasi brevi; niente gergo; nessuna metafora; massimo 900 parole.

Output: titolo, sottotitolo, testo finale.

Questa struttura appare semplice, ma contiene un pensiero progettuale. L'utente ha deciso chi deve parlare, che cosa deve fare, per chi, con quali limiti e in quale forma. Il modello non viene lasciato libero di interpretare l'intenzione generale. Viene guidato.

La scarsità di parole diventa un vantaggio solo quando è il risultato di una selezione. Tagliare senza progettare produce prompt deboli. Progettare e poi tagliare produce prompt forti.

7. Il metodo: da desiderio a istruzione eseguibile

Molti prompt falliscono perché partono da un desiderio e restano un desiderio. 'Voglio un testo migliore', 'voglio un riassunto chiaro', 'voglio una strategia efficace', 'voglio una lezione coinvolgente'. Tutte queste formulazioni esprimono un'intenzione, ma non ancora una specifica.

Il lavoro del prompt engineer consiste nel trasformare il desiderio in istruzione eseguibile. Questo passaggio può essere insegnato con una sequenza fissa.

1. Desiderio: che cosa voglio ottenere?
2. Risultato osservabile: come deve apparire l'output?
3. Destinatario: per chi e prodotto?
4. Dati disponibili: quali input deve usare?
5. Vincoli: che cosa deve rispettare?
6. Esclusioni: che cosa deve evitare?
7. Criteri di successo: come valuto se va bene?
8. Formato: come deve restituire il risultato?

Questa sequenza costringe l'utente a pensare prima di scrivere. È proprio qui avviene il salto di qualità. Il prompt non è un atto linguistico spontaneo; è un piccolo progetto. Prima si chiarisce il problema, poi si scrive la richiesta.

Un framework semplice è RTF: Role, Task, Format. È utile per iniziare, perché spinge a dichiarare chi deve essere il modello, quale compito deve svolgere e quale output deve produrre. Per usi professionali conviene estenderlo a TCOF: Task, Context, Output, Format. Nei processi complessi si può aggiungere una sezione di vincoli, esempi e criteri di verifica.

Template TCOF esteso:

Task: [azione precisa]

Context: [informazioni da considerare]

Input: [testo, dati o materiali]

Constraints: [limiti obbligatori]

Do not: [errori o comportamenti da evitare]

Output format: [struttura richiesta]

Quality check: [criteri di verifica]

Il valore di questi framework non sta nel nome. Sta nella disciplina. Ogni volta che l'utente compila i campi, è costretto a rendere esplicito ciò che normalmente darebbe per scontato. Questa è la vera competenza del prompting: non parlare meglio, ma pensare meglio.

8. L'italiano come rischio operativo

L'italiano è una lingua ricca, flessibile e adatta alla sfumatura. Queste qualità, nella scrittura umana, sono preziose. Nel prompting possono diventare un rischio operativo. La nostra lingua consente periodi lunghi, incisi, pronomi sottintesi, subordinate, registri formali, nominalizzazioni e costruzioni implicite. Tutti questi elementi possono aumentare l'ambiguità per un modello.

Un problema frequente è la nominalizzazione. Invece di scrivere 'riscrivi il testo', l'utente scrive 'procedi a una rielaborazione del contenuto'. Invece di 'elimina le ripetizioni', scrive 'opera un intervento di riduzione delle ridondanze espressive'. Le versioni nominalizzate sembrano più professionali, ma spesso sono meno dirette.

Un altro problema è l'uso di verbi attenuati: 'potresti', 'sarebbe utile', 'si potrebbe valutare', 'proverei a'. Tra umani queste forme sono cortesi. Nel prompt possono indebolire la distinzione tra obbligo e preferenza. Se un vincolo è obbligatorio, va scritto come obbligo.

Debole:

Sarebbe preferibile, se possibile, evitare un tono troppo promozionale.

Forte:

Vincolo obbligatorio: non usare tono promozionale.

Anche le subordinate possono creare problemi quando contengono troppe istruzioni. Un periodo italiano può tenere insieme molte relazioni logiche, ma il modello può non attribuire lo stesso peso a tutte. Nella pratica conviene spezzare. Una riga, una funzione: azione, vincolo, output, esclusione.

