UNIVERSITÀ
DI PAVIA

UNIVERSITÀ DEGLI STUDI DI PAVIA

Dipartimento di Studi Umanistici

Corso di Laurea Magistrale in Linguistica teorica, applicata e delle lingue moderne

PROMPT ENGINEERING E LLMS-AS-JUDGES. ETET COME CASO DI STUDIO

Relatrice:

Prof.ssa Chiara ZANCHI

Correlatrice:

Prof.ssa Claudia Roberta COMBEI

Tesi di Laurea Magistrale di

Paolo FECCIA

Matricola n. 522946

Anno Accademico 2024/2025

Abstract

L'uso sempre più capillare dei *Large Language Models* per risolvere task quotidiani e precedentemente portati a termine solo da esseri umani ha richiesto una maggior ricerca e consapevolezza sul *prompt engineering* e su come funzionino i prompt, strumenti essenziali nell'impartire indicazioni ai modelli.

Il presente lavoro ha lo scopo di indagare con quali strategie è possibile redarre efficacemente i prompt, seguendo sia indicazioni di buona scrittura, sia tecniche più complesse che coinvolgono la struttura del testo e stimolano gli LLM a compiere ragionamenti più approfonditi. Inoltre, viene fatta anche menzione dei sistemi di valutazione automatica impiegati nella valutazione delle prestazioni dei prompt. L'esperimento condotto ha coinvolto l'uso del modello interno di ETET (GPT-5.2) e ha visto l'utilizzo di vari prompt personalizzati con modifiche atte a dimostrare gli effetti che queste hanno avuto sulle prestazioni del sistema.

In conclusione, i risultati dell'esperimento dimostrano che una migliore separazione delle sezioni del prompt, insieme all'uso di titoli e di elenchi ordinati, aiutano a indirizzare meglio l'attenzione del modello e migliorano la comprensione del compito e delle azioni da intraprendere. Inoltre, anche una maggior sinteticità e un uso più consapevole delle tecniche presentate, come, ad esempio, il *role prompting*, conduce ad alcuni miglioramenti osservabili nelle prestazioni del modello.

Parole chiave: *Prompt engineering*, LLM, GPT-5.2, LLM*s-as-judges*, *Automated Essay Scoring* (AES).

Indice

Introduzione

Lo scopo di questo lavoro consiste nell'osservare da vicino come funzioni il *prompt engineering* e quali possono essere le modalità per redarre i prompt in modo efficace e ottimizzato, tramite sistemi collaudati. Il sempre maggior utilizzo degli *Large Language Models* (LLM) da parte di un ampio pubblico, per risolvere task quotidiani o che in passato venivano portati a termine solo da operatori umani, come il task di valutazione oggetto della presente ricerca (anche definito *Automated Essay Scoring*), ha condotto all'analisi più ravvicinata di quali possono essere le modalità migliori per sfruttare massimamente le capacità degli LLM attraverso i prompt, ossia le istruzioni verbali che vengono inviate ai modelli per eseguire i compiti. Queste istruzioni assumono l'aspetto di un comando verbale che non viene analizzato solo attraverso le sue parole chiave, ma anche tramite l'assegnazione di pesi alle parole per comprendere meglio il contesto sintattico e, per quanto possibile, quello semantico (Vaswani et al. 2017). Per i motivi appena elencati, risulta spontaneo ad un utilizzatore occasionale o non esperto concepire i prompt come semplici richieste scritte, al pari di una *query* immessa in un comune motore di ricerca. In realtà, il funzionamento dei prompt è più delicato di come si dipinge e, come dimostrerò nei capitoli a seguire (su tutti, il capitolo 2), la comprensione epistemica e contestuale degli LLM è ben diversa rispetto a quella degli esseri umani e procede secondo regole statistiche che nulla hanno a che fare coi sensi regolarmente usati per comprendere le situazioni e trarne significato. Per questo motivo, illustrerò nel capitolo 1 le basi teoriche fondamentali per comprendere come funzionano i prompt e il *prompt engineering*, derivate dalla letteratura scientifica disponibile sul tema, con una menzione particolare alle tecniche di buona scrittura e di ottimizzazione che hanno già trovato riscontri, oltre ai metodi di valutazione automatica che vengono usati per valutare le prestazioni dei modelli e dei relativi prompt usati. Successivamente, nel capitolo 2 mostrerò come odiernamente vengono usati gli LLM come "giudici" per valutare

prestazioni terze e come il modo di procedere dei modelli confligga particolarmente con il modo in cui invece gli esseri umani si relazionano col mondo e con i testi. Nel capitolo 3 parlerò del caso di studio, ossia una ricerca effettuata nell'ambito del mio tirocinio curricolare insieme al *team* di ETET nei riguardi del *prompt engineering* applicato alla valutazione automatizzata dei risultati di test linguistici sottoposti online ad un gruppo di utenti. Nel capitolo 4, dopo aver illustrato la ricerca effettuata in ETET, mostrerò come, sfruttando la teoria descritta nei capitoli 1 e 2, sia possibile migliorare il lavoro di redazione del prompt per ottenere prestazioni superiori dall'LLM. Infine, nel capitolo 5 presenterò le conclusione derivate dalle suddette modifiche e rifletterò su come abbiano alterato le prestazioni originarie del prompt.

Capitolo 1 Presupposti teorici

In questo capitolo verranno presentati i presupposti teorici che costituiscono le fondamenta del presente lavoro. Descriverò lo stato dell'arte del *prompt engineering*, illustrando come funzionano i prompt usati per istruire gli LLM e quali sono le principali tecniche per rendere efficace un input. In modo particolare, ci si concentrerà sulle dinamiche di interazione con il modello GPT-4 e, quando possibile, con GPT-4o, essendo quest'ultimo uno dei modelli che verranno utilizzati nell'esperimento illustrato nei capitoli 3 e 4, insieme a GPT-5.2.

1.1. Prompt e *prompt engineering*

Con l'avvento della *Natural Language Generation* (NLG) all'interno del *Natural Language Processing* (NLP) e, in particolare, con la nascita degli LLM basati su *transformer*, ossia intelligenze artificiali che sfruttano il *deep learning*, le reti neurali ricorrenti (RNN, *Recurrent Neural Network*) e il meccanismo dell'attenzione per analizzare e riprodurre il linguaggio umano, l'interazione per l'ottenimento di dati da questi modelli si sviluppa attraverso conversazioni; quindi, a un input (il cosiddetto prompt) inserito dall'utente che fruisce dell'interfaccia, segue un output, ovvero una risposta con la quale il modello porta a compimento il compito (o task) che gli è stato assegnato nella prima interazione. Il prompt può essere testuale o può consistere in altri tipi di medium, come immagini, audio o video; in questa sede analizzerò soltanto i prompt testuali, in quanto nel sistema studiato[1] è possibile utilizzare solo prompt testuali (Schulhoff et al. 2025: 5).

Il motivo per cui nasce la scienza del *prompt engineering* è da ricercare nella natura di *black box* degli LLM: prendendo come esempio ChatGPT (di cui parleremo più estesamente nei capitoli 3 e 4), questo fa parte della famiglia dei

[1] La piattaforma ETET valuta le risposte attraverso GPT-4o (poi GPT-5.2) di OpenAI8 e un sistema ASR basato su AzureAI6, che usa Whisper di OpenAI7.

GPT, *Generative Pre-trained Transformer*, ossia intelligenze artificiali che generano testi in linguaggio naturale basate sulla tecnologia dei *transformer* e sul meccanismo dell'attenzione; la terza caratteristica, il fatto di essere *pre-trained*, in italiano "pre-addestrato", dà origine alla *black box* menzionata sopra: il modello, al momento del suo uso, è già stato sottoposto a fasi di apprendimento sui dati e non è possibile intervenire su questi passaggi. Per questo motivo, il *prompt engineering* diventa di vitale importanza, dato che è grazie a esso che è possibile istruire il modello e ottenere output basati sui dati fornitigli in precedenza. Talvolta, può essere utile attuare il *fine-tuning*, ossia quella pratica che vede l'adattamento di un modello pre-addestrato a un task specifico, selezionando una parte dei dati complessivi e creando, di fatto, un insieme di dati più piccolo su cui il modello deve concentrarsi nell'apprendimento, ma questa possibilità nei modelli commerciali con interfaccia utente non c'è quasi mai. Quindi, ancora più vitale diventa il conoscere le strategie e le tecniche per rendere i prompt efficaci e "leggibili" dal modello, essendo il modo più diretto attraverso cui è possibile esplorare i dati e stimolare la generazione di materiale per assolvere ai compiti definiti negli input (Chen et al. 2025: 4).

Inoltre, il crescente uso delle intelligenze artificiali (IA) conversazionali anche da parte di non esperti del settore e, in generale, la diffusione dell'esigenza di sfruttare le IA per la risoluzione di task sempre più complessi alimenta il bisogno di conoscere metodi collaudati ed efficaci per impostare i prompt, usando in modo consapevole la loro struttura e sapendo sfruttare la conoscenza di come gli LLM lavorano.

Al fine di creare prompt in grado di ottenere le risposte desiderate dal modello è necessario individuare le varie parti di cui un prompt è composto e le loro caratteristiche, dato che ognuno di questi elementi possiede un determinato scopo e che autori diversi dividono il prompt in modo diverso e assegnando funzioni diverse. In Eager & Brunton (2023: 4) la struttura del prompt viene divisa in sei segmenti diversi e di seguito propongo l'esempio presentato nel lavoro citato per specificarne chiaramente le parti.

(1) “Scrivi un caso di studio per gli studenti del primo anno di un corso universitario di marketing. Il caso di studio dovrebbe illustrare le sfide affrontate da una piccola azienda nello sviluppo di una strategia di marketing sui social media per attrarre nuovi clienti. Il caso di studio verrà utilizzato per supportare gli studenti nel raggiungimento dei risultati di apprendimento previsti [inserire obiettivi di apprendimento]. Il caso di studio dovrebbe essere lungo circa 2000 parole, includere una breve descrizione dell’azienda e descrivere le sfide affrontate dall’azienda in relazione alla progettazione e alla fornitura di una strategia di social media e possibili soluzioni, e includere domande sul caso che gli studenti possono discutere in classe.” (Eager & Brunton, 2023: 4, traduzione mia)

La parte preponderante è quella del verbo (“scrivi”), ossia l’azione specifica che il modello deve eseguire, individuata nel solo predicato verbale; successivamente, il focus (“un caso di studio”) fornisce il processo o il risultato dell’azione, quindi l’oggetto dell’azione esplicitata dal verbo; il contesto (“per gli studenti del primo anno di un corso universitario di marketing.”) spiega lo scopo o i parametri del task; la sezione focus e condizione (“Il caso di studio dovrebbe [...] nuovi clienti.”) definisce meglio l’argomento e l’obiettivo primario del task; in seguito, l’allineamento (“Il caso di studio verrà [...] apprendimento.”) istruisce il modello per orientare il contenuto dell’output verso l’obiettivo desiderato; infine, i vincoli e le limitazioni (“Il caso di studio dovrebbe [...] in classe.”) rendono chiari i limiti cui l’IA deve aderire.

In Lemeš (2024: 166), il prompt ideale viene diviso in cinque parti principali e qui lo confronto con la versione di prompt precedentemente descritta in (1): una parte introduttiva con un esempio di *role prompting* (di cui si parlerà più estesamente nel paragrafo 1.1.2) che in (1) non è presente, ma che potrebbe essere “Sei un esperto in marketing.”; una seconda parte con l’effettiva spiegazione del task e quindi dell’azione che deve intraprendere il modello, che in (1) è “Scrivi un caso di studio […] marketing.”; una terza parte con

informazioni contestuali utili, rappresentate in (1) dalla sezione “Il caso di studio dovrebbe illustrare [...] apprendimento.”; una quarta parte con istruzioni più specifiche su ciò che deve generare il modello, che in (1) è la sezione finale “Il caso di studio dovrebbe essere [...] in classe.”; un’ultima parte con spiegazioni sul formato desiderato in output, che in (1) non è presente, ma può essere “Voglio che sia un file word.”.

In Ggaliwango et al. (2024: 390) il prompt viene diviso in quattro parti: la prima, quella delle “istruzioni”, rispecchia ciò che per Eager & Brunton (2023: 4) costituiscono il “verbo” e il “focus” insieme, ossia la parte in cui vengono fornite le direttive per il modello; la seconda parte costituisce i “dati di input”, ma non vengono fornite abbastanza informazioni per identificare correttamente la funzione di questa parte del prompt, perciò si può evidentemente supporre che sia una sezione in cui si possono trovare informazioni aggiuntive che aiutano il modello a capire meglio il prompt; la parte successiva, il “contesto”, aderisce, con alcune differenze, al “focus” descritto da Eager & Brunton (ibidem), rappresentando una sorta di oggetto dell’azione principale commissionata dal prompt; l’ultima parte, “l’indicatore di output”, descrive azioni aggiuntive secondarie che vengono ordinate al modello e rispecchia, per certi versi, la parte sopra descritta in Eager & Brunton (ibidem) di “focus e condizione”.

In Schulhoff et al. (2025: 5) vengono descritte sei parti che costituiscono il prompt: la “direttiva”, ossia la vera e propria istruzione o domanda che viene posta al modello, simile “all’istruzione” in Ggaliwango et al. (2024: 390); gli esempi, che, nel caso del *few-shot prompting* (di cui parlerò più estesamente nel paragrafo 1.1.2), possono essere aggiunti per mostrare al modello come deve effettuare il compito; il formato dell’output, ossia come si desidera che il modello fornisca l’output, ad esempio CSV o Markdown; eventuali istruzioni sullo stile da adottare nell’output; il possibile ruolo che può essere interpretato dal modello; infine, informazioni aggiuntive che possono fare da contesto all’istruzione.

Alla luce delle differenze presentate sopra nella divisione in parti del prompt, possiamo desumere che quelle fondamentali siano quelle che riguardano

l'azione principale cui il modello deve attenersi per portare a termine il task. Tuttavia, come i lavori confrontati evidenziano, presentare il mero compito da far eseguire al modello non basta per ottenere delle *performance* sufficienti; per questo motivo è necessario sia aggiungere una buona dose di contesto (inteso in senso generico di informazioni collaterali riguardanti il task), sia adottare delle pratiche di buona scrittura per rendere l'input testuale più comprensibile possibile al modello (Chen et al. 2025: 4). Inoltre, alcuni degli autori citati sopra, nel presentare le parti di cui può essere composto un buon prompt, sottolineano anche un ordine ideale da rispettare per mettere in sequenza le sezioni: in Eager & Brunton (2023: 4) e Ggaliwango et al. (2024: 390) l'impostazione del prompt esordisce direttamente con l'azione specifica che si vuole fare eseguire al modello, successivamente si forniscono informazioni contestuali aggiuntive e in chiusura informazioni che riguardano l'output. Invece, in Lemeš (2024: 166), si esordisce con un esempio di *role prompting* e, poi, si segue la struttura già citata di cui hanno fatto uso gli autori precedenti.

1.1.1. Pratiche di buona scrittura dei prompt

Prima di affrontare le vere e proprie tecniche di ottimizzazione dei prompt, è necessario riassumere alcune pratiche generali di buona scrittura utili a un'impostazione efficace del prompt e fondamentali per preparare il terreno a metodi più complessi. In Korzynski et al. (2023: 31) si trovano alcune raccomandazioni generali su come impostare un buon prompt: prima di tutto si suggerisce di articolare chiaramente l'istruzione su ciò che il modello deve fare, anche rendendo chiari gli elementi del prompt e specificando la conclusione dell'input scritto; successivamente, si suggerisce di fornire esempi (tecnica di cui si parlerà meglio nel paragrafo 1.1.2), di dividere un prompt troppo complesso in parti più semplici e di fare attenzione al numero massimo di token che un modello può supportare nei dati di input. In Ggaliwango et al. (2024: 392), oltre ad alcuni metodi già citati, è presente un'ulteriore riflessione interessante: si

suggerisce di sperimentare la scrittura di prompt con stili e formati diversi per osservare quali funzionano meglio, a causa della natura non deterministica degli LLM. In Chen et al. (2025: 5), oltre al consiglio basilare di fornire delle istruzioni più complete possibile per guidare l'output verso una risposta più specifica, si suggerisce di confezionare un prompt che sia chiaro e preciso, evitando ambiguità e genericità che possono fuorviare il modello: un prompt dettagliato e accurato permette al modello di generare un output meglio allineato con le caratteristiche del compito assegnato e riduce l'incertezza nella risposta. Oltre al consiglio di utilizzare, nei prompt più complessi, le virgolette triple per separare meglio il contenuto e renderlo più leggibile, in Chen et al. (2025) si suggerisce di provare il cosiddetto "*resampling*", ossia l'inviare più volte lo stesso input per notare le differenze nell'output generato e scegliere quello migliore; questo nuovamente a causa della natura non deterministica degli LLM, che li induce a produrre output diversi a fronte di uno stesso input e di impostazioni di generazione invariate.

1.1.2. Tecniche di ottimizzazione dei prompt

Oltre alle pratiche di buona scrittura illustrate nel paragrafo 1.1.1, sono discusse in letteratura numerose altre tecniche che approfondiscono i modi in cui si può potenziare od ottimizzare un prompt per ottenere risultati più efficaci e output più in linea con le istruzioni impartite ai modelli. Per rassegne esaustive, si vedano ad esempio White et al. (2023) e Schulhoff et al. (2025). In questa sede, si ritiene opportuno presentare alcuni dei procedimenti più interessanti e utili per la realizzazione di prompt più complessi ed efficaci al fine di esplorarli sperimentalmente nel capitolo 4.

In White et al. (2023: 1) si propone l'idea di *prompt pattern*, ossia strutture più avanzate degli input per superare task complessi con *performance* migliori, comparandola a quella di *software pattern*, dato il fatto che i prompt sono una forma di programmazione e perciò possono essere assimilati ai

procedimenti usati con i *software*. Questi due procedimenti si somigliano in quanto offrono soluzioni replicabili a problemi specifici e, nel lavoro citato, i *pattern* che riguardano i prompt vengono classificati in categorie che li raggruppano in base al tipo di ottimizzazione che forniscono e vengono corredati di descrizioni che chiariscono il problema che il *pattern* risolve e il compito che porta a termine, la spiegazione logica del problema e perché è importante risolverlo, la struttura e le idee chiave, oltre a esempi d'uso e alle conseguenze del suo utilizzo. Uno dei problemi principali che vengono sottolineati dagli autori del testo sopra citato è la necessità di adottare l'ottica di scrittura di un prompt per cui non basta utilizzare una buona grammatica, ma è necessario usare un modo chiaro di comunicare le informazioni che sappiamo essere più utili per il modello, così da concentrare il linguaggio e da renderlo più efficace, tenendo anche conto del fatto che i prompt non devono essere solo una serie di token, ma devono trasmettere idee più complesse e articolate. Inoltre, gli autori avanzano una proposta di descrizione dei *pattern* basata su "dichiarazioni contestuali fondamentali", ossia descrizioni delle idee più importanti su cui basare il prompt e che un utente qualunque può usare per riscrivere a suo piacimento, conscio di quali sono i punti basilari da non trascurare.

I *pattern* presentati in White et al. (2023: 4) appartengono a sei diverse categorie che tentano di inquadrare le funzionalità dei modelli cui si riferiscono: la prima categoria riguarda la "semantica di input" e ha a che fare con il modo in cui l'LLM comprende l'input e il modo in cui lo traduce in materiale che può usare per produrre l'output; la seconda categoria pertiene la "personalizzazione dell'output", ossia come si può limitare o adattare i tipi, i formati, le strutture o altre proprietà degli output; la terza categoria, quella di "identificazione dell'errore", si focalizza sulla ricerca e la risoluzione degli errori nell'output generato dall'LLM; nella quarta categoria, quella di "miglioramento del prompt", ci si focalizza sul migliorare la qualità dell'input e dell'output; la quinta categoria, che riguarda "l'interazione", si focalizza sull'interazione tra l'utente e l'LLM; infine, l'ultima categoria, che concerne il "controllo del contesto", vede i

pattern creati per controllare le informazioni contestuali in cui l'LLM opera. Come si vedrà in seguito, alcuni dei *pattern* proposti nel suddetto articolo hanno un funzionamento molto basilare e, per questo motivo, gli autori stessi raccomandano l'uso della maggioranza di questi *pattern* in combinazione tra di loro, in modo tale da rafforzarne l'effetto e da limitare eventuali *bias* o allucinazioni.