La soluzione non è scrivere in un italiano brutto. È scrivere in un italiano operativo: frasi brevi, verbi attivi, sostantivi concreti, elenchi, sezioni, delimitatori. L'italiano resta corretto, ma non cerca di fare bella figura. Cerca di far funzionare il modello.

9. Prompt engineering per agenti e automazioni

Quando si passa dalla chat casuale agli agenti e ai processi di automazione, la tesi diventa ancora più forte. Un agente non ha bisogno di una richiesta elegante. Ha bisogno di istruzioni stabili, verificabili, modulari e aggiornabili. Deve sapere cosa fare, quando fermarsi, quali strumenti usare, quali errori evitare e come restituire lo stato del lavoro.

Nei sistemi agentici, un prompt troppo letterario e fragile. Non si presta al versionamento, non è facile da testare, non permette di isolare le responsabilità. Un prompt strutturato, invece, può essere trattato come un asset operativo: si salva, si confronta, si migliora, si documenta, si riusa.

Esempio di istruzione per agente:

Missione: analizzare documenti DOCX e generare una sintesi editoriale.

Input: cartella con file sorgente.

Passi:

1. estrai testo;
2. identifica tesi, dati, esempi e fonti;
3. segnala contraddizioni;
4. genera struttura del manuale;
5. produce DOCX finale.

Vincoli:

- non inventare fonti;
- separa fatti da interpretazioni;
- usa tono didattico professionale;
- restituisci output in formato Word.

Criterio di completamento: il documento finale contiene titolo, indice, capitoli, esempi e nota bibliografica.

Questa impostazione è più lunga di una semplice frase, ma è meno ambigua. Ogni riga ha funzione operativa. L'agente può eseguire il lavoro per fasi, e l'utente può controllare se ogni fase è stata rispettata. La struttura è anche riutilizzabile: cambiando input e vincoli, lo stesso schema può servire per altri manuali.

Per un AI officer, questo è il punto centrale. La qualità del prompting non va misurata solo sulla singola risposta brillante. Va misurata sulla capacità di costruire

processi replicabili. Un prompt scritto come specifica può diventare parte di una procedura aziendale. Un prompt scritto come tema resta un testo isolato.

10. Esercizi e laboratori comparativi

Il modo più efficace per insegnare questa idea non è spiegarla in astratto, ma farla vedere. Un laboratorio comparativo mette gli studenti o i professionisti davanti a due prompt sullo stesso compito: uno grammaticalmente elegante ma poco strutturato, l'altro scarso ma specifico. Si eseguono entrambi, si confrontano gli output e si misura la differenza.

La valutazione deve essere concreta. Non basta chiedere quale risposta piace di più. Bisogna definire criteri: rispetto del formato, aderenza al pubblico, completezza, assenza di informazioni inventate, lunghezza, chiarezza, stabilità tra più esecuzioni. In questo modo il prompting smette di essere gusto personale e diventa pratica sperimentale.

Esercizio 1: riscrittura

Prompt A: scrivi una richiesta elegante e discorsiva.

Prompt B: scrivi una richiesta con ruolo, task, vincoli e output.

Valuta: quale rispetta meglio lunghezza, tono e pubblico?

Esercizio 2: sintesi

Prompt A: 'fammi un bel riassunto'.

Prompt B: definisci pubblico, massimo parole, struttura e divieti.

Valuta: quale output è più riutilizzabile?

Esercizio 3: analisi dati

Prompt A: chiedi un commento generale.

Prompt B: chiedi tabella con colonne obbligatorie e criteri.

Valuta: quale output è più verificabile?

Un buon laboratorio dovrebbe includere anche la fase di debugging del prompt. Quando l'output non va bene, gli studenti devono individuare quale blocco manca o quale vincolo è ambiguo. Questa pratica sviluppa una competenza molto più utile della semplice capacità di scrivere frasi eleganti: la capacità di diagnosticare una specifica.

L'esercizio finale può essere una trasformazione: partire da un prompt letterario e convertirlo in prompt operativo. L'obiettivo non è accorciarlo a tutti i costi, ma aumentare il rapporto tra parole utili e parole totali.

Toolkit operativo

Template 1 - Prompt editoriale

Ruolo: editor professionale.

Task: migliora il testo fornito.

Pubblico: [specificare destinatario].

Obiettivo: [chiarezza, autorevolezza, sintesi, conversione, didattica].

Vincoli:

- mantieni la tesi;
 - non aggiungere dati non presenti;
 - elimina ripetizioni;
 - usa frasi brevi;
 - massimo [x] parole.
- Output: restituisci solo il testo finale.

Template 2 - Prompt di analisi

Ruolo: analista.

Task: analizza il contenuto seguente.

Output:

1. Tesi principale.
2. Cinque evidenze.
3. Tre criticità.
4. Domande aperte.
5. Raccomandazione operativa.

Vincoli:

- separa fatti e interpretazioni;
- non inventare fonti;
- cita solo elementi presenti nell'input.

Template 3 - Prompt per agenti

Missione: [risultato finale].

Input disponibili: [file, testi, dati].

Strumenti ammessi: [strumenti].

Passi obbligatori:

1. [fase];

2. [fase];

3. [fase].

Vincoli:

- non modificare file sorgente;

- salva output in [formato];

- segnala incertezze.

Fine lavoro: [criterio verificabile].

Checklist di densità del prompt

Prima di inviare un prompt, controlla:

- ho scritto il task in una frase chiara?
- ho indicato pubblico o destinatario?
- ho separato contesto e istruzioni?
- ho dichiarato vincoli obbligatori?
- ho indicato cosa evitare?
- ho richiesto un formato preciso?
- ogni parola aggiunge informazione utile?
- posso verificare oggettivamente l'output?

Schede pratiche di trasformazione

La parte più utile della formazione sul prompting non è spiegare agli utenti che devono essere più precisi. Quasi tutti accettano questa idea in astratto. Il passaggio difficile è mostrare come si passa da una richiesta naturale, spesso educata e ben scritta, a una specifica utilizzabile. Le schede seguenti servono proprio a questo: ogni esempio prende un prompt grammaticalmente corretto ma debole e lo trasforma in una versione operativa.

Scheda 1 - Scrittura LinkedIn

Prompt debole:

Vorrei un post LinkedIn ben scritto, coinvolgente e professionale, che parli dell'importanza dell'intelligenza artificiale nelle aziende e che possa risultare interessante per imprenditori e manager.

Prompt operativo:

Ruolo: ghostwriter LinkedIn B2B.

Task: scrivi un post sul ruolo dell'AI nelle PMI.

Pubblico: imprenditori e manager non tecnici.

Tesi: l'AI non va adottata come moda, ma come processo operativo.

Vincoli:

- massimo 1.600 caratteri;
- tono diretto, non motivazionale;
- niente emoji;
- niente frasi generiche su innovazione e futuro;
- apri con una frase forte.

Output: solo il post finale.

La versione debole usa parole positive ma non operative: ben scritto, coinvolgente, professionale, interessante. La versione forte traduce quelle qualità in elementi controllabili: pubblico, tesi, tono, lunghezza, divieti e formato.

Scheda 2 - Sintesi didattica

Prompt debole:

Mi fai un riassunto chiaro di questo testo, mantenendo le parti più importanti e rendendolo adatto a degli studenti?

Prompt operativo:

Task: riassumi il testo per studenti di quinta superiore.

Obiettivo: far capire tesi, passaggi logici e parole chiave.

Vincoli:

- massimo 280 parole;
- linguaggio semplice ma non infantile;
- non aggiungere informazioni esterne;
- mantieni nomi propri e dati numerici;
- evidenzia eventuali concetti difficili.

Output:

1. Tesi principale.
2. Cinque punti chiave.
3. Glossario di massimo cinque termini.
4. Riassunto finale.

In questa trasformazione il miglioramento non deriva da una lingua più raffinata. Deriva dalla definizione di destinatario, obiettivo, limite, divieti e struttura dell'output. Lo studente riceve un materiale più utilizzabile perché il modello è stato guidato prima della generazione.

Scheda 3 - Analisi di documento aziendale

Prompt debole:

Analizza questo documento e dimmi se va bene, segnalando eventuali problemi e suggerendo miglioramenti.

Prompt operativo:

Ruolo: consulente operativo.

Task: analizza il documento allegato.