I *pattern* descritti in White et al. (2023: 4) che sottolineo in questa sede sono quelli che possono essere più utili nella fase di sperimentazione che verrà illustrata nel capitolo 4 e, tra quelli che appartengono alle categorie sopra citate, si vogliono evidenziare i seguenti. Il primo *pattern* citato, il *Meta Language Creation Pattern*, può avere un'utilità generale dato che, specificando al modello le coordinate di un linguaggio alternativo che si intende usare ed esplicitandolo attraverso delle traduzioni o delle spiegazioni inserite nel prompt, si facilita la comunicazione con l'LLM e si evitano ambiguità. Un altro *pattern* interessante è il *Persona Pattern* (più spesso definito *Role Prompting*), attraverso cui si istruisce il modello affinché adotti un punto di vista particolare (come quello di un professionista in un settore) per produrre un output più mirato alla risoluzione del task. Successivamente, il *Template Pattern* può essere utile per assicurarsi che l'output generato dal modello segua una precisa strutturazione. In alcuni casi, ad esempio quando si tratta di generare dati JSON (*JavaScript Object Notation*), c'è una notevole varietà nel modo in cui i dati possono venire rappresentati, per questo motivo il *template* può assicurare che questa rappresentazione rispetti i vincoli imposti dall'utente. Il *Reflection Pattern* si prefigge lo scopo di ottenere dal modello delle spiegazioni logiche sulle scelte che ha operato e sulle risposte fornite; questo *pattern* permette di valutare meglio la validità dell'output, chiarendo punti di confusione e rivelando lacune nelle conoscenze del modello. I *pattern* che appartengono alla categoria di *Prompt Improvement* rappresentano tutti esempi interessanti di tecniche volte a migliorare la qualità generale sia dell'input testuale, sia dell'output generato dal modello: il *Question Refinement Pattern* stimola l'LLM a suggerire prompt potenzialmente migliori o più rifiniti

che l'utente può usare al posto del prompt originale; l'*Alternative Approaches Pattern* ha lo scopo di offrire metodi alternativi per risolvere un task in modo tale che l'utente non utilizzi solo metodi che gli sono familiari; il *Cognitive Verifier Pattern* costringe l'LLM a suddividere sempre le richieste o le domande fornite nel prompt in molteplici sotto-richieste o sotto-domande, sfruttate per dare una risposta migliore alla domanda originale; il *Refusal Breaker Pattern* consiste nel richiedere esplicitamente all'LLM di riformulare il prompt qualora quest'ultimo si rifiutasse di fornire una risposta al prompt originale. Infine, l'ultimo *pattern* che si vuole evidenziare è il *Context Manager Pattern* che permette all'utente di specificare o rimuovere contesto in una conversazione con l'LLM al fine di focalizzare la conversazione su un argomento specifico o escludere argomenti non correlati con quello desiderato.

In Sahoo et al. (2024: 1) vengono presentate quelle che da ora in poi definirò tecniche oppure *template*, ossia meccanismi di ottimizzazione di prompt che hanno alcuni punti di tangenza coi *pattern* sopra citati, ma che hanno una differente strutturazione e una maggior presenza nella letteratura scientifica; in Schulhoff et al. (2025: 5) vengono definiti come funzioni che contengono una o più variabili colmate dal medium utilizzato per creare il prompt quindi il prompt può essere considerato un'istanza del *template*. Inoltre, si fa notare che la definizione di questi stratagemmi come *template* non ha molto a che vedere con il precedentemente citato *Template Pattern* (White et al. 2023: 12), ma vuole rappresentare un ambito più ampio di schematizzazioni di prompt pronti per un riutilizzo da parte degli utenti.

Le prime tecniche che vengono presentate non necessitano di un addestramento approfondito, perciò fanno solo affidamento su prompt elaborati accuratamente, e sono il *Zero-Shot Prompting* e il *Few-Shot Prompting*: il primo consiste nel fornire al modello una descrizione del task che deve affrontare senza esempi (*shot*) di alcun tipo e senza dati etichettati per addestramenti più specifici, in questo modo il modello per le sue previsioni si basa solamente sulla sua conoscenza preesistente; il secondo fornisce al modello alcuni esempi in

interazioni input-output per migliorare la comprensione del task da parte dell'LLM. Successivamente, vengono descritte tecniche che riguardano gli stimoli di ragionamento e logica del modello. In Schulhoff et al. (2025: 10), si descrivono il *Few-Shot Prompting* e lo *Zero-Shot Prompting* in modo più accurato e si forniscono suggerimenti e tecniche più approfondite per migliorare le loro prestazioni: per quanto riguarda i suggerimenti generali, si fa notare che deve essere posta particolare attenzione alla quantità di esempi che vengono scelti, al modo in cui vengono ordinati, alla distribuzione delle etichette, alla qualità delle etichette, al formato degli esempi e alla loro similarità, perché tutte queste caratteristiche influiscono sulle prestazioni del modello; per quanto riguarda le tecniche più approfondite di *Few-Shot Prompting* si citano il *K-Nearest Neighbor* (KNN), algoritmo che seleziona esempi simili ai dati di test, oppure il *Vote-K*, che assicura che nuovi esempi aggiunti in un gruppo di esempi già inseriti siano sufficientemente diversi, oppure ancora il *Self-Generated In-Context Learning* (SG-ICL), che usa una IA generativa per generare esempi, o infine il *Prompt Mining*, che attua un processo di ricerca di "*middle words*" ottimali per il prompt; per approfondire lo *Zero-Shot Prompting* vengono proposte altrettante tecniche: oltre ad alcuni *template* già citati, viene proposto lo *Style Prompting*, che punta a specificare nel prompt stile, tono e genere desiderati per l'output, oppure l'*Emotion Prompting*, che incorpora frasi di rilevanza psicologica, oppure ancora il *SimToM*, che tenta di creare un set di fatti che una persona conosce e poi risponde all'input basandosi solo su quei fatti, eliminando così la presenza di informazioni irrilevanti; il *Re-reading* (RE2) aggiunge al prompt la frase "leggi di nuovo la domanda" insieme alla ripetizione dell'input, mentre il *template Self-Ask* chiede all'LLM di decidere se necessita di domande supplementari per un dato prompt e, in caso positivo, l'LLM genera queste domande, risponde a esse e infine risponde alla domanda originale.

Il *Chain-of-Thought* (CoT) *Prompting* tenta di risolvere il problema della difficoltà da parte degli LLM di comprendere ragionamenti complessi esplicitando i passaggi che il modello deve seguire nella risoluzione del task e

riuscendo a ottenere risposte più strutturate e razionali (Sahoo et al. 2024: 2). Lo *Zero-Shot-CoT* ne è la versione più diretta e non contiene esempi, ma solo la frase che induce il ragionamento, ossia "*Let's think step-by-step*"; alcune modifiche al metodo possono essere lo *Step-Back Prompting*, in cui inizialmente si fa al modello una domanda generica riguardo concetti o fatti rilevanti prima di approfondire il ragionamento, oppure l'*Analogical Prompting*, che genera automaticamente esempi che includono il CoT (Schulhoff et al. 2025: 12). Dall'altra parte, il *Few-Shot-CoT*, chiamato anche *Manual* o *Golden CoT*, possiede i suoi specifici metodi di approfondimento, come l'*Uncertainty-Routed CoT Prompting*, che campiona multipli percorsi di ragionamento e poi seleziona la risposta maggioritaria se è sopra un certo limite, in caso contrario usa il campionamento *greedy*; oppure il *Complexity-based Prompting*, che seleziona esempi complessi per l'annotazione e l'inclusione nel prompt, basati su fattori come la lunghezza delle domande o fasi di ragionamento richieste, poi, durante l'inferenza, campiona catene di ragionamento multiple e seleziona quelle che raggiungono una certa lunghezza, premettendo che un ragionamento più lungo indica una miglior qualità; infine, il *Memory-of-Thought Prompting*, che sfrutta gli esempi non etichettati in addestramento per costruire prompt *Few-Shot CoT* come test (ivi: 13). L'*Automatic Chain-of-Thought* (Auto-CoT) *Prompting* vuole risolvere il problema del costoso dispendio di tempo e della difficoltà dati dal creare manualmente prompt di CoT ad alta qualità. L'*Auto-CoT* istruisce automaticamente l'LLM con il passaggio "*Let's think step-by-step*" aggiunto al prompt per generare automaticamente catene di ragionamento e, campionando molteplici domande, produce catene di ragionamento distinte per ciascuna di esse. Un metodo di CoT migliorato è la *Self-Consistency*, ossia un sistema, simile all'*Auto-CoT*, che campiona diversi percorsi di ragionamento e poi seleziona la risposta più consistente marginalizzando i percorsi di ragionamento campionati. La *Universal Self-Consistency*, invece di selezionare la risposta maggioritaria contando programmaticamente quante volte occorre, inserisce tutti gli output in un *template* che seleziona la risposta preponderante (ivi: 15). Il *Meta-Reasoning*

over Multiple CoTs prima genera multiple catene di ragionamento, poi inserisce tutte le catene in un singolo *template* e genera la risposta finale, in modo simile alla *Universal Self-Consistency* (ibidem). Il *Logical Chain-of-Thought* (LogiCoT) *Prompting* garantisce meccanismi di verifica aggiunti al normale CoT, sfruttando i principi della logica simbolica per migliorare il ragionamento; applica il concetto logico di *reductio ad absurdum* per verificare ogni fase di ragionamento e fornire un *feedback* mirato per correggere le fasi di ragionamento inesatte. Il meccanismo del *Tree-of-Thoughts* (ToT) *Prompting* estende il CoT architettando una struttura ad albero con passaggi intermedi di ragionamento, chiamati *thoughts*; ogni "pensiero" rappresenta una sequenza linguistica coerente direzionata verso la soluzione finale e genera dei progressi che vengono valutati dal modello. Il sistema *Graph-of-Thoughts* (GoT) si ispira alla natura non lineare del pensiero umano e si contrappone all'approccio convenzionale sequenziale del CoT, utilizzando un *framework* basato su grafici; questo metodo permette un'interazione dinamica, il *backtracking* (ossia la possibilità di tornare ai passaggi precedenti in un processo) e la valutazione dei vari "*thoughts*", permettendo l'aggregazione e la combinazione di "pensieri" provenienti da rami diversi. Il *template* di *System 2 Attention* (S2A) *Prompting* approfondisce il meccanismo dell'attenzione già presente negli LLM basati su *transformer* e risolve il problema dell'inclusione di informazioni contestuali irrilevanti (che influenzano negativamente la generazione dei token) rigenerando il contesto di input dopo aver affinato l'attenzione in modo selettivo sugli elementi più rilevanti; il processo si articola in due passaggi in cui si rigenera sia il contesto, sia la risposta con il contesto rifinito. Con il *Thread of Thoughts* (ThoT) *Prompting* l'LLM esamina sistematicamente contesti estesi in segmenti gestibili per un'analisi incrementale, utilizzando un approccio in due fasi in cui prima riassume ed esamina ogni segmento, poi perfeziona le informazioni in vista della risposta finale.

Per ridurre le allucinazioni, ossia output generati dagli LLM che possono risultare grammaticalmente corretti e internamente coerenti, ma senza senso o

fattualmente scorretti, in Sahoo et al. (2024: 4) vengono mostrati alcuni metodi che si riportano di seguito. Il primo metodo, il *Retrieval Augmented Generation* (RAG), comprendendo che gli LLM sono dipendenti da dati di addestramento limitati e statici, integra il recupero di informazioni nel processo di *prompting*: analizza l'input dell'utente, costruisce una *query* mirata e cerca le informazioni pertinenti in una *knowledge base* predefinita; le informazioni recuperate vengono poi incorporate nel prompt originale arricchendolo.

Il *ReAct Prompting* permette all'LLM di generare tracce di ragionamento e azioni *task-specific* contemporaneamente; questo processo migliora la sinergia tra ragionamento e azione, aiutando il modello nel gestire i piani d'azione; in particolare, *ReAct*, nella risposta alle domande e nella verifica dei fatti, argina il problema delle allucinazioni e della propagazione degli errori interagendo con una API (*Application Programming Interface*) di Wikipedia e, quindi, producendo traiettorie di risoluzione di task più interpretabili. Il *Chain-of-Verification* (CoVe) *Prompting* prevede un processo sistematico diviso in quattro passaggi per limitare le allucinazioni: il modello genera delle risposte di base, pianifica delle domande di verifica per controllare il lavoro svolto, risponde alle domande in modo indipendente e produce una risposta riveduta che incorpora la verifica fatta; in questo modo vengono migliorate le abilità di ragionamento logico e si riducono gli errori, anche quando sono presenti informazioni contraddittorie.

Il metodo di *Chain-of-Note* (CoN) *Prompting* si occupa di risolvere i problemi derivanti dalle risposte insufficienti forniti dagli LM *Retrieval-Augmented* (RALM), ossia modelli che già incorporano informazioni esterne per ridurre le allucinazioni, ma che non garantiscono l'affidabilità della conoscenza e possono condurre a risposte erronee o insufficienti; il CoN *Prompting* valuta sistematicamente la rilevanza dei documenti analizzati, enfatizzando le informazioni critiche e affidabili per filtrare i contenuti irrilevanti, ottenendo risposte più precise e contestualmente rilevanti.

L'ultimo metodo per ridurre le allucinazioni, il *Chain-of-Knowledge* (CoK) *Prompting*, parte dal presupposto che i modelli hanno più difficoltà nel gestire problemi complessi a causa delle risorse di conoscenza limitate, dell'inefficacia nella generazione di *query* strutturate e della mancanza di correzioni progressive; per questi motivi, il CoK *Prompting* suddivide sistematicamente task complessi in passaggi ben coordinati: in una fase di preparazione del ragionamento viene stabilito il contesto e inquadrato il problema, poi in una fase dinamica di adattamento delle conoscenze vengono raccolte informazioni dalle varie fonti interrogate, come la conoscenza di base interna del modello, *database* esterni e il prompt dato.

Per quanto riguarda l'interfaccia utente, è presente un singolo *template*, l'*Active Prompting*, basato sul meccanismo di *active learning* (Ggaliwango et al. 2024: 396), che tenta di risolvere il problema dell'adattamento dell'LLM a task di ragionamento diversi attraverso esempi di prompt *task-specific* con ragionamento *chain-of-thought*; invece che basarsi su set statici di esempi annotati da umani (come nei normali metodi CoT), l'*Active-Prompt* introduce un meccanismo per determinare le domande più significative per l'annotazione e, prendendo ispirazione dall'apprendimento basato sull'incertezza, usa varie metriche per definire l'incertezza e selezionare le domande più incerte per l'annotazione.

Per quanto riguarda il *fine-tuning* e l'ottimizzazione, il metodo presentato dell'*Automatic Prompt Engineer* (APE) genera e seleziona in modo dinamico i prompt più efficaci per task specifici, analizzando l'input dell'utente, creando istruzioni e sfruttando il *reinforcement learning* per scegliere il prompt ottimale, adattandolo ai diversi contesti. Anche la categoria di ragionamento e generazione *knowledge-based* annovera un solo metodo, *l'Automatic Reasoning and Tool-use* (ART): consente all'LLM di ragionare attraverso processi *multi-step* e di integrare perfettamente competenze esterne, con *tool* usati per conoscenze specialistiche e calcoli; l'ART automatizza le fasi di ragionamento per mezzo di programmi strutturati ed elimina la necessità di lavoro manuale mettendosi

autonomamente in pausa per integrare output da *tool* esterni e riprendendo il flusso senza soluzione di continuità. Per migliorare la consistenza e la coerenza, il metodo del *Contrastive Chain-of-Thought* (CCoT) *Prompting* fornisce dimostrazioni di ragionamento sia valide che non valide insieme ai prompt originali, imparando anche dagli errori, a differenza del CoT tradizionale.

Nella categoria di ottimizzazione ed efficienza, il metodo *Optimization by Prompting* (OPRO) utilizza i prompt in linguaggio naturale per generare iterativamente soluzioni basate sulla descrizione del problema, consentendo un rapido adattamento a task diversi e personalizzando il processo di ottimizzazione.

La penultima categoria di *template* presentata da Sahoo et al. (2024: 7) è quella che riguarda la comprensione dell'intento dell'utente da parte dell'LLM e, con il *Rephrase and Respond* (RaR) *Prompting*, al modello si consente di riformulare ed espandere le domande in un singolo prompt, dimostrando miglior comprensione e accuratezza nelle risposte; la variante di RaR a 2 fasi, che incorpora LLM di riformulazione delle domande e di risposta, ottiene miglioramenti sostanziali in vari task.

In Ggaliwango et al. (2024: 395) vengono presentate ulteriori tecniche avanzate di *prompt engineering* oltre a quelle già citate e, di seguito, si prendono in considerazione i meccanismi compatibili con i modelli basati su *transformer*, in particolare con GPT-4. Il *Knowledge generation prompting* (anche *Generated Knowledge* in Chen et al. [2025: 10]) incoraggia l'LLM a generare nuova conoscenza basandosi sulla sua conoscenza di base. Il *Dynamic Prompting* corregge dinamicamente il prompt in base alle risposte precedenti dell'LM per migliorare le *performance* nel tempo. Il *Transfer learning prompting* utilizza il *transfer learning* per adattare LM pre-addestrati a nuovi task o domini usando prompt redatti accuratamente. Il *Curriculum learning prompting* si basa sul *curriculum learning* per incrementare gradualmente la difficoltà dei prompt usati per addestrare un modello di linguaggio, permettendogli di imparare più velocemente. Infine, si citano applicazioni e strumenti terzi che possono aiutare nella creazione e nell'utilizzo di prompt, come, ad esempio, *LangChain*, una

libreria che aiuta lo sviluppo di applicazioni che combinano LLM con altre risorse.

In Chen et al. (2025: 7), dopo aver citato inizialmente un'alternativa ai già menzionati *Zero-Shot* e *Few-Shot Prompting*, ossia il *One-Shot Prompting*, consistente in un prompt che fornisce al modello un solo esempio per imparare il task, e, dopo aver sottolineato l'uso che si può fare dei parametri di temperatura e di *top-p* (il primo controlla la casualità dell'output modificando la distribuzione di probabilità prima che una parola venga selezionata, quindi più l'indice è basso e più conduce a risultati deterministici; il secondo controlla la casualità dell'output creando un gruppo dinamico di scelte sulla *next-word*, selezionando il set più piccolo dei token più probabili, quindi più l'indice è basso e più i risultati sono prevedibili), si porta l'attenzione sui metodi avanzati di *prompt engineering* sulla scia di quelli citati precedentemente; infatti, di seguito, ne verranno presentati alcuni che aggiungono novità interessanti.

Riguardo il già menzionato CoT, qui si fa riferimento a un nuovo concetto, il *Self-Education via CoT Reasoning* (SECToR), ossia un metodo grazie al quale gli LLM possono insegnarsi autonomamente nuove abilità tramite il CoT e il *reinforcement learning*. Un'altra specifica che approfondisce il CoT si trova nel *Golden CoT*, un metodo che sfrutta un set di soluzioni di CoT basate su verità fondamentali incorporate nel prompt per semplificare il task al modello aggirando la necessità di generare in modo indipendente il CoT. In seguito, il *Least-to-most prompting* scompone un problema complesso in una serie di sotto-problemi affrontati in sequenza: ogni sotto-problema viene risolto a turno e la soluzione di ognuno di questi funge da base per la risoluzione del sotto-problema successivo.

Successivamente, il *Decomposed prompting* (DECOMP) è un approccio modulare pensato per scomporre problemi complessi in sotto-task più semplici; questo metodo fa leva sulla capacità dell'LLM di creare un processo sistematico in cui ogni sotto-task è amministrata da gestori specializzati: in una prima fase il *decomposer* genera un programma di *prompting* per un task complesso; in

seguito, il programma, che è composto a sua volta da una sequenza di fasi, vede in ognuna di queste una sotto-*query* più semplice indirizzata a una funzione in un set ausiliario di funzioni di sotto-task; poi, un *controller* imperativo di alto livello gestisce l'esecuzione del programma, trasferendo gli input e gli output tra il *decomposer* e i gestori di sotto-task fino a che non viene elaborato l'output finale. Per insegnare all'LLM *decomposer* come compiere questo compito, vengono mostrati degli esempi che descrivono come avviene la decomposizione di *query* complesse in sotto-*query* più semplici. Ogni sotto-task è operazionalizzata dai gestori di sotto-task, che possono essere prompt aggiuntivi, funzioni simboliche o apprese dal modello. A proposito di decomposizione, altri metodi possono essere il *Plan-and-Solve Prompting*, uno *Zero-Shot CoT* migliorato che usa una formula come "*Let's first understand the problem and devise a plan to solve it. Then let's carry out the plan and solve the problem step-by-step*", oppure il metodo *Recursion-of-Thought* che, simile al normale CoT, ogni volta che incontra un problema complesso nella sua catena di ragionamento invia il problema con un *call* a un altro prompt o LLM, inserendo poi la risposta nel prompt originale; lo *Skeleton-of-Thoughts* si concentra sul velocizzare la risposta creando uno "scheletro" di risposta, ossia dei sotto-problemi che risolve in parallelo concatenando tutti gli output per la risposta finale; infine, il *Metacognitive Prompting* rispecchia i processi metacognitivi umani creando una catena di prompt in 5 parti che include la spiegazione della domanda, un giudizio preliminare, la valutazione della risposta, la conferma della decisione e la valutazione della *confidence* (Schulhoff et al. 2025: 14).

Successivamente, vengono presentati altri metodi di ottimizzazione dei prompt, come il *Prompt Optimization with Textual Gradients* (ProTeGi) che, ispirato alla tecnica della discesa del gradiente (che, nell'analisi di una funzione di più variabili, trova i punti che risultano ottimali), adatta questo concetto alla natura discreta e non parametrica dell'NLP; invece di affidarsi a gradienti numerici, ProTeGi genera gradienti testuali, ossia descrizioni in linguaggio naturale dei difetti di un prompt in base alle sue prestazioni in un piccolo gruppo

di dati. Questi gradienti indicano la direzione semantica in cui il prompt deve essere migliorato. Inoltre ProTeGi migliora il processo di ottimizzazione applicando questi gradienti testuali per modificare il prompt nella direzione semantica opposta, come nella discesa inversa del gradiente. Questo processo iterativo è guidato da un algoritmo di *beam search* (algoritmo di ricerca basato su euristiche che esplora un grafo espandendo il nodo più promettente in un insieme limitato di nodi) combinato con una strategia di *bandit selection* che esplora efficacemente lo spazio di possibili prompt e seleziona i candidati più promettenti. Il *Black-box prompt optimization* (BPO) si prefigge l'obiettivo di allineare l'LLM con gli scopi degli utenti senza dover attuare un riaddestramento del modello; le tecniche di allineamento tradizionali, come il *Reinforcement Learning from Human Feedback* (RLHF) e il *Direct Preference Optimization* (DPO), richiedono solitamente ingenti risorse computazionali e l'accesso diretto ai parametri del modello, operazione questa non sempre attuabile o efficace, in particolare con modelli a *closed-source*, come GPT-4 o Claude-2.