Criteri di analisi:

- chiarezza dell'obiettivo;
- coerenza tra sezioni;
- rischi o punti ambigui;
- informazioni mancanti;
- azioni successive.

Output: tabella con colonne Area, Problema, Impatto, Correzione proposta, Priorità.

Vincoli:

- non riscrivere tutto il documento;
- non inventare dati;
- cita solo punti presenti nel testo.

Il prompt debole chiede un giudizio generale. Il prompt operativo definisce il tipo di analisi e obbliga il modello a produrre una tabella azionabile. Questo è un cambio di natura: da opinione testuale a strumento di lavoro.

Scheda 4 - Estrazione dati

Prompt debole:

Leggi il testo e tira fuori le informazioni importanti.

Prompt operativo:

Task: estrai dati dal testo.

Campi obbligatori:

- nome soggetto;
- data;
- importo;
- scadenza;
- obbligo richiesto;
- riferimento normativo o documento, se presente.

Output: JSON valido.

Vincoli:

- se un campo non è presente, usa null;
- non dedurre valori impliciti;
- non aggiungere spiegazioni fuori dal JSON.

Qui la differenza è ancora più netta. La richiesta 'informazioni importanti' lascia al modello la definizione di importanza. La specifica elenca i campi e definisce cosa fare quando un dato manca. Questa è la logica da usare quando l'output deve entrare in un flusso di automazione.

Scheda 5 - Revisione di prompt

Prompt debole:

Migliora questo prompt e rendilo più efficace.

Prompt operativo:

Ruolo: prompt engineer.

Task: revisiona il prompt seguente.

Analizza:

1. obiettivo;
2. input richiesti;
3. vincoli mancanti;
4. formato di output;
5. ambiguità;

6. rischi di allucinazione.

Output:

- diagnosi in massimo 8 righe;
- prompt riscritto;
- tre motivi per cui la nuova versione è più controllabile.

Anche quando si chiede di migliorare un prompt, il miglioramento va specificato. 'Piu efficace' è troppo generico. Una buona revisione deve spiegare quali componenti mancano e perché la nuova versione riduce l'ambiguità.

Rubrica di valutazione dei prompt

Per rendere il prompting una competenza insegnabile, serve una rubrica di valutazione. Senza criteri condivisi, il giudizio resta soggettivo: un prompt piace perché suona bene o perché produce una risposta gradevole. Con una rubrica, invece, si può discutere la qualità del prompt in modo tecnico.

Scala 0-3 per ciascun criterio:

1. Obiettivo

0 = assente; 1 = implicito; 2 = presente ma generico; 3 = chiaro e verificabile.

2. Contesto

0 = assente; 1 = eccessivo o confuso; 2 = utile ma non separato; 3 = essenziale e ben delimitato.

3. Vincoli

0 = assenti; 1 = vaghi; 2 = presenti ma incompleti; 3 = espliciti, non contraddittori e controllabili.

4. Output

0 = non definito; 1 = richiesto in modo generico; 2 = struttura parziale; 3 = formato preciso e riutilizzabile.

5. Densità informativa

0 = molto rumore; 1 = molte parole decorative; 2 = discreto rapporto segnale/rumore; 3 = ogni blocco ha funzione.

6. Verificabilità

0 = impossibile valutare; 1 = valutazione soggettiva; 2 = alcuni criteri; 3 = criteri chiari di successo.

Un prompt professionale non deve necessariamente ottenere 18 punti su 18. Deve però rendere visibili le sue scelte. La rubrica aiuta a capire dove intervenire: se manca l'output, si lavora sul formato; se manca la verificabilità, si aggiungono criteri; se il prompt è prolisso, si eliminano parole senza funzione.

Questa rubrica può essere usata anche con i team aziendali. Ogni reparto può costruire una libreria di prompt valutati e migliorati nel tempo. In questo modo il prompting diventa patrimonio operativo, non talento individuale.

Errori ricorrenti da correggere

Errore 1 - Chiedere qualità senza definirla

Parole come bello, efficace, professionale, chiaro e interessante sono utili nella conversazione umana, ma troppo generiche per un prompt. Vanno tradotte in criteri. 'Professionale' può significare sobrio, tecnico, commerciale, istituzionale o sintetico. 'Chiaro' può significare con esempi, con frasi brevi, senza gergo o con struttura a punti.