Per questo motivo è stato introdotto il BPO, dato che sposta l'attenzione da un'ottimizzazione incentrata sul modello a una incentrata sull'input, con l'idea chiave di rifinire il prompt dell'utente invece che alterare i parametri interni del modello; questo approccio fa leva sui *feedback* ottenuti da dati preesistenti che contengono preferenze esplicitate da utenti umani, così da creare coppie di prompt originali e ottimizzati. Queste coppie sono poi usate per addestrare un modello *sequence-to-sequence* pensato per riscrivere il prompt così da migliorare l'allineamento con le aspettative dell'utente. Il BPO è *model-agnostic*, perciò può essere usato con vari LLM, e migliora l'interpretabilità, dato che i cambiamenti effettuati sui prompt sono direttamente osservabili e dimostrano come e perché un determinato prompt dà un allineamento migliore.

Un ulteriore metodo di ottimizzazione dei prompt è il *Model-adaptive Prompt Optimization* (MAPO) che, discostandosi dalla tradizione per cui si lega il prompt a task specifici per migliorare le prestazioni, adatta il prompt a caratteristiche specifiche dei diversi LLM; per questo motivo il MAPO introduce

un processo a due fasi per affrontare la variabilità inerente nei diversi modi in cui gli LLM rispondono allo stesso prompt: la prima fase consiste nello stabilire un *dataset* di *warm-up* in cui i prompt candidati vengono generati e viene valutata la loro idoneità per gli LLM; segue una combinazione di *Supervised Fine-Tuning* (SFT) e *Reinforcement Learning* (RL), usando tecniche come la *Proximal Policy Optimization* (PPO) e il *Ranking Responses from Model Feedback* (RRMF).

In seguito, viene presentato un altro metodo di ottimizzazione dei prompt: il *PromptAgent*. Questo sistema, che utilizza come base la ricerca ad albero Monte Carlo (*Monte Carlo tree search*, MCTS), un algoritmo di ricerca euristica che procede per alberi di decisione, usa un meccanismo di *trial-and-error* ispirato alle strategie umane di risoluzione dei problemi; in questo modo, il modello rifinisce iterativamente i prompt in base ai *feedback* sugli errori e prioritizza i percorsi che conferiscono ricompense più alte. Il sistema di *Reinforcement learning*, già citato in più occasioni, consiste nel rifinire iterativamente i prompt usati durante l'addestramento e l'inferenza definendo una funzione di ricompensa per valutare l'efficacia dei diversi prompt; poi, il modello usa il *feedback* per correggere e ottimizzare i prompt attraverso una serie di iterazioni che massimizzano le prestazioni dei prompt in un task target sfruttando l'abilità del modello di imparare dalle sue iterazioni con l'ambiente.

Vale la pena di citare anche il riferimento che viene fatto, al termine del paragrafo sul *Prompt Optimization*, ai GPT *plugin*, ossia assistenti esterni che aiutano a "levigare" il prompt e sono abili nell'analizzare l'input dell'utente per produrre output pertinenti in un contesto autodefinito. In alcuni usi, la definizione dei *plugin* viene incorporata nel prompt e, alterando il modo in cui il modello interpreta e reagisce ai prompt, dimostra una connessione tra *prompt engineering* e *plugin*. Un'altra importante categoria che viene presentata in Chen et al. (2025: 21) è quella che riguarda il *Retrieval Augmentation*, metodo usato per ridurre le allucinazioni di cui si è già parlato citando Sahoo et al. (2024: 4): quando si usa l'IA generativa è comune rilevare allucinazioni e accadono poiché il modello non ha trovato sufficienti riscontri nei dati di addestramento per rispondere

efficacemente o poiché generalizza alcuni *pattern* mentre cerca di generare un output coerente. Quest'operazione sofisticata, come si è già detto, recupera presso una fonte esterna informazioni che vengono poi usate come conoscenza di base per l'input. Oltre al metodo RAG, vengono citati anche il *Fusion-in-Decoder* (FiD), il metodo *Seq2seq* e l'utilizzo del CoVe.

Per concludere la rassegna delle tecniche di ottimizzazione dei prompt, cito l'elenco di *template* che viene fatto in Schulhoff et al. (2025: 8) e, soprattutto, voglio descrivere quei *template* che ancora non sono stati citati nell'analisi della letteratura precedente. Dopo aver già menzionato le tecniche di *In-Context Learning* (ICL), come, ad esempio, il *Few-Shot Prompting*, le tecniche di *Thought Generation*, come il CoT, e le tecniche di decomposizione, come il DECOMP, si vogliono ora illustrare i *template* che si rifanno all'*Ensembling*, ossia quel processo per cui si usano prompt diversi per risolvere uno stesso problema, aggregando poi le risposte nell'output finale.

Il *Demonstration Ensembling* (DENSE) crea vari prompt con *Few-Shot*, ognuno contenente un distinto sottogruppo di esempi tratti dal set di addestramento, e aggrega i loro output per generare la risposta finale. Il *Mixture of Reasoning Experts* (MoRE) crea un insieme di diversi esperti di ragionamento usando prompt specializzati per differenti tipi di ragionamento. Il *Max Mutual Information Method* crea multipli *template* con vari stili ed esempi, poi seleziona il *template* ottimale che massimizza l'informazione reciproca tra prompt e output dell'LLM. Il DiVeRSe crea multipli prompt, poi esegue la *Self-Consistency* per ognuno di essi generando vari percorsi di ragionamento, che vengono valutati sulla base di ogni loro passaggio. Il *Consistency-based Self-adaptive Prompting* (COSP) costruisce dei prompt con il *Few-Shot CoT* operando lo *Zero-Shot CoT* con la *Self-Consistency* su un set di esempi, poi seleziona un sottoinsieme di output ad alto grado di accordo da inserire nel prompt finale come esempi, infine esegue la *Self-Consistency* con il prompt finale. Lo *Universal Self-adaptive Prompting* (USP) si basa sul COSP e mira a generalizzarlo per ogni task, dato che usa dati non etichettati per generare esempi e una funzione di punteggio più

complessa per selezionarli. Infine, il *Prompt Paraphrasing* è una tecnica di *Data Augmentation* che mantiene il significato generale di un prompt, ma ne cambia le parole.

Un'altra categoria di *template* presentata in Schulhoff et al. (ivi: 15) è quella del *Self-Criticism* che contiene *template* che stimolano l'autocritica dell'LLM nei confronti dei suoi stessi output, espressa tramite un giudizio oppure con un *feedback* utilizzato poi per migliorare la risposta. La *Self-Calibration* invia un prompt all'LLM per avere una risposta a una domanda, poi costruisce un nuovo prompt che include la domanda, la risposta dell'LLM e un'istruzione aggiuntiva che chiede se la risposta è corretta. Il *Self-Refine* è un *framework* iterativo in cui, data una risposta iniziale dell'LLM, invia prompt allo stesso LLM per avere *feedback* sulla risposta e poi chiede al modello di migliorare la risposta basandosi sul *feedback*. Il *Reversing CoT* (RCoT) prima invia un prompt all'LLM per ricostruire il problema a partire dalla risposta generata, poi fa una comparazione *fine-grained* tra il problema originale e quello ricostruito per verificare la presenza di inconsistenze che, in caso ci siano, vengono trasformate in *feedback* per correggere la risposta generata. La *Self-Verification* genera multiple soluzioni con CoT, poi assegna un punteggio a ogni soluzione mascherando alcune parti della domanda originale e chiedendo all'LLM di predirle basandosi sul resto della domanda e sulla soluzione generata. Infine, il *Cumulative Reasoning* prima genera diverse fasi con potenziali risposte alla domanda, poi un LLM decide se accettare o rifiutare le fasi, infine verifica se ha ottenuto la risposta finale e, in caso positivo, termina il processo, in caso contrario lo ripete.

Oltre ai metodi di *prompting* appena presentati, in Schulhoff et al. (ivi: 16) vengono presentate alcune tecniche di *prompt engineering* utilizzate per ottimizzare automaticamente i prompt, come l'*AutoPrompt*, che usa un LLM *frozen* (ossia un modello i cui parametri interni rimangono fissi durante l'uso in task specifici) e un *template* che include alcuni *trigger token* i cui valori vengono aggiornati tramite *backpropagation* durante l'addestramento, o come il

Gradientfree Instructional Prompt Search (GrIPS), metodo simile al già citato APE, ma con in più un set di operazioni più complesse come cancellazione, aggiunta, sostituzione e parafrasi per creare variazioni di una richiesta iniziale.

Cito di seguito altri due metodi interessanti. L'*RLPrompt* usa un LLM *frozen* insieme a un modello non *frozen* per generare *template*, dargli un punteggio su un *dataset* e aggiornare il modulo non *frozen* usando un *Soft Q-Learning*, un algoritmo di *reinforcement learning*. La *Dialogue-comprised Policy-gradient-based Discrete Prompt Optimization* (DP20) coinvolge il *reinforcement learning*, una funzione personalizzata di punteggio e conversazioni con l'LLM per costruire il prompt.

L'ulteriore categoria di *Answer Engineering* (ivi: 18) illustra alcune scelte di design che possono essere fatte per ottimizzare l'output del modello e si divide in tre sezioni: la prima riguarda la forma che la risposta deve avere, il suo formato, come, per esempio, un singolo token, un intervallo di token o un'immagine; la seconda riguarda lo spazio di risposta, ossia il dominio di valori che la struttura può contenere; la terza riguarda l'estrazione della risposta: in caso non sia possibile controllare completamente lo spazio della risposta o la risposta si trovi da qualche parte all'interno dell'output è possibile definire una regola per estrarre la risposta finale, che può essere un *verbalizer*, una *regex* oppure un LLM separato.

Si fa particolare attenzione ai metodi di *prompting* che coinvolgono i modelli multilingue, anche data la disparità di rappresentazione tra la lingua inglese e le altre lingue, in particolar modo le *low-resource language* (ivi: 20). La strategia più semplice che viene presentata è il *Translate First Prompting*, ossia la traduzione di contenuti in lingua inglese per sfruttare l'ampia e diffusa conoscenza di questa lingua in un comprensione migliore del contenuto.

Dopo questo primo metodo, viene presentata l'attuazione di alcune strategie già presentate, ma applicate ai modelli multilingue, come la CoT utilizzata nel *Cross-Lingual Thought* (XLT) e nel *Cross-Lingual Self Consistent Prompting* (CLSP): nel primo si usa un *template* composto da sei istruzioni

indipendenti, tra cui l'assegnamento di un ruolo, il *cross-lingual thinking* e la CoT; nel secondo si introduce un insieme di tecniche che costruisce percorsi di ragionamento in lingue diverse per rispondere a una stessa domanda.

Un altro metodo già osservato, l'*In-Context Learning*, si applica ad altri metodi come l'*X-InSTA Prompting* e il *Cross-lingual Transfer* (In-CLT) *Prompting*: il primo sistema esplora tre approcci distinti per allineare gli esempi in contesto con l'input - l'allineamento semantico con esempi semanticamente simili all'input, l'allineamento *task-based* per esempi che condividono la stessa etichetta dell'input e l'allineamento che combina i due metodi precedenti; la seconda tecnica sfrutta sia la *source language*, sia la *target language* per creare esempi contestuali, contrariamente al metodo tradizionale in cui si usano esempi con la *source language*.

Un sotto-metodo dell'*In-Context Learning* è l'*In-Context Example Selection*, poiché è fondamentale trovare esempi in contesto che siano semanticamente simili alla fonte; tuttavia, anche utilizzare esempi dissimili può migliorare le *performance* e, in interazione con frasi potenzialmente ambigue, utilizzare esempi con parole polisemiche o desuete può migliorare le prestazioni; in questo contesto, il sistema PARC (*Prompts Augmented by Retrieval Cross-lingually*) introduce un *framework* che recupera esempi rilevanti da una lingua *high resourced*. Successivamente, un'altra caratteristica di cui tenere conto nell'utilizzo di modelli multilingue è la selezione della lingua da utilizzare per il *template*, perché in base a queste le *performance* del modello possono cambiare radicalmente.

Creare un prompt in inglese dà spesso risultati migliori rispetto al crearlo con la lingua originaria del task che si sta svolgendo, data la predominanza dei dati in inglese nel pre-addestramento. Però, in molti casi, i *benchmark* di *prompting* multilingue usano prompt con la lingua del task per casi d'uso specifici della lingua in questione, perciò i risultati dei *template* possono differire in base ai task e ai modelli. Oltre ai sistemi sopra descritti, esistono altri metodi utilizzati per migliorare le traduzioni operate dalle IA generative: il *Multi-Aspect*

Prompting and Selection (MAPS) imita il processo di traduzione umana e prevede più fasi preparatorie a cominciare dall'estrazione di conoscenze (parole chiave e argomenti, insieme alla generazione di esempi di traduzione) dalla frase fonte, seguita dalla realizzazione di multiple possibili traduzioni e dalla scelta della più accurata; il metodo *Chain-of-Dictionary* (CoD) prima estrae parole da una frase fonte, poi genera una lista dei loro significati in diverse lingue in modo automatico tramite il recupero di informazioni da un dizionario, poi si antepongono le conoscenze derivate dal dizionario al prompt, chiedendo esplicitamente al modello di usarle durante la traduzione; un caso simile al CoD è il *Dictionary-based Prompting for Machine Translation* (DiPMT) che fornisce solo definizioni nella lingua fonte e in quella target e le formatta in modo diverso; il *Decomposed Prompting for MT* (DecoMT) divide il testo fonte in diverse parti e le traduce in modo indipendente usando il *few-shot prompting*, poi usa le traduzioni generate e le informazioni contestuali tra le varie parti per creare la traduzione finale.

Riguardo gli approcci *human-in-the-loop*, ossia che prevedono la presenza attiva di un operatore umano nei processi che seguono la creazione del prompt originale, l'*Interactive-Chain-Prompting* (ICP) risolve le potenziali ambiguità nella traduzione chiedendo prima al modello di fare delle sotto-domande sulle parti non chiare nella frase da tradurre, cosicché gli operatori umani possano rispondere a queste domande e il sistema utilizzi le informazioni ottenute nella generazione della traduzione finale; l'*Iterative Prompting* consiste nell'inviare un primo prompt all'LLM per ottenere una bozza di traduzione, che viene poi rifinita integrando informazioni ottenute da sistemi di recupero automatico o dal *feedback* diretto degli operatori umani.

Al termine di questo paragrafo, in Schulhoff et al. (ivi: 31) vengono presentati modi per migliorare l'allineamento tra l'output generato effettivamente dal modello e le richieste dell'utente, espresse per mezzo del prompt testuale. In particolare, l'output generato deve presentare il meno possibile contenuti dannosi, risposte inconsistenti o affette da *bias* di qualsiasi tipo. Per questo

motivo, si parla di *prompt sensitivity*, ossia la caratteristica degli LLM di essere molto sensibili ai prompt e di presentare output potenzialmente molto diversi anche solo modificando l'ordine degli elementi presenti nel prompt. Il fenomeno di *Small Changes in the Prompt* descrive come spazi aggiuntivi inseriti nel prompt, lettere maiuscole diverse, delimitatori differenti o sinonimi possano impattare in modo significativo le *performance* del modello.

Il *Task Format* descrive modi diversi di redigere un prompt per far eseguire a un LLM lo stesso task, permettendo al modello di focalizzare l'attenzione su dettagli diversi. Infine, il *Prompt Drift* è un fenomeno che accade nel momento in cui il modello alla base di una API viene modificato nel tempo, perciò uno stesso prompt può produrre risultati diversi nelle due o più versioni diverse del modello; motivo per il quale necessita di un costante monitoraggio e aggiornamento.

Un'altra caratteristica propria degli LLM, insieme alla *sensitivity*, è l'*overconfidence*, ossia l'eccessiva "sicurezza" che un modello dimostra di avere nei confronti dei suoi stessi output, causa di un'eccessiva fiducia che anche gli utenti possono avere verso le risposte degli LLM. Per risolvere questo problema, è necessario attuare la calibrazione della *confidence*, che restituisce un punteggio che rappresenta la *confidence* del modello e può essere effettuata con le seguenti tecniche di *prompting*: il *Verbalized Score* è una semplice tecnica di calibrazione che consiste nel chiedere direttamente all'LLM di generare un punteggio sulla sua stessa *confidence*, ma la sua efficacia è dubbia, dato che, come si è osservato, i modelli solitamente sono troppo sicuri dei loro stessi output; al contrario prompt semplici possono ottenere una calibrazione più accurata rispetto rispetto al calcolo delle probabilità dei token di output del modello, indicato come la soluzione più naturale e diretta per la calibrazione della *confidence*.

La *sycophancy* è un altro concetto collegato all'*overconfidence* e descrive il fenomeno in cui l'LLM è d'accordo con l'utente anche quando si contraddice l'output iniziale del modello stesso. Ad esempio, se si chiede al modello di esprimere un'opinione su un argomento e viene inclusa nel prompt l'opinione

personale dell'utente, quest'ultima può influenzare l'output del modello, soprattutto nei modelli di maggiori dimensioni e in quelli *instruction-tuned*. Pertanto, per evitare questa influenza, è necessario escludere le opinioni personali dell'utente in fase di redazione del prompt.

Successivamente, in Schulhoff et al. (ivi: 32), viene affrontato il problema dei *bias*, degli stereotipi e delle influenze culturali che si trovano nei dati, ponendo l'attenzione sui metodi per arginare la perpetuazione di queste parzialità nei dati e rendere gli output dei modelli più equi per tutti gli utenti. A questo proposito, si citano il *Vanilla Prompting* (o *moral self-correction*), ossia una semplice istruzione inserita nel prompt che indica al modello di fare attenzione a non usare *bias*, il *Selecting Balanced Demonstrations*, ovvero l'ottenimento di dimostrazioni ottimizzate secondo metriche di equità per ridurre i *bias*, la *Cultural Awareness*, cioè l'uso di una serie di prompt che, con l'inclusione della richiesta di rifinitura dell'output e di istruzioni sull'uso di parole culturalmente rilevanti, tentano di agevolare l'LLM nell'adattamento culturale e, infine, l'*AttrPrompt,* una tecnica di *prompting* che, per evitare *bias*, chiede all'LLM di generare attributi specifici importanti da modificare per tenere sotto controllo la diversità nei dati e, successivamente, di generare dati sintetici variando ognuno di questi attributi.

Infine, in Schulhoff et al. (ibidem), viene analizzato anche il problema dell'ambiguità, ossia delle diverse interpretazioni che il modello può effettuare nei confronti di una domanda e delle conseguenti risposte diverse che può potenzialmente dare. Le *Ambiguous Demonstrations* sono esempi che hanno un set di etichette ambiguo e includerli nel prompt può migliorare le perfomance nell'ICL, mentre la *Question Clarification* permette all'LLM di identificare ambiguità nel prompt e genera domande chiarificatrici da porre all'utente per fugare eventuali dubbi; una volta che le ambiguità vengono chiarite dall'utente, il modello può rigenerare la risposta.

Nell'articolo di Schulhoff et al. (2025) menzionato sopra (ivi: 26), una componente fondamentale per il presente lavoro è illustrata nella sezione in cui si

esplicano le potenzialità degli LLM di competere con i valutatori nel giudicare la qualità di un testo o anche dell'output prodotto da un modello precedente in accordo con determinate metriche presentate nel prompt. Vengono descritte quattro componenti del *framework* di valutazione: le tecniche di *prompting* usate, il formato dell'output della valutazione, il *framework* della *pipeline* di valutazione e, infine, scelte alternative di design metodologico.

Riguardo le tecniche di *prompting* impiegate, oltre all'ICL, che viene frequentemente utilizzato nei prompt di valutazione, si suggerisce l'uso della valutazione *role-based*, che presenta identiche istruzioni di valutazione, ma che affida ruoli diversi nei vari prompt, per analizzare le differenze nelle valutazioni generate. Dopo aver citato il metodo del CoT che può contribuire a migliorare le *performance* di valutazione, si descrivono, infine, le *model-generated guidelines*, ossia linee guida per la valutazione generate all'LLM stesso; questo metodo riduce il problema dell'*insufficient prompting* che nasce da linee guida e spazi di output mal definiti. Riguardo il formato dell'output di valutazione, questo può modificare significativamente le *performance* della valutazione: lo *styling* dell'output impostato come XML o come JSON può migliorare l'accuratezza del giudizio generato dal valutatore; per quanto riguarda il formato del punteggio che si può utilizzare, la scala lineare è un formato di output molto semplice (ad esempio, si imposta un punteggio che va da 1 a 5, o da 1 a 10, e così via) e si può richiedere al modello di produrre un output con un punteggio discreto o continuo; il punteggio prodotto dal modello può anche essere di tipo binario, come sì/no o vero/falso; infine, si può utilizzare anche la scala Likert, che presenta una scala di atteggiamenti positivi o negativi esplicitati verbalmente. Per quanto riguarda il *framework* della *pipeline* di valutazione tramite *prompting*, vengono citati tre differenti sistemi: LLM-EVAL, che usa un singolo prompt che contiene uno schema di variabili da valutare, un'istruzione che descrive al modello i punteggi di output per ogni variabile entro un determinato range e il contenuto da valutare; G-EVAL, che somiglia a LLM-EVAL, ma include passaggi di *Auto-CoT* nel prompt, passaggi generati in accordo con le istruzioni di valutazione e inseriti nel

prompt finale; infine, ChatEval, che usa un *framework multi-agent* in cui ogni *agent* ha un ruolo separato. Infine, riguardo le scelte alternative di design metodologico, si illustra la differenza tra un approccio esplicito, in cui si richiede direttamente all'LLM di generare valutazioni sulla qualità di uno scritto, e un approccio di punteggio implicito in cui, ad esempio, il punteggio sulla qualità può essere derivato usando la *confidence* del modello nella sua previsione, o la probabilità di generare l'output, o tramite la spiegazione dei modelli (come il conteggio di errori), o con valutazione su task *proxy* (inconsistenza fattuale tramite implicazione); un altro stratagemma citato è il *Batch Prompting*, ossia l'uso di un singolo prompt che contiene diversi elementi da valutare contemporaneamente, oppure uno stesso elemento viene valutato con criteri o ruoli diversi, anche se valutare elementi multipli in un singolo prompt spesso peggiora le *performance* (Schulhoff et al. 2025: 28); infine, la *Pairwise Evaluation*, che consiste nel richiedere esplicitamente all'LLM di generare un punteggio individuale per i riassunti, in modo tale da evitare comparazioni dirette tra due testi, dato che possono portare a risultati peggiori.

1.1.3. *Meta-prompting*

Prima di procedere con l'analisi delle tecniche che vengono usate per valutare la qualità dei prompt in modo preciso e quantificabile e di descrivere il ruolo che gli LLM stanno assumendo come "giudici", ossia come valutatori di materiale prodotto da terzi che necessita un giudizio automatizzato, voglio esplorare un'ultima declinazione delle tecniche di *prompting*: il *meta-prompting*. Questo particolare sistema istruisce l'LLM per eseguire diversi passaggi in una sorta di *pipeline*: in un primo momento divide task complesse in parti più semplici; poi, assegna le parti semplificate a modelli "esperti" specializzati, addestrati con istruzioni dettagliate riguardo il linguaggio naturale; successivamente, coordina e supervisiona la comunicazione tra i diversi esperti interpellati; infine, durante tutto il processo, applica le proprie capacità critiche di ragionamento e verifica

(Suzgun & Kalai 2024: 2). Quando gli viene sottoposta una *query*, il modello istruito con meta-prompt funziona da conduttore e produce una cronologia comprensiva di tutte le risposte tratte dai vari esperti; il modello non solo è responsabile della cronologia che viene generata e che include la selezione degli esperti e le istruzioni specifiche a loro sottoposte, ma lavora anche come esperto indipendente, generando output basati sulla sua competenza e sull'informazione scelta dal conduttore per ogni *query* (ibidem). In questo modo, il modello mantiene una singola linea di ragionamento che rimane coerente nonostante le "chiamate" rivolte ai vari esperti e permette al modello di funzionare in modo efficace sia come conduttore centrale sia come gruppo di esperti per produrre risposte più accurate.

Mentre un singolo modello generico può dare soluzioni utili per *query* generiche, combinare i punti di vista e le conclusioni di multipli modelli che operano su specifici domini può fornire risposte più robuste e articolate (ivi: 3). Il modello principale (quello che in precedenza è stato definito conduttore) viene definito *meta model* e ricopre il ruolo più importante nella struttura gerarchica che gestisce la *pipeline* di *meta-prompting*: ha il compito di combinare le soluzioni fornite dagli altri modelli per mettere a punto una risposta accurata a uno specifico problema, supervisionando accuratamente i modelli esperti che operano sotto il suo controllo (ivi: 4). I modelli esperti possono essere chiamati solo dal *meta model* e non possono comunicare fra di loro, anche se il modello principale può condividere o combinare le informazioni derivanti da esperti quando interagisce con nuovi esperti.

Un'altra caratteristica fondamentale presente in questa tecnica è rappresentata dal concetto di *fresh eyes*: la possibilità di incorporare prospettive nuove grazie al lavoro dei modelli esperti permette di attenuare problemi come l'*overconfidence* e la perpetuazione di errori; il *meta-prompting*, includendo a ogni passaggio i punti di vista nuovi e diversi dei modelli esperti (che non possono accedere alla cronologia completa delle informazioni recepite dal *meta*

model), non trovano solo soluzioni, ma identificano e correggono gli errori effettuati dagli esperti precedenti (ivi: 10).

In Zhang et al. (2025: 2), il *Meta Prompting* nasce dalla volontà di architettare un sistema più strutturato rispetto al *prompting* comune basato sul contenuto: il bisogno di una procedura formale porta alla creazione di categorie di task e categorie di prompt, così che le strategie di *problem-solving* possano essere mappate su strutture di prompt modulari e riutilizzabili. Un'ulteriore innovazione è *Recursive Meta Prompting* (RMP) che istruisce l'LLM affinché generi e rifinisca i suoi stessi prompt, in modo tale da migliorare le sue stesse strategie di *problem-solving*.

1.1.4. Tecniche di valutazione della qualità dei prompt

Illustrerò ora quali sono i criteri e i sistemi oggettivi che permettono di mettere in luce caratteristiche comuni dei prompt e degli output prodotti tramite il loro utilizzo per valutare in modo chiaro e univoco le prestazioni dei prompt. In Chen et al. (2025: 26) si opera un'iniziale divisione tra metodi di valutazione soggettivi e oggettivi, di cui si consiglia un utilizzo coordinato. La principale differenza tra le due categorie consiste nell'impiego di valutatori umani in quella soggettiva e nell'impiego di "metodi di valutazione automatica" tramite algoritmi in quella oggettiva. In particolare, con quest'ultimo metodo, è possibile sia giudicare la qualità dell'output generato da un LLM, sia misurare quantitativamente l'efficacia dei metodi di *prompting* testandoli sui *benchmark*. Entrambe le categorie hanno vantaggi e svantaggi: la valutazione soggettiva è più costosa e dispendiosa in termini di tempo, mentre quella oggettiva è meno costosa e più rapida, ma manca della componente di intuizione umana presente, invece, nella prima. Il modo migliore per utilizzare questi due metodi e la loro interazione dipende dagli specifici casi di applicazione. Le valutazioni soggettive, come detto, dipendono da valutatori umani che analizzano un testo e lo giudicano tenendo in considerazione aspetti di un testo come la fluenza, l'accuratezza,

l'originalità e la pertinenza. In un esempio citato, si usano valutazioni umane in interazione con il metodo *Chain of Density* che consiste, attraverso un prompt apposito, nella produzione di riassunti con un livello di densità di entità linguistiche sempre più alto; successivamente, si mescolano casualmente questi riassunti generati con altri riassunti prodotti da esseri umani e si fanno giudicare dai valutatori umani. In generale, le valutazioni soggettive sono sempre più utilizzate per valutare il contenuto generato da modelli in aree più astratte e più difficili da rappresentare con *dataset*, come la scrittura o i riassunti.

Riguardo le valutazioni oggettive, come anticipato sopra, esse consistono nell'uso di algoritmi per valutare automaticamente la qualità dell'output generato da un LLM e, in Chen et al. (ivi: 27), si fa menzione di alcuni di questi metodi, come il *Human-AI Language-based Interaction Evaluation* (*HALIE*), che si concentra sull'interazione tra LM e apporto umano, oppure il *BiLingual Evaluation Understudy* (*BLEU*), che assegna, tramite metriche automatiche, un punteggio agli ouput generati dal sistema in modo da comparare sistemi diversi e controllare i loro progressi. Altri sistemi citati sono il *Recall-Oriented Understudy for Gisting Evaluation* (*ROUGE*) e il *Metric for Evaluation of Translation with Explicit ORdering* (METEOR) che valutano la similarità tra il testo generato e il testo di riferimento, e BERTScore, che compie lo stesso lavoro, ma su un livello semantico superiore, sfruttando gli *embeddings* contestuali di modelli pre-addestrati come BERT. Tuttavia, questo tipo di metriche automatiche va usato con cautela, dato che non riesce pienamente a cogliere i risultati delle valutazioni fatte da valutatori umani.

Molti ricercatori valutano i loro metodi quantificando le prestazioni in task specifici, come il Gioco del 24 o il Cruciverba 5x5; gli altri task, i già citati *benchmark*, sono *dataset* che contengono istruzioni per la realizzazione dei modelli. Oltre a *Beyond the Imitation Game benchmark* (BIGbench) e *Big-Bench Hard* (BBH), che sono set completi di *benchmark* e valutano la coerenza logica degli argomenti, vengono citati altri quattro tipi di *benchmark* che forniscono

task e *dataset* standardizzati che agevolano valutazioni coerenti e comparabili tra approcci diversi.

Il primo di questi quattro tipi riguarda i *Math Word Problems* (MWPs) che testano la capacità di un modello di comprendere domande legate ai numeri; questo tipo di task è complesso da risolvere poiché il modello deve sia comprendere informazioni tratte dal testo in linguaggio naturale, sia compiere ragionamenti matematici per risolverlo. La complessità dell'MWPs può essere misurata lungo molti assi, ad esempio la complessità linguistica e di ragionamento e la conoscenza del mondo e del dominio. Si citano, a titolo di esempio, il *Simple Variations on Arithmetic Math word Problems* (SVAMP), che serve a risolvere problemi di matematica di livello elementare, e GSM8K, che invece richiede ai modelli di risolvere problemi matematici complessi, sottolineando la necessità di comprendere in modo approfondito i concetti matematici e il ragionamento che vi sta alla base.

Il secondo dei quattro tipi si riferisce ai *Question Answering* (QA) *Tasks*, che richiedono ai modelli di fornire *feedback* sulle domande poste; ad esempio, il *Massive Multitask Language Understanding* (MMLU) è un *benchmark* di QA pensato per misurare le conoscenze acquisite durante il pre-addestramento valutando i modelli esclusivamente in contesti di *zero-shot* e *few-shot*. Alcuni *benchmark* di QA sono legati a task basati sulle conoscenze, come il *Fact Extraction and VERification* (FEVER) che si concentra sulla verifica dei fatti richiedendo ai modelli di agire per le affermazioni generate dall'alterazione di frasi estratte da Wikipedia. I *benchmark* di QA mettono alla prova l'abilità di ragionamento e di uso delle conoscenze di senso comune.

Il terzo dei quattro tipi fa riferimento ai *Language Understanding Tasks*: prima di tutto, viene citato il *Text REtrieval Conference* (TREC), funzionale per il recupero di risposte piuttosto che degli elenchi di documenti; la *Stanford Sentiment Treebank* (SST) è costruita con alberi di analisi sintattica completamente etichettati, consentendo un'analisi completa degli effetti compositivi del *sentiment* nel linguaggio; altri *benchmark* citati sono, ad

esempio, il "*Less Likely Brainstorming*", è un *benchmark* che opera analisi chiedendo al modello di generare output che gli esseri umani ritengono rilevanti ma meno probabili, oppure il *SAlient Long-Tail Translation Error Detection* (SALTED), che si concentra sull'identificare errori nelle traduzioni. Le valutazioni prodotte da questi sistemi evidenziano la capacità dei modelli di comprendere ed elaborare un testo, effettuando previsioni accurate basate sul contenuto.

Infine, l'ultimo dei quattro tipi di *benchmark* citati in Chen et al. (ivi: 29) riguarda i *Multimodal Tasks*, progettati per valutare le abilità degli MMLM di elaborare e integrare informazioni da più fonti, come testo e immagini; si citano RefCOCO, RefCOCO+ e RefCOCOg, che forniscono espressioni di riferimento per gli oggetti nelle immagini, testando la capacità dei modelli di collegare le descrizioni con il contenuto visivo. Queste valutazioni sono fondamentali per lo sviluppo di modelli in grado di comprendere e interagire in modo intermodale, essenziali per applicazioni come la risposta a domande visive e la didascalia delle immagini.

Oltre ai metodi di *benchmarking* soggettivi e oggettivi citati sopra, in Chen et al. (ibidem) vengono introdotti anche i sistemi per comparare diversi metodi di *prompting* tra loro in base ai punteggi delle loro prestazioni, usati poi come *benchmark* per valutare i modelli; ad esempio, *LLM-Eval* è stato sviluppato per valutare le conversazioni a dominio aperto con gli LLM utilizzando vari *dataset* come, ad esempio, *Dynabench*. Oltre a confrontare metodi diversi di *prompting* tramite punteggio, si possono usare altri indicatori per avere informazioni aggiuntive, come la *prediction accuracy* e la *proof accuracy*, oppure il costo economico, la velocità di valutazione o altri ancora. Nella comparazione di metodi di *prompting*, si possono usare anche comparazioni soggettive con valutatori umani per osservare l'allineamento dei metodi al tipo di ragionamento umano.

Infine, si cita anche il *framework* di valutazione generale *InstructEval*, che consente una valutazione completa delle tecniche di *prompting* su più modelli e

task; attraverso questo metodo si sono raggiunte le seguenti conclusioni: in contesti *few-shot*, l'omissione di prompt o l'utilizzo di prompt generici *task-agnostic* tendono a superare altri metodi, con i prompt che hanno un impatto minimo sulle prestazioni; in contesti *zero-shot*, i prompt specifici per i task scritti da esperti possono aumentare significativamente le prestazioni, con i prompt automatizzati che non superano le *baseline* più semplici; infine, le prestazioni dei metodi automatizzati di generazione dei prompt sono incoerenti, variano a seconda dei diversi modelli e dei tipi di task e mostrano una scarsità di generalizzazione (Chen et al. 2025: 30).

In Schulhoff et al. (2025: 33) vengono comprate tra loro diverse tecniche di *prompting*, impostando come base un *template* che evidenzia le varie parti del prompt e poi variandole usando sei metodi diversi (*Zero-Shot*, tecniche diverse di *Zero-Shot-CoT* e le stesse tecniche, ma con il *Few-Shot*). Poi viene calcolata l'*accuracy* delle diverse tecniche e i risultati vengono confrontati tra loro. Vengono inoltre usati due formati diversi per presentare le domande poste al modello: uno in cui le opzioni vengono presentate consecutivamente sulla stessa riga, un altro in cui vengono poste in colonna.

Capitolo 2 LLM*s-as-judges* e il task di valutazione

Guardando più da vicino l'oggetto principale della presente ricerca e anticipando ciò che verrà esposto più nel dettaglio nel secondo capitolo, è necessario specificare meglio le caratteristiche degli LLM*s-as-judges* (anche denominati LLM*s judges*, LLM-*as-a-judge* o LLM*s-based judges*), di cui in Li et al. (2024: 3) si fa una descrizione chiara:

> *The LLMs-as-judges paradigm is a flexible and powerful evaluation framework where LLMs are employed as evaluative tools, responsible for assessing the quality, relevance, and effectiveness of generated outputs based on defined evaluation criteria.*

perciò consistono nella valutazione degli output dei modelli non più attraverso *benchmark* fissi (come si è visto nel paragrafo 1.1.4), ma tramite modelli che garantiscono modalità di analisi flessibili e adattive, con la possibilità di dividere la valutazione in criteri separati e indipendenti. Nonostante le valutazioni umane rispecchino il *gold standard* nella valutazione dei modelli, affidarsi agli LLM-*as-judges* permette di risparmiare denaro e tempo, oltre a servire come modelli di ricompensa durante l'addestramento per migliorare i modelli e come verificatori durante l'inferenza per selezionare la risposta migliore tra gli output candidati (Tan et al. 2025: 1).

Nonostante l'alta qualità delle valutazioni che gli LLM-*as-judges* possono dare su task complessi, questi non sono esenti da *bias* e da debolezze strutturali, motivo per cui è necessario che vengano progettati in modo rigoroso e, soprattutto, standardizzato (Gu et al. 2025; 1). In Li et al. (2024: 5) vengono presentate tre configurazioni principali della funzione di valutazione.

Il *single-LLM evaluation system* vede l'attività di un singolo modello che valuta l'output; permette un utilizzo semplice in interazione con task generici, ma ha una flessibilità limitata e si trova in difficoltà nell'affrontare compiti troppo complessi, con il rischio di introdurre *bias* se non opportunamente addestrato;

consta di tre componenti basilari, che sono il *prompt engineering* (che sfrutta l'ICL, il ragionamento *step-by-step*, la *definition augmentation* e la *multi-turn optimization*), il tuning (che può essere *score-based* oppure *preference-based*) e il *post-processing* (che consiste in un ulteriore affinamento dei risultati della valutazione controllando che siano non solo accurati, ma anche ben allineati col task, e può incorporare la *probability calibration* e il *text reprocessing*).

Il secondo tipo di configurazione, il *multi-LLM evaluation system*, integra multipli modelli che possono collaborare nella valutazione o competere tra di loro, garantendo valori aggiunti in entrambi i casi: la "comunicazione" tra i diversi modelli, se è del primo tipo citato (*cooperation*), consiste nella condivisione di informazioni per raggiungere l'obiettivo comune, se invece è del secondo tipo (*competition*), vede il confronto tra i diversi modelli in una struttura centralizzata (con un solo modello che dirige il lavoro degli altri), o decentralizzata (in cui tutti i modelli hanno uno statuto di pari grado e creano una struttura decisionale distribuita); se, invece, tra i modelli non c'è una vera e propria comunicazione, i giudizi vengono generati dai modelli in modo indipendenti e vengono poi sintetizzati in una decisione finale attraverso strategie di aggregazione (come, ad esempio, il voto di maggioranza, le medie ponderate o il dare priorità alle previsioni con il più alto grado di affidabilità).

Il *multi-LLM evaluation system* permette un'analisi più approfondita e con una quantità maggiore di criteri di valutazione rispetto al *single-LLM evaluation system*, ma ha un costo più elevato e richiede più risorse computazionali (Li et al. 2024: 20). Infine, il terzo tipo di configurazione, lo *human-AI collaboration system*, integra sia la valutazione automatica, sia valutatori umani che collaborano col modello; incorporando il punto di vista umano, i sistemi di questo tipo possono garantire un giudizio finale più affidabile e possono migliorare le prestazioni del modello attraverso un *loop* di *feedback*. Per quanto riguarda l'input di valutazione, oltre all'oggetto vero e proprio della valutazione, viene descritto anche il tipo di valutazione, ossia se *pointwise*, che valuta ogni oggetto individualmente, pairwise, in cui compara due oggetti, oppure *listwise*, in

cui valuta collettivamente un'intera lista di oggetti. Insieme al tipo, vengono descritti anche i criteri di valutazione e i riferimenti: i primi sono gli standard specifici di valutazione e rispecchiano gli attributi legati al task (tipicamente, comprendono la qualità linguistica, l'accuratezza del contenuto e alcune metriche specifiche del task); i secondi, possono essere presenti come dati su cui si basa l'LLM-*as-judge* per valutare le prestazioni (modello di valutazione *reference-based*), in caso contrario il valutatore giudica in base a standard di qualità intrinseci o in base all'allineamento con un contesto originario (modello di valutazione *reference-free*) (Li et al. 2024: 8). L'output di valutazione consta di "risultato", ossia l'output primario della valutazione, "spiegazione", che contiene il ragionamento seguito per giungere al risultato, e *feedback*, con suggerimenti e raccomandazioni utili per rifinire il contenuto.

Il processo per valutare un valutatore (*meta-evaluation*) è fondamentale per comprendere l'affidabilità degli LLM-*as-judges*. Può essere effettuato in due modi principali: attraverso il semplice utilizzo di *benchmark* (come si è già visto in precedenza), citati in Li et al. (2024: 27) in ordine di task, oppure con l'uso di metriche, come, a titolo di esempio, il calcolo dell'*accuracy*, il kappa di Cohen o il coefficiente di correlazione di Pearson[2].

Un altro argomento fondamentale di cui è necessario tenere conto riguarda la diffusione di *bias* e le fragilità inerenti i modelli e il loro utilizzo del tipo citato sopra. In Li et al. (2024: 36) e in Gu et al. (2025: 21) vengono presentate due classificazioni diverse dei *bias* e dei limiti incontrati nell'utilizzo di LLM-*as-judges* (una per cause, l'altra per aderenza a task specifici), ma la maggior parte di questi fenomeni viene citata in entrambi i lavori e di seguito se ne fa un breve riassunto.