Errore 2 - Inserire il vincolo importante alla fine di un periodo lungo

Se una regola è obbligatoria, deve stare in una riga propria. I vincoli nascosti nei periodi lunghi sono facili da perdere. Un prompt non è una narrazione; è una gerarchia di priorità.

Errore 3 - Confondere contesto e istruzione

Il contesto spiega la situazione. L'istruzione dice cosa fare. Se i due livelli sono mescolati, il modello può trattare una descrizione come comando o un comando come semplice informazione. Separare le sezioni riduce questo rischio.

Errore 4 - Non specificare cosa non fare

Molti output sbagliati nascono da divieti non dichiarati: non inventare fonti, non usare tono promozionale, non aggiungere dati esterni, non spiegare il processo, non superare una certa lunghezza. I vincoli negativi sono parte della specifica.

Errore 5 - Chiedere ragionamento quando serve solo output

In alcuni casi è utile chiedere al modello di ragionare per passaggi. In altri, soprattutto nei flussi editoriali o produttivi, serve solo il risultato finale. Il prompt deve distinguere tra processo interno, controllo di qualità e output consegnabile.

Errore 6 - Aggiungere esempi incoerenti

Gli esempi sono potenti, ma devono essere coerenti con il risultato desiderato. Un esempio troppo diverso dal caso reale può deviare il modello. Meglio un esempio breve, pulito e aderente al formato che tre esempi lunghi e ambigui.

Errore 7 - Usare il prompt una volta sola

Un prompt professionale va trattato come una versione. Si prova, si valuta, si corregge, si salva. Se un prompt viene riscritto ogni volta da zero, l'organizzazione perde apprendimento.

Glossario operativo

Prompt: istruzione o insieme di istruzioni fornite a un modello generativo. Non coincide con una domanda: può contenere ruolo, contesto, vincoli, esempi, formato e criteri.

Specificazione: trasformazione di un'intenzione in una richiesta eseguibile. Una specificazione chiarisce cosa deve essere prodotto e come valutarlo.

Vincolo: regola che limita lo spazio di risposta del modello. Può riguardare lunghezza, tono, fonti, struttura, divieti, livello tecnico o formato.

Formato di output: struttura richiesta per la risposta. Può essere testo, tabella, elenco, JSON, Markdown, schema, checklist o documento.

Densità informativa: rapporto tra parole utili e parole totali. Un prompt ad alta densità contiene pochi elementi decorativi e molti segnali operativi.

Sottospecificazione: condizione in cui il prompt lascia implicite informazioni necessarie. Il modello può compensare, ma il risultato diventa meno stabile.

Rumore semantico: parole o frasi che non aiutano il modello a svolgere il compito e possono deviare tono, priorità o formato.

Prompt monolitico: prompt scritto come unico blocco narrativo. È spesso leggibile per un umano, ma difficile da diagnosticare e migliorare.

Prompt modulare: prompt diviso in sezioni con responsabilità distinte. È più adatto a processi professionali, agenti e automazioni.

Conclusione

La grammatica perfetta non è il nemico dell'AI. Il nemico è la confusione vestita da buona scrittura. Un prompt scritto benissimo può essere ambiguo, sottospecificato e poco controllabile. Un prompt più scarno, se progettato come specifica, può produrre risultati migliori perché riduce le decisioni implicite e orienta il modello verso un output verificabile.

La competenza centrale non è parlare all'AI in modo più elegante, ma trasferire intenzione operativa in modo più esatto. Questo richiede analisi, scomposizione, priorità, vincoli, esempi e criteri. Richiede di pensare come progettisti di interfacce, non come autori di temi.

Per docenti e formatori, il messaggio pedagogico è forte: insegnare prompting non significa insegnare a fare domande belle. Significa insegnare a trasformare desideri vaghi in istruzioni eseguibili. Per professionisti e AI officer, il messaggio operativo è altrettanto chiaro: i prompt migliori sono quelli che possono essere testati, riusati, versionati e migliorati.

Scrivere prompt efficaci non significa impoverire la lingua. Significa usarla con disciplina. La frase non deve impressionare il modello. Deve governarlo.