Il *position bias* consiste nella tendenza da parte del modello a preferire le risposte posizionate in punti precisi del prompt e per misurarlo si usano quattro

[2] L'*accuracy* (precisione) può essere calcolata dividendo le risposte corrette per il totale degli esempi presentati; il kappa di Cohen è un coefficiente statistico che mostra il grado di accordo tra due valutazioni qualitative una volta ottenuti i riscontri positivi, negativi, falsi positivi e falsi negativi posizionati in una matrice di confusione; il coefficiente di correlazione di Pearson consiste nella covarianza divisa per il prodotto delle deviazioni standard di due variabili, così da misurare la correlazione lineare tra queste.

metriche differenti: la *position consistency*, che quantifica la frequenza con cui il modello sceglie una stessa risposta dopo che questa ha cambiato posizione, la *preference fairness*, che misura il modo in cui il modello preferisce le risposte in determinate posizioni, il *conflict rate*, che misura la percentuale di disaccordo dopo il cambio di posizione di due risposte diverse, e la *repetition stability*, usate per distinguere un *position bias* sistematico da casuali oscillazioni nei risultati; in Li et al. (2024: 38) vengono citati tre metodi per mitigare il *position bias* e sono i seguenti: il primo metodo, *swap-based*, si occupa di scambiare la posizione delle risposte e, nella versione del metodo basata sul punteggio, vengono valutate tutte le risposte e si tiene conto del punteggio medio conteggiato dopo vari scambi come punteggio finale della risposta, mentre, nella versione del metodo basata sulle comparazioni, si analizza il divario di qualità tra le varie risposte; il secondo metodo, detto *alignment-based*, simula strategie di comparazione umane dividendo le risposte in segmenti e allineandole per contenuti simili; l'ultimo metodo, detto *discussion-based*, che incorpora *peer ranking* e discussioni per migliorare l'accuratezza della valutazione.

Il *verbosity bias*, che in Gu et al. (2025: 22) viene inserito come caso specifico del *length bias*[3], descrive la tendenza del modello a preferire risposte più estese, al di là della qualità del contenuto, e viene mitigato con metodi che sforzano il modello a concentrarsi sul contenuto e sulla sua pertinenza.

L'*authority bias*, che in Gu et al. (ibidem) viene descritto come caso specifico del *concreteness bias* insieme al *citation bias*, consiste nel fenomeno per cui il valutatore preferisce risposte che contengono dettagli specifici, come citazioni da fonti autorevoli, valori numerici o termini complessi e tecnici, indipendentemente dalle prove concrete che possono sostenere tali risposte e dalla validità generale del contenuto; un metodo citato per risolvere tale *bias* è tramite l'utilizzo di RAG per verificare la validità delle fonti autorevoli attraverso basi di conoscenza esterne.

[3] Rappresenta la tendenza da parte del modello a preferire risposte di una lunghezza specifica.

Il *bandwagon-effect bias* descrive la tendenza che hanno i modelli ad allineare i propri giudizi alle tendenze o alle opinioni più condivise, senza curarsi della qualità effettiva o della correttezza del contenuto oggetto di valutazione; di conseguenza, quando alcune risposte vengono presentate come aventi consenso popolare il modello può erroneamente preferirle, anche se il consenso è falso o frutto di *bias*. Una soluzione citata consiste nell'anonimizzare le informazioni su opinioni terze, in modo tale che il modello si concentri solo sulla qualità formale dell'input.

Il *compassion-fade bias* descrive il fenomeno per cui il modello valutatore può venir fuorviato dalla menzione esplicita del nome del modello che fornisce la risposta (ad esempio, se si specifica che il modello di riferimento è GPT-4 il valutatore può essere indotto a dare una risposta più positiva a causa della fama di qualità del modello).

Il *diversity bias* si riferisce alla tendenza del modello di modificare i suoi giudizi in base a elementi legati all'identità, come il genere, l'etnia, l'orientamento sessuale, l'appartenenza a gruppi demografici o religiosi; il modello, a causa di questo *bias*, può dimostrare di trattare in modo diseguale le identità basandosi su stereotipi impliciti, perciò è fondamentale assicurare l'equità e l'inclusività nei giudizi espressi dal valutatore.

Il *cultural bias* descrive il fenomeno in cui il modello può interpretare in modo fallace espressioni che appartengono a culture diverse rispetto a quella di riferimento rappresentata nei dati di addestramento, oppure il fenomeno in cui il modello fallisce nel riconoscere varianti regionali delle lingue.

Il *sentiment bias*, che in Gu et al. (2025: 22) viene classificato come caso specifico dello *style bias*, descrive la tendenza dei modelli a preferire risposte che hanno un tono particolarmente emotivo, oppure che hanno un contenuto allegro od ottimistico, senza tener conto dell'effettiva qualità del contenuto, al contrario il modello è portato a penalizzare un contenuto emotivamente contrassegnato come negativo o troppo intenso. Lo *style bias*, invece, descrive la mera preferenza di contenuti esteticamente curati, preferendo la forma al contenuto.

Il *token bias* consiste nella preferenza da parte del modello di alcuni token a causa della priorità che si dà nei dati di pre-addestramento ai token che occorrono più frequentemente, senza contare la loro appropriatezza o correttezza contestuale.

Il *contextual bias* si riferisce alla tendenza dei modelli di fornire giudizi imprecisi in base al contesto specifico in cui sono usati, compresi eventuali esempi contestuali forniti al modello.

L'*overconfidence bias* (di cui si è già parlato nel paragrafo 1.1.2) rispecchia l'eccessiva fiducia che i modelli dimostrano rispetto alle risposte che danno, senza curarsi di verificare l'effettiva affidabilità della risposta; per affrontare questo *bias* vengono citati alcuni sistemi, tra cui il *Cascaded Selective Evaluation*, in cui degli annotatori simulati stimano la *confidence* del modello, oppure altri sistemi in cui più LLM si scontrano in un dibattito per discutere risultati diversi.

Il *self-enhancement bias* consiste nella propensione di un modello a preferire i suoi stessi output rispetto a quelli di altri modelli; due sistemi scelti d'esempio per arginare questo *bias* sono il *peer rank* (PR) e la *peer discussion* (PD): il primo sistema usa più LLM come revisori, ciascuno dei quali valuta con comparazioni pairwise le risposte provenienti da diversi modelli, aggregando poi le risposte e pesandole in base alla *consistency* con giudizi umani; il secondo sistema stimola il dialogo tra due LLM per giungere a un accordo comune sulla loro preferenza tra due risposte, incoraggiando i modelli a valutare nuovamente le loro posizioni iniziali e a prendere in considerazione diverse prospettive.

Il *refinement-aware bias* richiama la propensione di un modello a valutare diversamente le risposte se sono originali rifinite oppure se includono la cronologia delle revisioni effettuate; una risposta modificata più volte può venire interpretata più positivamente rispetto all'originale, anche se non sono stati apportati reali miglioramenti alla qualità del contenuto. Una delle possibili soluzioni è incorporare un meccanismo di *feedback* esterno durante la

valutazione, introducendo un giudizio oggettivo e indipendente non influenzato dalle iterazioni interne dell'LLM o dalla sua auto-percezione.

Il *distraction bias* descrive la tendenza dei modelli a venir influenzati da dettagli irrilevanti o di poca importanza durante le valutazioni; anche introdurre informazioni non pertinenti può fuorviare il modello e inficiare le valutazioni. Non ci sono vere e proprie strategie per arginare questo *bias*, anche se alcune soluzioni possono consistere nella scrematura dell'input in *pre-processing*, rimuovendo le informazioni superflue prima di presentarlo al modello, oppure redarre il prompt con istruzioni esplicite e chiare, così da focalizzare l'attenzione del modello solo sugli aspetti legati al task.

L'ultimo *bias* citato, il *fallacy-oversight bias* si riferisce al fenomeno in cui i modelli falliscono nell'identificare le fallacie o le inconsistenze logiche nelle risposte valutate, considerando valide le risposte che le contengono, quindi compromettendo la validità della valutazione.

Dopo aver elencato i vari tipi di *bias* che affliggono gli LLM-*as-judges*, pongo l'attenzione su recenti studi che hanno sottolineato quali sono le reali difficoltà alla base del modo in cui ragionano gli LLM.

In Loru et al. (2025), si afferma che gli LLM operano attraverso associazioni lessicali, priorità statistiche e indicazioni strutturali, piuttosto che mediante una vera e propria interpretazione contestuale e che questa approssimazione statistica produce asimmetrie sistematiche, come, ad esempio, un *pattern* che somiglia al pregiudizio dello scetticismo osservato negli esseri umani: un rifiuto eccessivo di informazioni accurate. Gli LLM replicano in parte le regolarità comportamentali identificate in psicologia, pur basandosi su meccanismi di valutazione fondamentalmente diversi. Ciò che emerge non è semplicemente un divario di accuratezza, ma un cambiamento strutturale nel modo in cui la valutazione stessa viene resa operativa quando il giudizio viene delegato a sistemi automatizzati. Mentre gli esseri umani interpretano l'accuratezza attraverso la comprensione del contenuto e il ragionamento pragmatico, gli LLM la ricavano dalle regolarità statistiche codificate durante

l'addestramento. Inoltre, gli LLM enfatizzano la "*ownership transparency*", allineandosi ai protocolli professionali di *fact-checking*, mentre gli esseri umani danno più importanza alle caratteristiche stilistiche e retoriche come il tono e la fluidità della scrittura. Queste preferenze rispecchiano euristiche note come la fluidità di elaborazione, in base alla quale la chiarezza e la neutralità emotiva migliorano la percezione della verità. Queste discrepanze sottolineano una differenza strutturale tra esseri umani e LLM: essendo questi sistemi sempre più integrati nei processi decisionali, diventa fondamentale valutare non solo se i loro risultati appaiono ragionevoli, ma anche come le loro procedure interne rendono operative categorie normative come l'affidabilità e i *bias*. Ciò, come dicono gli autori, è particolarmente urgente in un ecosistema informativo già contrassegnato da infodemie, dove l'eccesso di offerta di informazioni di bassa qualità o contraddittorie intacca la fiducia e amplifica la polarizzazione. Infine, l'apparente allineamento tra gli LLM e i giudizi degli esperti potrebbe mascherare solo una convergenza superficiale dei risultati. Delegare compiti di valutazione a questi sistemi rischia di incorporare *frameworks* guidati da associazioni lessicali e statistiche piuttosto che dal ragionamento deliberativo, amplificando i problemi informativi esistenti.

In Quattrociocchi, Capraro, Perc (2025) si afferma che gli LLM hanno delle caratteristiche che li pongono a notevole distanza dal modo in cui gli esseri umani ragionano e valutano, dato che non sono agenti epistemici, ma sistemi di completamento di *pattern* stocastici. Di conseguenza, vengono comparate le *pipeline* epistemiche umane e artificiali in 7 fasi e per ogni fase si identifica una "linea di rottura" epistemica, ossia un punto in cui i due pensieri divergono. Inoltre, viene introdotto il concetto di *Epistemia*, ossia la condizione in cui la plausibilità linguistica di un contenuto sostituisce strutturalmente la valutazione epistemica.

Gli LLM agiscono su regolarità statistiche estratte dai testi umani e imparano il funzionamento del linguaggio, ma non assumono conoscenze riguardo il mondo e le sue rappresentazioni; non individuano condizioni di verità

o strutture di causalità, ma *pattern* di co-occorrenze, associazione e continuazione in un testo (ivi: 2). In questo senso, incrementare il volume dei dati e il numero dei parametri migliora l'allineamento di superficie con l'output umano, senza, però, indurre la convergenza nei processi interni, motivo per cui strategie odierne di sviluppo, per compensare questo limite, aggiungono meccanismi supplementari al nucleo generativo. Su tutti, spiccano la RAG, l'uso di *tool* e i moduli di memoria esterna, che tentano di riconnettere i modelli a fonti esterne di informazione, come documenti, *database* o API. In questo modo, l'architettura produce risposte che sembrano sempre più affidabili, senza possedere l'apparato che rende l'affidabilità possibile. In questo modo la plausibilità sostituisce la verifica (ivi: 3).

La generazione di testo negli LLM può essere descritta come un processo stocastico che si evolve su uno spazio di stato discreto e ad alta dimensionalità. Ogni output è quindi la realizzazione di una traiettoria stocastica generata dal campionamento locale in questo spazio di stato. Ma procedure come la decodifica *greedy*, il *temperature scaling*, il *top-k* e il campionamento del nucleo non introducono invarianti, vincoli od obiettivi associati alla verità, al riferimento o alla validità, modificando semplicemente il modo in cui viene esplorata la massa di probabilità. Le distribuzioni linguistiche empiriche sono *heavy-tailed* (ossia hanno una probabilità maggiore di presentare valori anomali rispetto alle distribuzioni normali o esponenziali) e strutturalmente anisotrope: la massa di probabilità si concentra in un numero limitato di regioni corrispondenti alle costruzioni frequenti, agli schemi dominanti e ai modelli di co-occorrenza statisticamente rinforzati. In questo modo, le traiettorie sono dinamicamente sproporzionate verso bacini ad alta intensità, creando un'attrazione statistica spesso confusa con stabilità concettuale. In realtà, le regioni citate non sono attrattori semantici ma aggregati statistici e ciò che stabilizza è la distribuzione, non il significato. In questo contesto, il problema delle allucinazioni, che abbiamo già citato in 1.1.2, non è un'anomalia, ma un risultato atteso del campionamento da un modello statistico che non codifica riferimenti, condizioni

di verità o vincoli probatori. All'aumentare della scala, la qualità non cambia: i modelli più estesi perfezionano le stime di probabilità migliorando fluidità e coerenza interna, ma non viene aggiunto materiale epistemico, perciò aumenta solo la probabilità, non la validità dei risultati, e le conclusioni cui giunge il modello non sono il punto finale di una valutazione, ma il termine di una traiettoria stocastica (ibidem). Il lavoro di Quattrociocchi, Capraro, Perc (2025) si focalizza sul dividere il giudizio umano in 7 fasi sequenziali che possono essere imperfette e parziali, ma fanno parte di un ciclo epistemico in cui gli agenti esterni reagiscono, limitando l'errore. A queste 7 fasi ne vengono accostate altrettante che riguardano il giudizio dell'LLM, evidenziandone le differenze e i "punti di rottura" a partire dai quali i due ragionamenti divergono.

La prima fase vede contrapporsi le informazioni sensoriali e sociali umane da una parte e l'input testuale dall'altra. Il giudizio umano si genera con l'acquisizione di informazioni sensoriali e sociali in un ambiente multimodale; l'informazione acquisita attraverso i sensi e le emozioni è inserita in un contesto sociale pieno di segnali affettivi. Invece, gli LLM, basandosi su input testuali, non hanno contatto col mondo, ma operano su astrazioni che lo rappresentano, sequenze di simboli il cui significato è interamente derivato da schemi statistici imparati durante il *pre-training* e aggiustati successivamente con il *fine-tuning* supervisionato e basato sul *reinforcement*. Nonostante esistano recenti modelli multimodali, il loro "accesso percettivo" rimane derivativo: il sistema riceve *embedding* pre-addestrati anziché impegnarsi nell'esplorazione sensomotoria o nell'interazione corporea. La diretta conseguenza di questa assenza di una base percettiva è che l'LLM a volte produce ragionamenti impensabili per un umano in uno stadio iniziale di acquisizione di input (ivi: 4).

La seconda fase consiste nel confronto tra l'analisi percettiva e situazionale, operata dagli esseri umani dopo aver ricevuto le informazioni sensoriali e sociali, e la tokenizzazione e il *preprocessing* del testo operate dai modelli. L'attività umana organizza gli stimoli ricevuti per dare significato al vissuto, identificando oggetti e opportunità di azione e, attraverso processi

concettuali fondati, riconoscere agenti, intenzioni e potenziali minacce. Il corrispettivo degli LLM sono due attività meccaniche, in cui i dati linguistici grezzi vengono segmentati in token in accordo con un vocabolario predeterminato e, poi, standardizzati. A differenza dell'azione umana, in quella effettuata dall'LLM non vengono aggiunte strutture di significato, dato che produce una rappresentazione strutturalmente conveniente, ma semanticamente debole, essendo pensata per il calcolo numerico e non per essere interpretata. Per questo motivo, gli LLM possono compiere errori che un umano non compirebbe mai in questo stadio di analisi, dato che i modelli processano stringhe e possono interpretare in modo scorretto le unità in cui si divide una parola. Pertanto, in questa seconda fase, la linea di demarcazione epistemologica si amplifica: mentre la percezione umana ha già costruito un modello stratificato e significativo dell'ambiente, gli LLM, in una fase equivalente, hanno eseguito solo una suddivisione formale del testo. Un sistema analizza un mondo; l'altro segmenta una stringa (ivi: 5).

Nella terza fase si contrappongono memoria, intuizioni e concetti appresi dagli umani e il riconoscimento di *pattern* negli *embedding*. In questa fase, sia umani che LLM attingono alle conoscenze pregresse, ma lo fanno in modo diverso: gli esseri umani fanno affidamento sulla memoria episodica, sulla fisica e sulla psicologia intuitive e sui concetti appresi, il tutto codificato con dettagli spaziali e temporali, permettendogli di riconoscere analogie, di anticipare conseguenze sociali e di interpretare le situazioni nuove attraverso l'esperienza maturata nei momenti già vissuti. Invece, gli LLM, si basano sull'estrazione di *pattern* statistici in spazi di *embedding* ad alta dimensionalità: parole che co-occorrono, frasi che condividono la stessa struttura o concetti che appaiono in contesti simili. Una conseguenza diretta di queste differenze è che gli LLM possono considerare plausibili impossibilità fisiche ogni volta che tali scenari compaiono nei corpora linguistici, oppure possono anche non riuscire a riconoscere credenze, intenzioni e bugie in situazioni in cui anche i bambini riescono, perché mancano di una psicologia intuitiva per rappresentare stati

mentali distinti. In questa terza parte del processo si amplia ulteriormente la linea di demarcazione: gli esseri umani basano l'interpretazione sull'esperienza vissuta, su modelli intuitivi del mondo fisico e sociale e sulla comprensione concettuale, mentre gli LLM si basano esclusivamente su associazioni statistiche apprese dal linguaggio (ibidem).

Nella quarta fase si scontrano l'emozione, la motivazione e gli obiettivi umani con l'inferenza statistica tramite strati (*layer*) neurali degli LLM. Mentre gli esseri umani elaborano le percezioni e recuperano le conoscenze pregresse, i loro giudizi sono continuamente influenzati dalle emozioni, dalle motivazioni e dagli obiettivi, ossia le forze affettive e intenzionali che danno direzione alla cognizione. Per gli LLM lo stadio corrispettivo è l'inferenza statistica attraverso il calcolo neurale a strati; date le rappresentazioni di input e di *embedding* tokenizzati, il modello propaga le attivazioni attraverso la sua architettura di *transformer*, aggiornando le rappresentazioni vettoriali in base ai parametri appresi. Ogni strato esegue trasformazioni lineari, aggregazione ponderata in base all'attenzione e mappature non lineari che calcolano la distribuzione di probabilità sui token successivi. Questo processo è interamente meccanicistico e orientato all'ottimizzazione, dato che mira a minimizzare l'errore predittivo, non a perseguire obiettivi o a rispondere alla rilevanza emotiva. Il modello non si cura di nulla al di là delle attivazioni numeriche che codificano le associazioni statistiche. Anche metodi di addestramento come il RLHF, che introduce segnali di ricompensa per indirizzare il modello verso valori umani, non bastano per dare al modello obiettivi o motivazioni intrinseche; si limitano ad aggiustare le tendenze statistiche attraverso un'ulteriore ottimizzazione. A questo punto, la frattura epistemologica si fa più profonda: gli esseri umani formulano giudizi sotto l'influenza di obiettivi ed emozioni che conferiscono significato, priorità e direzione, mentre gli LLM eseguono trasformazioni statistiche indipendenti dal contesto prive di obiettivi intrinseci (ivi: 6).

Nella quinta fase, nel tentativo di produrre una risposta coerente dati gli input precedenti, gli esseri umani usano il ragionamento e l'integrazione di

informazioni, mentre gli LLM integrano il contesto testuale. I primi sfruttano il processo cognitivo per trarre inferenze, integrare prove, creare spiegazioni causali, considerare possibilità controfattuali e costruire piani a lungo termine, andando oltre la percezione immediata e dando vita a conclusioni basate non tanto su regole universali, quanto su obiettivi, valori e coscienza dei limiti facenti parte della conoscenza personale dell'individuo. I secondi, invece, non applicano nessun autentico ragionamento nel loro processo; data una sequenza di token, il modello li integra attraverso il meccanismo dell'attenzione che pesa le relazioni tra gli elementi nella finestra di input. Questo permette all'LLM di mantenere una coerenza tematica, di tenere traccia dei referenti e di adattare l'output al contenuto precedente, ma questa forma di "integrazione" rimane puramente sintattica e statistica: il modello non genera ipotesi causali, non giudica interpretazioni contrastanti e non costruisce un modello causale del mondo. A questo punto, la divergenza epistemologica diventa inconciliabile: la cognizione umana è orientata a obiettivi e organizzata attorno a modelli causali del mondo, mentre la cognizione LLM è priva di obiettivi intrinseci ed è guidata da modelli statistici vulnerabili a correlazioni spurie (ibidem).

Nella penultima fase si contrappongono da una parte le capacità umane di calibrazione metacognitiva e di monitoraggio degli errori, dall'altra l'eccessiva *confidence* e le allucinazioni dei modelli. Dopo aver iniziato a ragionare, gli essere umani compiono valutazioni metacognitive soppesando l'incertezza, eventuali errori, la *confidence* e, eventualmente, sospendendo il giudizio. Questi segnali servono all'individuo per capire quando deve controllare i fatti, recuperare più informazioni, rivedere ipotesi errate o sospendere il parere. Al contrario, gli LLM mancano totalmente di riflessione metacognitiva, non hanno possibilità di confrontarsi internamente sui conflitti e non possono monitorare l'affidabilità delle proprie rappresentazioni; non possono rendersi conto di non sapere qualcosa e, come detto anche in 1.1.2, sono restii ad ammetterlo. Anche quando un LLM produce una risposta come "Non sono sicuro", questa non è segnale di *confidence* interna, ma una costruzione linguistica. Da qui in poi la

divergenza epistemologica è definitiva: gli esseri umani possiedono un meccanismo di autoregolazione e di sensibilità all'incertezza che sovrintende alla formazione del giudizio, mentre gli LLM non ne possiedono nessuno (ibidem).

Nell'ultima fase si distinguono il giudizio sensibile al valore proprio del modo di esperire il mondo da parte degli esseri umani e il giudizio probabilistico proprio degli LLM. Gli individui, al termine della *pipeline* epistemica, formano giudizi sensibili al valore e dipendenti dal contesto, basandosi su valori personali, norme culturali, preoccupazioni reputazionali e obiettivi a lungo termine. Gli LLM, al contrario, producono giudizi probabilistici basati sul testo, determinati dalla struttura statistica dei loro dati di *training* e dall'immediato prompt testuale. Il "giudizio" di un LLM è semplicemente la distribuzione del token successivo, condizionata dal contesto. Non valuta la verità, il peso morale o le conseguenze pragmatiche: stima quali sequenze di parole sono più probabili dati gli schemi appresi dai corpora. In questa fase, il divario epistemologico ha conseguenze reali. Per gli esseri umani, un giudizio è un impegno orientato verso il mondo e infuso di valori, che integra modelli causali del mondo con emozioni, identità e scopi morali. Per gli LLM, un giudizio è semplicemente una previsione linguistica; anche quando i loro risultati sono superficialmente allineati, la procedura epistemica sottostante è fondamentalmente diversa (ivi: 7).

Riassumendo meglio queste sette incrinature tra il pensiero umano e quello dei modelli, vengono individuati dei "difetti" e sono rispettivamente i seguenti: il difetto di *grounding*, il difetto di *parsing*, il difetto dell'esperienza, il difetto di motivazione, il difetto di causalità, il difetto metacognitivo e il difetto di valore. Di conseguenza, viene introdotto il concetto di "*Epistemia*", ossia una condizione strutturale in cui la plausibilità linguistica sostituisce la valutazione epistemica. Descrive un sistema in cui vengono prodotte risposte sintatticamente ben formate, semanticamente fluenti e retoricamente convincenti, senza derivare i processi attraverso i quali le convinzioni vengono normalmente formate, testate e riviste. L'utente sperimenta il possesso di una risposta senza aver attraversato il

lavoro cognitivo del giudizio (ivi: 8). La riflessione più interessante che riguarda da vicino questo lavoro è la seguente:

> *The defining mark of Epistemia is the decoupling of content from evaluation. In human cognition, judgment is embedded in an epistemic loop: claims are checked against evidence, beliefs collide with counterexamples, and conclusions remain revisable in light of new information and social feedback. In generative systems, by contrast, there is no internal locus where claims can be tested, withdrawn, or defended. The model does not distinguish between "true" and "false" continuations; it distinguishes between more and less likely ones. What is generated is not what holds, but what fits.* (Quattrociocchi, Capraro, Perc 2025: 8)

Per questo motivo, l'Epistemia è il risultato di un disallineamento tra una competenza linguistica molto sofisticata e l'assenza di controllo epistemico. L'Epistemia non dipende dal tasso di errore, ma dal fatto che la valutazione stessa viene bypassata in modo strutturale; la trasformazione non è nel sapere in sé, ma in come il sapere viene prodotto, spostando l'attività epistemica da un processo a un prodotto.

Infine, vengono presentate tre implicazioni su come i sistemi generativi vengono valutati, gestiti e integrati nelle pratiche epistemiche. La prima riguarda la valutazione epistemica oltre l'allineamento superficiale, ossia il fatto che gli attuali paradigmi di valutazione di LLM si basano sull'allineamento superficiale: accordo con le risposte umane, sul successo nel task o sulla somiglianza comportamentale con prompt controllati. Dal punto di vista dell'Epistemia, queste analisi non sono sistematicamente sufficienti, dato che testano quando un output appare giusto, non se è prodotto attraverso processi che sostengono il giudizio in condizioni di incertezza, contestazione e costrizioni del mondo esterno.

La seconda implicazione riguarda la *governance* epistemica al di là dell'allineamento comportamentale, per il fatto che il discorso contemporaneo su LLM e IA generative si focalizza sulla correttezza e sull'accettabilità degli output invece di garantire i processi epistemici alla base di questi output. Il focus sopra citato è necessario, ma insufficiente alla luce di Epistemia, perché il problema principale non è che un sistema dica qualcosa di dannoso, ma che sia usato per sostituire o impedire il giudizio umano e istituzionale pur mancando delle capacità epistemiche che renderebbero legittima tale sostituzione. La *governance* dell'IA richiede quindi un passaggio dalla regolamentazione di cosa i sistemi dicono alla regolamentazione di come gli output generativi vengono introdotti nei flussi di lavoro epistemici e dove possono sostituire in modo ammissibile il giudizio umano.

La terza implicazione riguarda l'alfabetizzazione epistemica oltre il pensiero critico, dato che, per le condizioni di Epistemia, gli utenti sono sempre più esposti a output fluenti che simulano giudizi. Per questo, viene proposta l'alfabetizzazione epistemica come competenza distinta che deve venire insegnata e concettualizzata e che contiene in sé tre gruppi di competenze: il primo riguarda la consapevolezza della *pipeline*, ossia il comprende la differenza tra un sistema che recupera evidenze e uno che sintetizza del testo, oppure riconoscere quando una risposta è un completamento o una valutazione; il secondo sono garanzie procedurali per l'uso quotidiano, come routine per il controllo incrociato con fonti indipendenti, o norme esplicite su quando rinviare il giudizio; il terzo riguarda le competenze istituzionali, ossia progettare *workflows*, pratiche didattiche e standard professionali che impediscano l'esternalizzazione della responsabilità epistemica alle interfacce generative. L'alfabetizzazione epistemica non sostituisce il pensiero critico, ma lo completa e lo estende a domini in cui la sfida epistemica centrale non è più la valutazione di argomenti, ma la *governance* del giudizio in sistemi ibridi umano-IA (ivi: 11).

Capitolo 3 Task e metodi

In questo capitolo illustrerò l'analisi del lavoro svolto nell'ambito di un tirocinio curricolare frequentato presso ETET, azienda con sede a Genova che si occupa di creare e somministrare online test linguistici valutati automaticamente tramite l'utilizzo di intelligenze artificiali. Per cominciare, nel paragrafo 3.1 descriverò il lavoro svolto tra dicembre 2024 e febbraio 2025 consistente nella somministrazione a un gruppo di persone (scelto tra familiari, amici e conoscenze in ambito universitario di differenti età e percorsi di studi) di un test di lingua inglese diviso in cinque sezioni che riflettono le *skill* oggetto d'indagine per verificare la corretta conoscenza della lingua inglese da parte dell'utente, ossia *listening*, *reading, reading comprehension, speaking* e *writing*. Di seguito, nel paragrafo 3.1.1 illustrerò nel dettaglio la struttura del test, poi, nel paragrafo 3.1.2 mostrerò il lavoro che è stato fatto sui prompt usati per istruire il modello a valutare le risposte della sezione di *writing* e di quella di *speaking*.

3.1. Descrizione del lavoro svolto e del metodo impiegato

Il test di lingua inglese creato sulla web-app di ETET è stato creato da un *team* di lavoro composto da membri (tra studenti e professori) dell'Università degli Studi di Pavia e membri interni di ETET ed è stato somministrato al gruppo di utenti dopo aver raccolto una preadesione al test tramite un modulo redatto su Google Form in cui si chiedeva al partecipante di inserire il proprio indirizzo email per poter inviare successivamente una spiegazione più dettagliata dell'iter da seguire, che consisteva nella compilazione di un secondo modulo con informazioni riguardo il parlante e le sue abitudini comunicative, nello svolgimento vero e proprio del test e, infine, nella compilazione di un terzo modulo per condividere *feedback* maturati dopo la partecipazione al test.

Il test è costituito, come si è detto a inizio capitolo, da cinque parti che corrispondono alle quattro *skill* (*listening, reading, speaking e writing*) che

abbiamo ritenuto opportuno indagare per avere una conferma più chiara possibile del livello di conoscenza della lingua inglese da parte dei fruitori del test. In particolare, si è voluto rispettare la precisa sequenza di somministrazione presentata per avvicinare l'utente in modo graduale alle varie modalità di fruizione del test. Una prima metà, con le domande di *listening*, di *reading* e di *reading comprehension*, vuole permettere all'utente di avvicinarsi in modo più leggero ai compiti che verranno presentati; infatti, le domande di questa prima parte, richiedono tutte una minima interazione nella risposta, avendo delle risposte multiple "a crocetta" oppure dei *fill-in-the-blank* con menu a tendina. La seconda metà invece, presenta la parte più corposa del test e necessità da parte dell'utente uno sforzo cognitivo maggiore, dato che le *skill* di *speaking* e *writing* vengono analizzate con prove libere che richiedono, nel primo caso, di registrare tracce audio e, nel secondo caso, di scrivere liberamente dei testi.

In particolare, per quanto riguarda le caratteristiche tecniche del sistema utilizzato da ETET: la piattaforma usa algoritmi di *feedback* in tempo reale, LLM e tecnologie di ASR. Il sistema ASR è basato su AzureAI6, che sfrutta il modello Whisper di OpenAI7. Gli input vocali vengono acquisiti tramite browser in formato .ogg o .wav (canale mono) e non vengono normalizzati. Le domande aperte sono valutate tramite il GPT-4o (poi GPT-5.2) di OpenAI8, un modello accessibile via API, con prompt personalizzati per ciascun tipo di test e domanda. La valutazione è asincrona, avviene in *background* e viene restituita solo al termine del test (Vignoli, Combei, Zappulla 2025).

3.1.1. Struttura del test

La sezione di *listening* (ascolto) consiste di 15 domande di varia difficoltà: 5 di livello facile, 5 di livello intermedio e 5 di livello avanzato (la dicitura delle difficoltà è quella usata in fase di redazione del test). Nell'affrontare le domande di questa sezione, l'utente deve, seguendo delle brevissime indicazioni, ascoltare una traccia audio e, subito dopo, rispondere a una domanda scritta mediante la

selezione dell'unica risposta corretta tra quattro possibilità. La durata delle tracce audio varia dai 2 ai 70 secondi con la possibilità di essere riascoltate e la durata complessiva delle domande (considerando un tempo doppio di ascolto e un tempo per rispondere alla domanda) varia dai 15 ai 100 secondi. La competenza di *listening* richiede, nel modo in cui sono state impostate le domande, sia di leggere le istruzioni (formulate sempre in lingue inglese per tutto il test), sia di ascoltare la traccia audio, per poi di nuovo leggere la domanda posta, le scelte multiple e scegliere la risposta corretta.

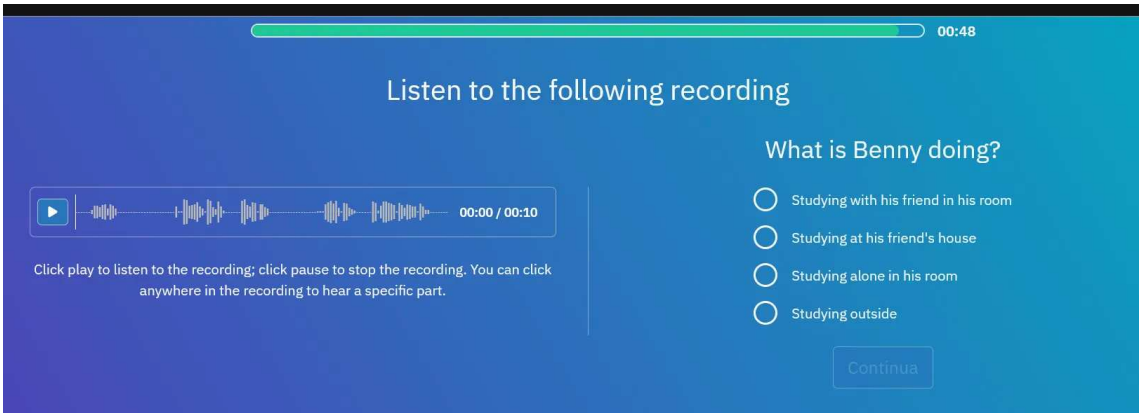


Figura 1: Schermata di ETET con domanda di listening *a "scelta singola" e allegato audio.*

La sezione di *reading* (comprensione del testo scritto), essendo composta solo da testi scritti e da sole domande di tipo *fill-in-the-blank* con scelte limitate, costituisce la parte più semplice e quella in cui l'utente deve apportare la minima interazione possibile (un semplice click col cursore). Questa sezione è formata da 18 domande di livello vario: 8 di livello facile, 4 di livello intermedio e 6 di livello avanzato. In particolare, all'utente viene presentata una frase con uno spazio vuoto e, cliccando su questo, si apre un menu a tendina con le scelte disponibili. Come si nota in fig. 2, la domanda è sempre corredata da un'indicazione per guidare l'utente alla corretta compilazione ("*Fill in the blank with the correct alternative from the drop-down menu.*" [trad. "Riempi lo spazio vuoto con l'alternativa corretta nel menu a tendina"] oppure "*Fill in the blank with the correct option*" [trad. "Riempi lo spazio vuoto con l'opzione corretta"]).

Inoltre, la finestra di tempo messa a disposizione dell'utente per compilare la domanda è posta, per tutte le domande della sezione, a 10 o 30 secondi, che iniziano a scorrere non appena l'utente clicca sul tasto per iniziare il test o per terminare la domanda precedente. Questa sezione, come si avrà modo di vedere anche a confronto con il capoverso successivo sulla *reading comprehension* (comprensione di un testo scritto esteso), non consiste in una tradizionale sezione di *reading*, quanto piuttosto in una sezione di *grammar* (grammatica), dato che le domande presenti si focalizzano quasi sempre su specifici *item* (oggetti) grammaticali e richiedono all'utente di sapere esattamente quelli per selezionare correttamente la risposta. Perciò, la vera e propria competenza di lettura non viene testata in modo del tutto soddisfacente, dato che vengono presentate domande molto brevi e con degli specifici target.

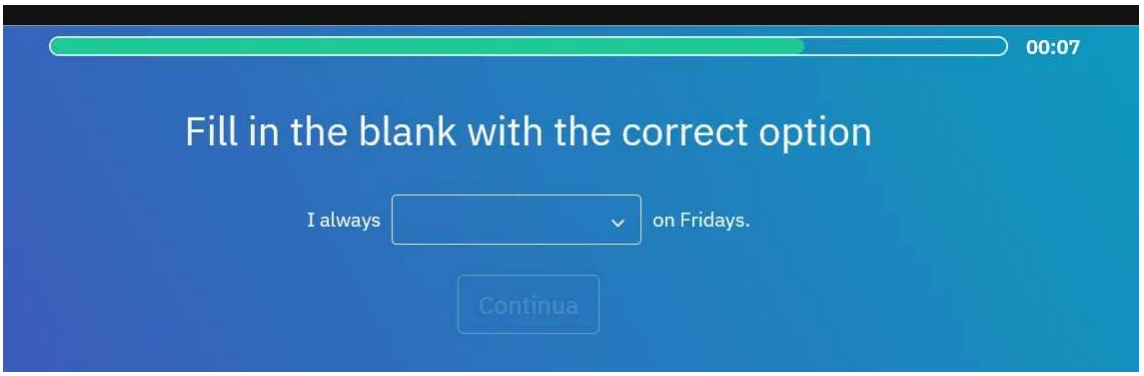


Figura 2: Schermata di ETET con domanda di reading *"riempi gli spazi vuoti a scelta".*

La parte di *reading comprehension* è costituita da soli 3 quesiti, di cui due divisi in due parti, e consistono di testi scritti (la cui dimensione varia dai 654 ai 2021 caratteri) seguiti da una domanda con risposta a scelta multipla. La difficoltà è intermedia per il primo quesito (diviso in due parti) e avanzata per gli ultimi due (di cui il secondo diviso in due parti). Dopo una semplice indicazione ("*Read the following text and choose the correct answer*" [trad. "Leggi il testo seguente e scegli la risposta corretta"]), viene mostrato il testo (nel primo caso vengono descritti i rischi che corrono le città di costiera a fronte

dell'innalzamento del livello dei mari e dell'aumento di precipitazioni; nel secondo caso un breve scritto illustra un tour fittizio nella città di Londra; nel terzo caso si mostra un adattamento di un brano tratto da *The Great Gatsby* di Francis Scott Fitzgerald) e, accanto al testo, si pone la domanda con le relative risposte. Le tempistiche assegnate variano da 50 a 290 secondi in base alla lunghezza del testo da leggere.

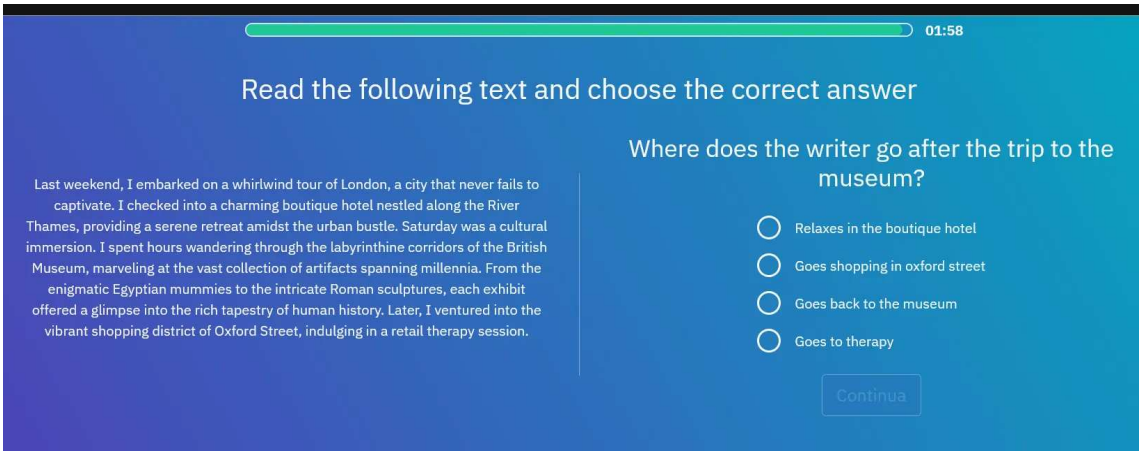


Figura 3: Schermata di ETET con domanda di reading *comprehension a "scelta singola".*

La sezione di *speaking* (parlato), con cui inizia la seconda metà del test dedicata alle produzioni libere da parte dell'utente, è composta da 10 quesiti di difficoltà variabile (5 di difficoltà facile, 3 di difficoltà intermedia e 2 di difficoltà avanzata) che presentano in 8 casi su 10 delle tracce audio da ascoltare e con cui bisogna interagire in modo diverso: in un caso bisogna riassumere il contenuto dell'audio, in quattro casi bisogna dimostrare di aver compreso il contenuto della traccia audio rispondendo a domande di comprensione, in un altro caso è necessario "rispondere" all'audio simulando una conversazione con elementi forniti nella descrizione scritta del quesito, in un altro caso bisogna riconoscere una forma scorretta usata nella traccia audio e nell'ultimo quesito bisogna dare un consiglio libero data una situazione presentata nell'audio relativo. Negli ultimi due casi, le domande non contengono file audio da ascoltare per rispondere alla domanda, ma consistono una in una descrizione vocale di una foto mostrata a

schermo, l'altra nell'esplicitazione di un'opinione personale dell'utente fornita dopo aver letto un testo scritto che tratta del cambiamento climatico. Le indicazioni che vogliono guidare l'utente, data la natura così diverse delle varie domande, cambia radicalmente da quesito a quesito (dalla più semplice "*Briefly summarize the conversation below.*", alla più complessa "*Jane visits your house for dinner. Answer by simulating a conversation and incorporating the phrases provided in brackets (study, his room). Your response should not exceed 60 seconds.*"), così come le tempistiche, che vanno da 35 a 176 secondi. Oltre a questi tempi generali, nella maggior parte delle indicazioni fornite all'utente si specificano dei tempi da non superare per quanto riguarda la sola produzione orale (in alcuni casi si chiede di non superare i 15 secondi, in altri i 60 secondi).

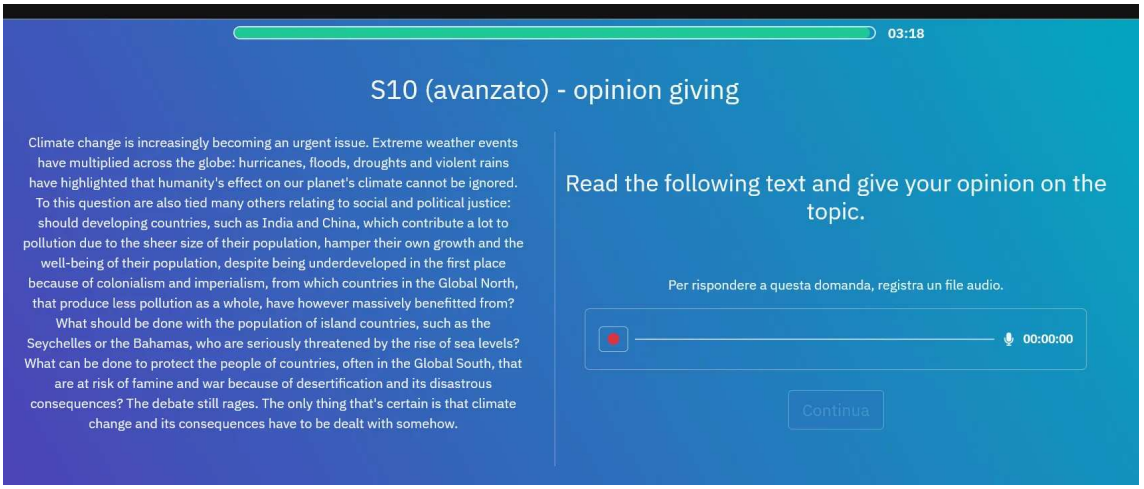


Figura 4: Schermata di ETET con domanda di speaking *a "composizione libera" con testo esteso.*

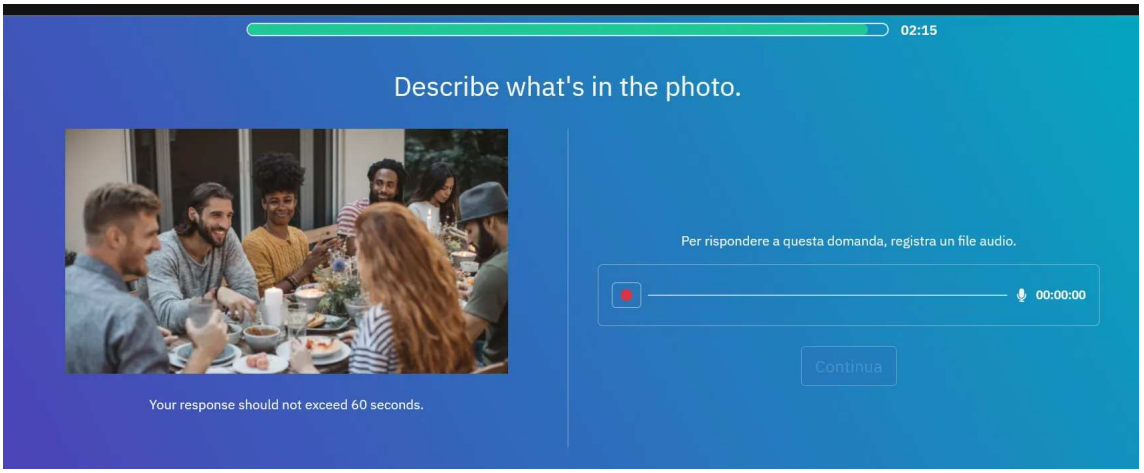


Figura 5: Schermata di ETET con domanda di speaking *a "composizione libera" con immagine.*

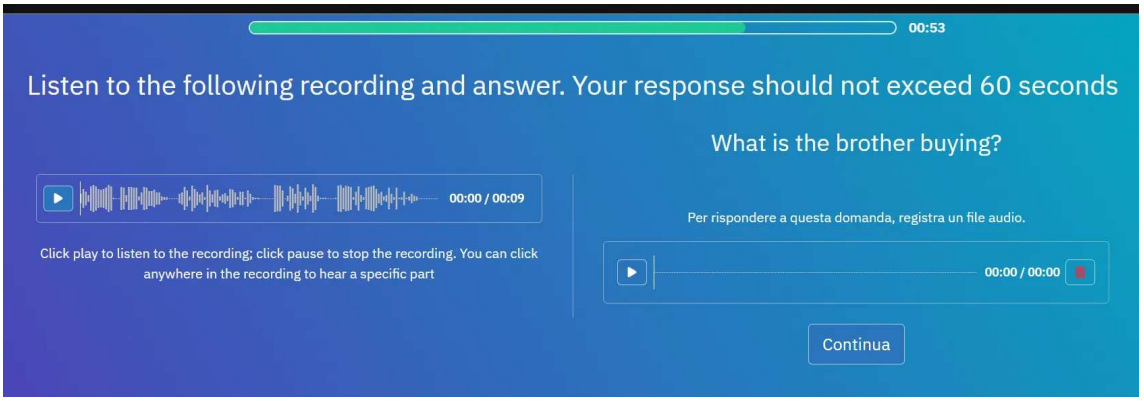


Figura 6: Schermata di ETET con domanda di speaking *a "composizione libera" con traccia audio.*

L'ultima sezione, quella di *writing* (scrittura), è composta da 14 domande di varia difficoltà (7 di livello facile, 4 di livello intermedio e 3 di livello avanzato) presentate in tre modalità diverse. La prima modalità si trova in 12 domande e consiste in brevi frasi con un *fill-in-the-blank* (in due casi gli spazi vuoti sono due, in un caso gli spazi vuoti sono tre, negli altri lo spazio mancante è uno solo) e lo spazio vuoto compilabile liberamente scrivendo con la tastiera. In alcuni casi la risposta attesa è costituita da più parole, per questo motivo nelle indicazioni di compilazione viene specificato il massimo di parole utilizzabili nello spazio vuoto (si va da un massimo di 1 parola a un massimo di 4 parole). La seconda modalità è presente in un solo quesito e consiste nel riordino di parti

disordinate di una frase, i cui elementi vengono esplicitati all'utente nell'ordine sbagliato. La terza modalità si trova nel quesito che rappresenta effettivamente la competenza di scrittura: l'utente deve scrivere liberamente una lettera motivazionale (lunga al massimo due paragrafi) da indirizzare a un'azienda che si occupa di tecnologia. Chiaramente, come già detto per la sezione di *reading*, anche in questa sezione la vera e propria attività di *writing* si ritrova solo nel quesito con la redazione della lettera motivazionale; in tutti gli altri casi l'attività di writing è ibridata a domande che richiedono la sola applicazione di *item* grammaticali, seppur calati in contesti più o meno estesi. Dal punto di vista delle tempistiche concesse all'utente per la compilazione, vanno da 15 a 210 secondi, in base al compito specifico da portare a termine.

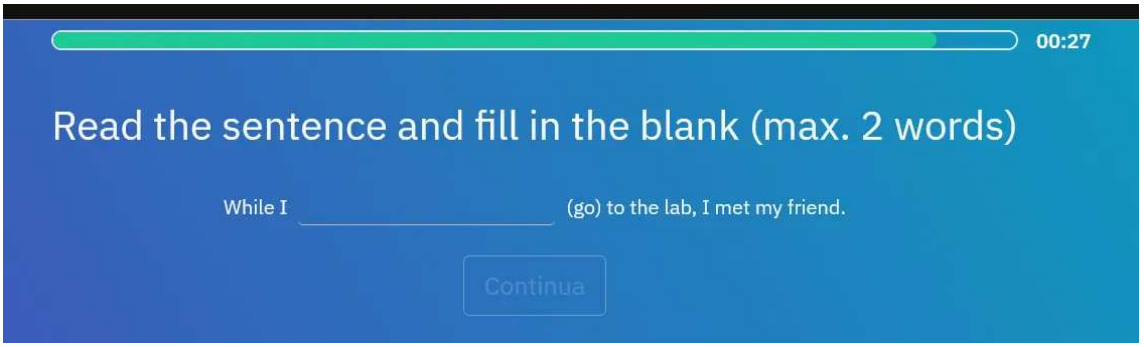


Figura 7: Schermata di ETET con domanda di writing *"riempi gli spazi vuoti liberamente".*

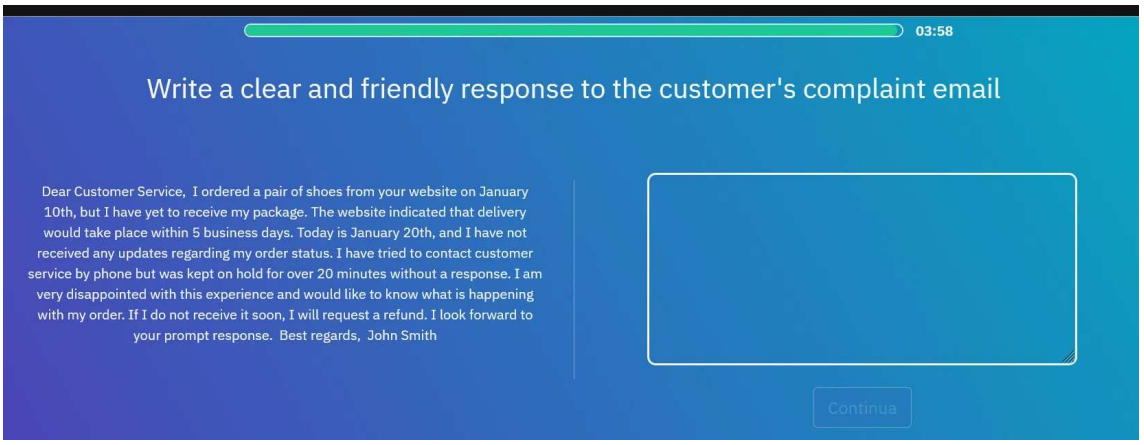


Figura 8: Schermata di ETET con domanda di writing a *"composizione libera".*

Dopo aver illustrato la composizione del test che è stato somministrato ai più di 44 partecipanti, mostro il metodo che è stato utilizzato sia per scegliere gli *item* grammaticali da indagare, sia per scegliere le tempistiche da utilizzare in ogni tipo di domanda. Innanzitutto, guardando alla totalità del test e ai tipi di domande che possono essere inseriti tramite il sistema, sono stati usati in tutto 4 tipi diversi di modalità: le domande a "scelta singola", ossia che presentano una serie di 4 risposte, mostrate come in fig. 3, di cui una sola è corretta; le domande denominate "riempi gli spazi vuoti a scelta", in cui le 4 scelte sono mostrate tramite il menu "a tendina", come in fig. 2; le domande denominate "riempi gli spazi vuoti liberamente", in cui si può scrivere manualmente nello spazio vuoto, come mostrato in fig. 7; infine, le domande a "composizione libera", in cui si può produrre liberamente sia un testo, sia una traccia audio, come si nota in fig. 6 e 8. A ognuna di queste modalità è possibile allegare una traccia audio, un'immagine o un testo più o meno breve; come si è già mostrato precedentemente in questo paragrafo, questo tipo di allegati è stato inserito solo in alcuni casi (ad es. le tracce audio sono state aggiunte solo nelle domande di *listening* di tipo "scelta singola", o in alcune domande di *speaking*, che hanno per tipo "composizione libera"). Per selezionare gli oggetti grammaticali da richiedere nelle singole domande, si è usata la teoria della processabilità (Pienemann 1998), che descrive il livello di difficoltà di un item grammaticale come correlato alla sua complessità sintattica e i livelli proposti dal CEFR (A1, A2, B1, B2, C1 e C2); i livelli di difficoltà presentati sopra (facile, intermedio e avanzato) sono modellati sui punteggi CEFR secondo il seguente schema: il livello facile corrisponde ai livelli CEFR A1 e A2, il livello intermedio corrisponde ai livelli B1 e B2, mentre il livello avanzato corrisponde ai livelli C1 e C2 (nel capoverso successivo se ne esplicitano meglio le corrispondenze). La quantità di domande per ogni livello di difficoltà è stata scelta in interazione con le tempistiche, impostate per raggiungere circa 30 minuti complessivi in modo tale da rendere il test comodamente fruibile online. Per questo motivo, basandosi su dati empirici, si sono determinate delle durate standard per ogni task affrontabile nelle varie

sezioni del test e, sommando le varie durate e raggiungendo i circa 30 minuti di esecuzione del test, si è distribuito il livello di difficoltà come segue: 15 domande totali di livello facile, 17 domande totali di livello intermedio e 18 domande totali di livello avanzato.

Dopo aver illustrato brevemente la teoria su cui ci si è basati per scegliere le tempistiche e gli *item* grammaticali, descrivo ora l'iter seguito dall'LLM per valutare le domande e per far convergere i punteggi in modo tale da inquadrare il risultato dell'utente in una classificazione precisa e coerente.

ETET	CEFR
86-100	C2 (Proficient)
71-85	C1 (Advanced)
51-70	B2 (Upper Intermediate)
41-50	B1 (Upper Intermediate)
31-40	A2 (Upper Intermediate)
0-30	A1 (Beginner)

Tabella 1: Corrispondenze tra i punteggi CEFR e i punteggi interni di ETET.

Innanzitutto, si può osservare in tab. 1 la scansione delle corrispondenze tra le voci CEFR che non hanno una quantificazione numerica e l'equivalenza fatta in ETET per rendere le fasce in un intervallo numero 0 - 100. Inoltre, prima di proseguire, è bene specificare un ulteriore dettaglio riguardante i punteggi per come sono riportati sul portale ETET: ogni quesito possiede un *tag* che serve al modello di valutazione per inquadrare le sezioni (i 4 *tag* disponibili sono *listening*, *reading*, *speaking* e *writing*) e un relativo peso, espresso in 3 punti, 6 punti oppure 9 punti. Questi tre punteggi sono affidati alle domande in base allo sforzo cognitivo che richiedono nella compilazione da parte dell'utente. Le

domande che hanno punteggio 3 richiedono all'utente il minor sforzo cognitivo e sono tutte quelle di tipo "riempi gli spazi vuoti a scelta" della sezione di *reading*, dato che presentano brevi testi da leggere e la selezione di una risposta tra quattro scelte già riportate a schermo. Le domande che hanno punteggio 6 richiedono uno sforzo cognitivo maggiore, date le consegne più complesse e le modalità di compilazione che richiedono uno sforzo pratico maggiore, e consistono in diversi tipi di quesiti: alcune domande di *speaking* che richiedono una semplice risposta di comprensione del testo o di una traccia audio; tutte le domande di *listening*, dato che consistono nell'ascolto di una traccia audio e nella compilazione di un domanda a scelta multipla; tutte le domande di *reading comprehension*, dato che richiedono l'elaborazione di testi più complessi; infine, le domande di writing del tipo "riempi gli spazi vuoti liberamente". Le domande che hanno punteggio 9 sono quelle più complesse e che richiedono da parte dell'utente il grado maggiore di sforzo cognitivo: hanno punteggio 9 le domande più articolate di *speaking*, ossia quelle che richiedono di esprimere verbalmente concetti elaborati e opinioni personali, e la domanda di *writing* che richiede di scrivere la lettera motivazionale, data la libertà nella produzione.

Di seguito, riporto l'iter preciso che seguono il modello e le sue applicazioni per ricavare i punteggi dalle domande di *speaking* e *writing*, che sono le due parti del test che ci interessano maggiormente ai fini dell'analisi del prompt:

1. Il modello valuta le singole risposte dell'utente attraverso 5 parametri di valutazione: *comprehension* (comprensione), ossia l'aderenza della risposta al contesto del compito che è stato assegnato all'utente; *content* (contenuto), ossia la correttezza a livello di contenuto della risposta; *grammar* (grammatica), ossia la correttezza grammaticale della risposta; *lexis* (lessico), ossia la varietà lessicale nella risposta dell'utente; infine *coherence* (coerenza), la coerenza interna testuale della risposta. Il punteggio viene conferito in centesimi e i 5 parametri elencati pesano nel

seguente modo: per le domande a composizione libera di *writing*, *grammar* ha un peso di 50%, *content* ha un peso di 25%, *lexis* ha un peso di 15%, *comprehension* ha un peso di 5% e *coherence* ha un peso di 5%; per le domande a composizione libera di *speaking*, *grammar* ha un peso di 50%, *content* ha un peso di 25%, *lexis* ha un peso di 10%, comprehension ha un peso di 5% e *coherence,* insieme al parametro di *pronunciation* (pronuncia) ottenuto tramite Azure, hanno un peso di 10%. Per le metriche che riguardano il puro contenuto testuale, la valutazione viene fatta da una parte sul testo scritto della domanda di *writing*, dall'altra su una trascrizione delle risposte di composizione libera della sezione di *speaking*, dato che il modello in uso non è di tipo multimodale;

2. Il risultato in centesimi prodotto nella fase precedente viene convertito nel punteggio impostato per la singola domanda di 3, 6 o 9 punti;
3. Il punteggio convertito di tutte le domande viene accorpato per avere il totale complessivo delle sezioni, per poi elaborare anche il totale globale del test e passare all'ultima fase. Ad esempio, nel test somministrato i punteggi di sezione erano 84 per *writing* e *reading*, 72 per lo *speaking* e 90 per il *listening*, per un totale di 330 punti.
4. Nell'ultima fase, il punteggio complessivo e i punteggi di sezione vengono convertiti in percentuale perché possano aderire alla tabella mostrata in tab. 1 identificando un preciso livello della scala CEFR restituendone un valore sia totale, sia particolareggiato per ogni *skill*.

3.1.2. Struttura del prompt

Il prompt elaborato per valutare le risposte alle domande di composizione libera delle sezioni di *writing* e *speaking* è costituito da diverse parti e assolve al task di valutazione o, in questo caso specifico, *Automated Essay Scoring* (AES) (Pack, Barrett, Escalante 2024). Di seguito se ne fa un'analisi punto per punto, riportando il testo originale senza correzioni o modifiche di sorta.

> (Prompt v. 1.0) Sei un madrelingua inglese che ha esperienza nella valutazione delle competenze linguistiche di candidati che parlano lingue diverse dall'inglese e che devono sostenere un esame che certifichi il loro livello nella Classificazione CEFR del Quadro Comune Europeo di Riferimento per le lingue. Hai più di vent'anni di esperienza e quindi sei considerato uno dei migliori esperti in questo campo nel panorama europeo.
> Il tuo scopo è valutare una risposta a testo libero.

Questa prima parte del prompt, secondo le descrizioni che ho riportato in 1.1, contiene, per la tassonomia di Eager & Brunton (2023: 4), il verbo che imposta il task che il modello deve eseguire ("valutare") e un focus parziale ("una risposta a testo libero"). L'altra parte del focus e una parte del contesto si accostano all'utilizzo iniziale della tecnica di *role prompting* di cui si è parlato nel paragrafo 1.1.2 e che talvolta, come in questo caso, viene usata in esordio (Lemeš 2024: 166). Questa tecnica viene usata in modo piuttosto esteso e, all'interno di essa, vengono già inserite informazioni preziose sia riguardo il contesto, sia riguardo il task: si specifica che la lingua oggetto di valutazione è l'inglese e che verrà impiegato dai parlanti come lingua seconda. Inoltre, si fa riferimento a uno dei contesti teorici su cui ci si basa per la valutazione, ossia la classificazione CEFR. Viene specificata la funzione di valutatore del modello e qual è il suo obiettivo specifico, ossia le risposte a testo libero. Per questo motivo, nella parte successiva vengono descritti in modo particolareggiato i 5 parametri di valutazione su cui il modello dovrà basarsi per valutare il livello delle risposte del parlante.

> Devi valutare la risposta in base ai seguenti parametri fornendo una valutazione in una scala da 0 a 100 ed una spiegazione dettagliata per ciascuno:
> COMPREHENSION: aderenza della risposta al contesto del compito che è stato assegnato al candidato: devi valutare quanto il candidato è stato

> coerente con il contesto che gli è stato fornito nel compito che gli è stato assegnato, a livello linguistico, situazionale e di relazione fra i partecipanti all'evento comunicativo rappresentato; in casi in cui sia richiesto di fornire la propria opinione, non bisogna penalizzare il candidato per aver fornito opinioni non abbastanza moderate, o per non aver considerato le opinioni opposte: quello che conta è l'aver compreso e rielaborato il contenuto della consegna. Anche se il candidato è stato duro e/o severo nei confronti dell'opinione avversa, ciò non costituisce una ragione per penalizzare il suo punteggio. Il candidato non ha nessun obbligo di essere imparziale, e va giudicato esclusivamente sulla base di quello che ha compreso;

In questa seconda parte del prompt si presenta, attraverso delle indicazioni iniziali, la scala di punteggio che il modello deve usare, ossia da 0 a 100, e che deve fornire una spiegazione dettagliata della sua valutazione. In particolare, si fornisce una definizione del primo criterio di valutazione, ossia *comprehension*. Insieme alla definizione, si aggiungono informazioni utili al fine di non permettere che il modello vizi il punteggio fuorviando il reale obiettivo di valutazione: essendo che i modelli come quello in uso tendono a valutare eticamente il contenuto degli elaborati in favore di un equilibrio morale e svalutano qualsiasi tipo di violenza, anche quella che si dimostra verso opinioni, nel prompt si sono presi provvedimenti per aggirare questo *bias* di ipercorrettismo, che può essere identificato come un *bandwagon-effect bias*, citato nel capitolo 2.

> CONTENT: correttezza a livello di contenuto: devi valutare quanto il contenuto risulta equilibrato nelle sue varie componenti ed esplicativo dell'argomento in forma coerente e comprensibile; ciò non implica imparzialità del candidato quando si tratta di dare opinioni, piuttosto quanto sia, corretto e pertinente alla consegna

> GRAMMAR: correttezza a livello grammaticale: devi valutare, fornendo una spiegazione, quanto il testo è corretto dal punto di vista della grammatica e della sintassi della lingua inglese;

Nella sezione del prompt sopra rappresentata si passa alla definizione di altri due criteri di valutazione che si occupano di indirizzare l'analisi del valutatore verso l'effettivo contenuto della risposta e verso la conoscenza dell'uso della grammatica. Nella descrizione del parametro *content*, si fa riferimento anche alla coerenza testuale e si torna a specificare che il modello non deve tenere conto delle implicazioni etiche delle affermazioni dell'utente, ma deve solo attenersi alla valutazione dell'attinenza del contenuto rispetto al compito assegnato, come già detto nel capoverso precedente sulla voce *comprehension* e sul rischio che si manifesti il *bandwagon-effect bias*. Per quanto riguarda il parametro *grammar*, si attiene alla mera descrizione dell'oggetto di valutazione, specificando di fare attenzione anche alla sintassi.

> LEXIS: varietà lessicale: devi valutare, fornendo una spiegazione, quanto il lessico impiegato sia appropriato e variegato. Esso deve dunque essere pertinente all'argomento trattato, ma la valutazione sarà positivamente influenzata nel caso vengano utilizzati termini più vari e/o più inusuali. L'utilizzo di parole o espressioni di natura più colloquiale non detrae dal punteggio, eccetto che nei casi in cui il compito richieda di creare un testo di natura esplicitamente formale, oppure nel caso esso evochi situazioni inerentemente formali. Allo stesso modo, l'utilizzo di varianti e sinonimi regionali e/o dialettali di parole inglesi non detrae dal punteggio, a meno che non siano in contrasto con il tono e/o il contesto della consegna. Il candidato non deve essere eccessivamente penalizzato per mancanze di precisione. Per quanto essere precisi sia sicuramente meglio, essendo questi testi creati sul momento è normale che si faccia qualche imprecisione; finché l'imprecisione non impatta la consegna specifica della domanda, essa non deve essere considerata un errore grave;

La voce che riguarda la definizione del parametro *lexis* chiarisce al modello come deve valutare il materiale lessicale. Un'importante specifica che viene fatta riguarda il contesto cui il lessico deve adeguarsi: il modello deve saper valutare quando l'utente usa una terminologia non adatta alla situazione comunicativa richiesta nel compito assegnato dalle singole domande. Un'altra specifica che viene fatta riguarda l'adesione o meno alla lingua standard nell'utilizzo di varianti regionali o dialettali, ma questo dettaglio si collega in modo stretto con l'intenzione pedagogica del test e lo status dell'inglese britannico come standard nell'insegnamento. Il riferimento che si fa al termine di questa parte sulla mancanza di precisione si riferisce in modo specifico alla trascrizione delle risposte audio, che possono venire interpretate erroneamente dal modello come risposte pensate per essere scritte. Chiaramente, le due modalità hanno caratteristiche che le differenziano in modo molto riconoscibile, ma le sfumature non vengono sempre intercettate dal modello e si è ritenuto d'aiuto specificare un'attenzione particolare a questi aspetti. Per esempio, nelle risposte orali è naturale la presenza di qualche ripetizione, cosa che, invece, nelle risposte scritte non è concepibile.

> COHERENCE: coerenza testuale: devi valutare, fornendo una spiegazione, la coerenza interna del testo fornito, in particolare, valutando l'utilizzo di connettori, strutture sintattiche e/o strategie di organizzazione del discorso appropriati per la comprensione il più possibile fluente del testo. In caso si tratti di una risposta di tipo 'composizione scritta' devi essere più severo, mentre in caso di tipo 'composizione orale' puoi tollerare qualche ripetizione o riformulazione.

In questa parte si fornisce una descrizione più articolata di cosa si intende con *coherence* presentando anche degli esempi di elementi che rendono un testo internamente coerente. In coda, come nel parametro precedente, si specifica la differenza di cui il modello deve tenere conto tra una risposta scritta e una risposta trascritta da una traccia audio: anche in questo caso, la differenza è di

vitale importanza, dato che eventuali ripetizioni o pause dell'orale inficiano molto la coerenza, a maggior ragione se vengono considerate risposte pensate per essere scritte.

> La valutazione dev'essere effettuata in base alla lingua che ti viene fornita: se l'esaminando ha risposto in una lingua diversa, tutti i punteggi delle valutazioni devono essere a 0. I punteggi dei vari parametri devono essere indipendenti gli uni dagli altri: la valutazione di uno non può influenzare la valutazione dell'altro, e una stessa argomentazione non può essere addotta a supporto della valutazione da due parametri diversi.
> Il tuo lavoro è molto prezioso.
> Grazie per il tuo impegno.

Dopo aver descritto i 5 criteri di valutazione, vengono aggiunte due osservazioni generali: la prima descrive la reazione da adottare in caso un utente utilizzi una lingua che non è quella esaminata nel test; la seconda pone una linea guida sull'interazione tra parametri diversi, cercando di restringere il più possibile le probabilità di interferenza. Infine, in quest'ultima parte che chiude il prompt, vengono aggiunte due frasi finali che si rifanno al *role prompting* iniziale e tendono a richiamare il contesto in cui il modello deve operare, tentando di rafforzare una coerenza contestuale.

Di seguito, si riporta la struttura del prompt ripartita e schematizzata in base alla tassonomia di Schulhoff (2025: 5), fornita nel paragrafo 1.1, ossia la classificazione che secondo il mio parere si adatta meglio al prompt mostrato.

Parte 1	Ruolo
	Contesto
	Ruolo
	Direttiva
Parte 2	Contesto
Parte 3	Contesto
Parte 4	Contesto
Parte 5	Contesto
Parte 6	Indicazione di stile
	Formato output
	Direttiva
	Ruolo

Tabella 2: Classificazione del prompt secondo Schulhoff (2025: 5).

Capitolo 4 Risultati

Dopo aver presentato la struttura del test e quella del prompt, e dopo aver messo a fuoco quali sono le caratteristiche più importanti di quest'ultimo, passerò ora all'analisi dell'efficacia del prompt e di quelle che possono essere le sue criticità. Innanzitutto, specifico che, prima di valutare le prestazioni del prompt, è necessario definire un *benchmark* preciso dei risultati attesi dalla valutazione dei test. Infatti, si è ritenuto opportuno preparare una base di punteggio con l'annotazione effettuata da annotatori umani su un campione di 10 partecipanti all'esperimento, scelti con criteri di rappresentabilità sul totale di 44 partecipanti, in modo tale da avere una linea guida per valutare l'efficacia dei giudizi del modello.

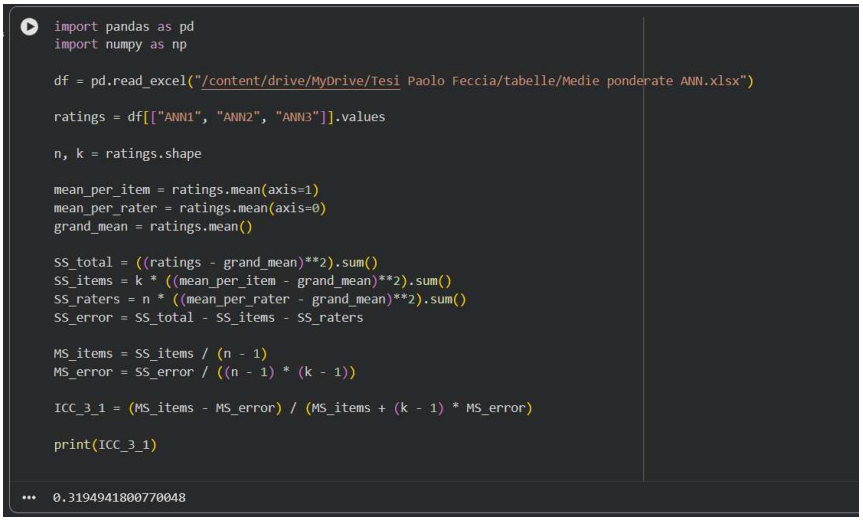
4.1. Valutazione del test tramite annotatori

Una volta scelto il campione rappresentativo del gruppo di partecipanti per facilitare la valutazione, tenendo conto di tutti gli aspetti anagrafici degni di nota, come l'età, il genere, l'occupazione, il titolo di studio e il livello di competenza autopercepito per come sono stati comunicati nel secondo modulo fornito ai partecipanti, le risposte anonimizzate del campione sono state condivise con un gruppo di tre annotatori esperti che hanno valutato le 10 risposte alle domande di *speaking* e la singola risposta alla domanda di *writing*.

Codice identificativo	Età	Genere	Occupazione	Titolo di studio	Livello di competenza autodichiarato
AFST-01	18-24	D	Studentessa	Laurea Triennale	B1
ZUSS-01	25-34	U	Studente	Scuola Superiore	B1
AUST-01	18-24	U	Studente	Laurea Triennale	B2
ZFSM-01	25-34	D	Studentessa	Laurea Magistrale	B2
BUIU-01	55-64	U	Imprenditore	Master Universitario	C1
AUST-02	18-24	U	Studente	Laurea triennale	C1
AFSS-01	18-24	D	Studentessa	Scuola Superiore	C1
ZFLM-01	25-34	D	Lavoratrice	Laurea Magistrale	C1
BULM-01	55-64	U	Lavoratore	Laurea Magistrale	A2
ZFLM-02	25-34	D	Lavoratrice	Laurea Magistrale	C2

Tabella 3: Campione scelto di 10 partecipanti su 44 totali.

Ai tre annotatori sono state fornite le linee guida per la valutazione che consistono negli stessi criteri già citati (*comprehension, content, grammar, lexis e coherence*), con in più, per le domande orali, tre parametri che consistono in una scorporazione del parametro *pronunciation* (pronuncia) già citato, ossia le seguenti tre metriche: fluenza, accuratezza e pronuncia.[4] I tre annotatori hanno fornito un punteggio da 0 a 100 in ogni criterio per ognuna delle 10 domande di *speaking* e dell'unica domanda di *writing*. Sono state determinate le medie ponderate dei punteggi (basando i pesi su quelli impostati dal modello specificati in 3.1.1 e generando un'ulteriore media nei riguardi dei parametri di *coherence*, fluenza, accuratezza e pronuncia che hanno tutti un unico peso, ossia il 10%, per le domande di *speaking*) per confrontare meglio i risultati e per calcolare l'*inter annotator agreement* (accordo tra annotatori) si è usato l'*Intraclass Correlation Coefficient* (coefficiente di correlazione intraclasse; ICC) data la presenza di tre annotatori, di un identico set di dati valutato da tutti e tre e di punteggi continui ordinali (0-100). Nei calcoli sotto riportati sono stati eliminati i punteggi nulli generati o da malfunzionamenti nel test o da risposte non date da parte degli utenti.



```
import pandas as pd
import numpy as np

df = pd.read_excel("/content/drive/MyDrive/Tesi Paolo Feccia/tabelle/Medie ponderate ANN.xlsx")

ratings = df[["ANN1", "ANN2", "ANN3"]].values

n, k = ratings.shape

mean_per_item = ratings.mean(axis=1)
mean_per_rater = ratings.mean(axis=0)
grand_mean = ratings.mean()

SS_total = ((ratings - grand_mean)**2).sum()
SS_items = k * ((mean_per_item - grand_mean)**2).sum()
SS_raters = n * ((mean_per_rater - grand_mean)**2).sum()
SS_error = SS_total - SS_items - SS_raters

MS_items = SS_items / (n - 1)
MS_error = SS_error / ((n - 1) * (k - 1))

ICC_3_1 = (MS_items - MS_error) / (MS_items + (k - 1) * MS_error)

print(ICC_3_1)
```

```
0.3194941800770048
```

Figura 9: Calcolo dell'ICC(3,1) effettuato con Python su Google Colab.

[4] La metrica di fluenza analizza la scorrevolezza del parlato, prendendo in considerazione l'eventuale presenza di riformulazioni o ripetizioni; la metrica di accuratezza analizza la produzione delle parole, con attenzione al rispetto delle distinzioni fonologiche della lingua inglese; la metrica di pronuncia analizza la precisione nell'uso della prosodia e dei profili intonativi.

Il calcolo effettuato con uno script in Python su Google Colab ha dato un risultato di 0.31, che, dato il range di punteggio da 0 (scarso accordo) a 1 (massimo accordo), dimostra uno basso accordo tra i tre annotatori.

Calcolando anche l'alfa di Krippendorff[5] tra i tre annotatori si ottiene un risultato di 0,15, che conferma lo scarso accordo tra i tre annotatori.

Ho effettuato anche il calcolo del coefficiente di Pearson[6] in tutte e tre le combinazioni tra i tre annotatori e tra ANN1 e ANN2 il coefficiente è di 0,23, tra ANN2 e ANN3 è di 0,26 e tra ANN1 e ANN3 è di 0,53, con una media totale di 0,34 punti.

Nonostante la bassa affidabilità nell'accordo tra gli annotatori, si useranno comunque le medie dei loro punteggi come *gold standard* per tentare un confronto coi risultati dell'LLM.

Di seguito, fornisco un riassunto dei risultati dati dalle metriche impiegate:

- ICC(3,1): 0,31;
- Coefficiente di Pearson (medio): 0,34;
- Alfa di Krippendorff: 0,15.

4.2. Valutazione del test da parte del modello e confronto con le annotazioni

Dopo la precedente analisi, vediamo ora quali sono stati i punteggi risultanti dal lavoro di valutazione effettuato dal modello sulle risposte degli utenti campione. Qui, il modello ha fornito le sue valutazioni basandosi sui criteri che ho illustrato nel paragrafo 3.1.1 e, una volta ottenuti i punteggi, ho calcolato la media ponderata per avere un punteggio complessivo in centesimi per ogni domanda, così da confrontare più agevolmente i dati con i risultati dei tre annotatori. Successivamente, ho anche calcolato la media dei tre risultati degli annotatori per

[5] Krippendorff 2013: 221.
[6] Sedgwick 2012.

confrontarla con la media ottenuta dai risultati dell'LLM. Ove possibile, ho effettuato i calcoli direttamente tra il punteggio totale dell'LLM e i punteggi dei singoli annotatori. I dati presentati in questo paragrafo sono stati ottenuti con la prima versione del prompt (v. 1.0) e risalgono al periodo in cui è stata effettuata la prima analisi completa all'inizio dell'anno 2025.

Per prima cosa, si è proceduto calcolando il *Mean Absolute Error* (MAE) che evidenzia l'errore medio assoluto mostrando le differenze tra i risultati attesi (l'annotazione umana) e quelli previsti dall'LLM. Il risultato è stato diviso per il numero di eventi osservati e il risultato di 17 punti ci testimonia che la concordanza tra le prestazioni contiene delle oscillazioni degne di nota.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

Figura 10: Formula usata per calcolare il MAE.

Effettuando anche il calcolo dell'indice di correlazione di Pearson tra i dati degli annotatori e i valori dell'LLM, risulta, dalla media delle sei combinazioni, una correlazione di 0,31 punti, quindi le variabili risultano non ben allineate e seguono una tendenza non del tutto omogenea. Calcolando successivamente l'alfa di Krippendorff si ottiene un valore di 0,15 punti. Inoltre, calcolando l'ICC tra gli annotatori e l'LLM si ottiene un punteggio di 0,29 mantenendo i punteggi dei tre annotatori scorporati; invece, facendone una media, il punteggio scende a 0,21 punti. Riassumo di seguito i valori sopra citati:

- MAE: 17;
- Coefficiente di Pearson: 0,31;
- Alfa di Krippendorff: 0,15;
- ICC(3,1): 0,29.

Dopo le precedenti analisi quantitative, voglio porre l'attenzione sull'analisi qualitativa di alcuni casi estremi che dimostrano le aderenze o le incongruenze tra i punteggi assegnati alle risposte. I casi in cui LLM e annotatori concordano perfettamente sono nella valutazione della domanda S4 di ZUSS-01,

della S6 di AUST-01, della S5 di AFSS-01 e della S2 di ZFLM-02. Mostro come esempio le valutazioni della domanda S4 di ZUSS-01, di cui riporto anche il contenuto sia della traccia audio con la consegna (2), sia della risposta dell'utente (3). I punteggi degli annotatori vengono mostrati scorporati, ma sono stati confrontati con quelli dell'LLM dopo averne fatto una media sui totali ponderati.

ZUSS-01	S4	Comprehension	Content	Grammar	Lexis	Coherence
	LLM	70	65	90	80	85
	ANN1	90	100	90	90	90
	ANN2	70	70	100	100	92
	ANN3	50	60	60	60	63

Tabella 4: Valutazioni della domanda S4 di ZUSS-01.

(2) Contenuto trascritto della traccia audio: "*Hi, John. I haven't seen you in a while. What are you doing here? / Madeline, hi. I'm just buying a pair of jeans for one of my kids. / Wow, you have kids. I didn't know. How many of them? / Oh, just two, but they're very naughty. They make enough trouble for twenty*."
Contenuto della consegna: *What is John buying? And for whom?*

(3) Trascrizione della risposta di ZUSS-01: *He is buying trousers for their kids, their two kids*.

Le medie ponderate degli annotatori e dell'LLM danno come risultato 81 punti, corrispondendo perfettamente. Possiamo notare che l'LLM ha premiato particolarmente la *grammar*, che è il criterio di valutazione con peso più alto, mentre ha giustamente abbassato i voti riguardanti *content* e *comprehension* a causa di alcune imprecisioni nella risposta; i due criteri sono comunque correlati, dato che il secondo criterio si evince dal contenuto della risposta. Per quanto riguarda la forma, l'LLM attribuisce un voto leggermente più alto dei due precedenti a *lexis* e *coherence*, data la brevità della risposta; basandomi sui giudizi espressi dall'LLM, si può comprendere come abbia diminuito il punteggio di *lexis* a causa dell'uso di *trousers* invece di jeans, mentre ha

penalizzato la *coherence* a causa della ripetizione che l'utente ha aggiunto come specifica alla fine dell'audio. Il ragionamento più simile è stato fatto dall'annotatore ANN2, mentre ANN1 ha premiato il *content* a discapito di tutti gli altri parametri, mentre ANN3 ha premiato la *coherence*, ma penalizzato la *comprehension*.

Per quanto riguarda i casi negativi, ossia di un disaccordo significativo tra annotatori e modello, porto come esempio le valutazioni della domanda S2 di BUIU-01, che dimostra una discrepanza di ben 60 punti tra la valutazione degli annotatori e dell'LLM. Anche qui vengono mostrati i punteggi scorporati dei tre annotatori, ma sono stati confrontati col modello dopo aver fatto una media sui totali ponderati. Inoltre, riporto anche il contenuto della domanda con la consegna (4) e la risposta dell'utente (5).

BUIU-01	S2	Comprehension	Content	Grammar	Lexis	Coherence
	LLM	40	30	20	20	25
	ANN1	90	90	90	90	95
	ANN2	100	100	100	100	100
	ANN3	60	60	60	65	62

Tabella 5: Valutazioni della domanda S2 di BUIU-01.

(4) Contenuto trascritto della traccia audio: "*My brother and I are throwing a party tonight. He is going to the supermarket to buy a few things. We need two packs of crisps, three large cokes and a bunch of plastic cutlery.*"
Contenuto della consegna: *What is the brother buying?*

(5) Trascrizione della risposta di BUIU-01: *Is binds crisps, not muncho, cuttery and coke.*

Qui, il modello affida un punteggio totale medio di 24 punti, mentre il punteggio medio dei tre annotatori è di 84 punti. Come si può notare, è chiaro il motivo del punteggio così basso assegnato dal modello: la trascrizione, a causa di un audio di bassa qualità, non ha funzionato a dovere e ha confuso alcune parole senza

capire su quale contesto affidarsi. La trascrizione corretta doveva essere: "*He's buying crisps, not much cutlery and coke*". Sorvolando sul problema tecnico appena descritto, l'LLM risponde comunque a dovere, ritenendo la risposta gravemente insufficiente in tutti gli aspetti. La *comprehension* ha un punteggio superiore agli altri parametri perché, a fronte di una risposta così breve, la presenza di parole riconducibili correttamente al contesto di partenza (*crisps* e *coke*) basta al modello per asserire che l'utente si sta riferendo parzialmente al contesto corretto. È chiaro che la situazione appena presentata costituisce un caso limite, ma ho voluto includerla per evidenziare una possibile debolezza del modello.

4.3. Valutazione del prompt

Dopo aver analizzato e confrontato le valutazioni fornite dagli annotatori e dal modello per mettere in luce pregi e difetti di quest'ultimo e il modo in cui opera, voglio concentrarmi sull'oggetto principale della presente ricerca, ossia il prompt utilizzato per valutare gli aspetti fino a ora descritti. Per fornire un'analisi più completa e affidabile possibile del prompt, oltre al tentativo di utilizzo dei metodi di valutazione descritti in 1.1.4 e nel capitolo 2, si valuterà il loro operato attraverso confronti con il *gold standard* degli annotatori, insieme a una valutazione iniziale della *consistency* del prompt verificata attraverso 10 iterazioni su due domande: una di *writing* e una di *speaking*, denominate in fase di sperimentazione rispettivamente W11 e S10 (poi, internamente a ETET EN-W-I-4 e EN-S-A-2, scelte perché richiedono dall'utente una risposta più estesa rispetto alle altre domande. Ho scelto per quest'analisi le risposte degli utenti BUIU-01, AFSS-01, BULM-01 e ZFLM-02 perché risultano le più complete e variano in modo interessante per livello ed estensione del contenuto.

Per quanto riguarda la valutazione della *consistency* del prompt, è stata calcolata la deviazione standard in tutte le 10 iterazioni per ogni criterio di punteggio e poi ne è stata fatta una media, come si può osservare in tab. 6.

		Media punteggio	Dev. standard
BUIU-01	S10	58	5
BUIU-01	W11	72	5
BULM-01	S10	42	11
BULM-01	W11	58	6
AFSS-01	S10	83	5
AFSS-01	W11	75	5
ZFLM-02	S10	77	5
ZFLM-02	W11	80	5

Tabella 6: Calcolo della consistency *del prompt tramite deviazione standard con GPT-4o.*

Come si nota, l'oscillazione del punteggio medio varia da un minimo di 5 fino a un massimo di 11 punti. Una distribuzione simile di punteggi indica che è presente una variazione media di 6 punti.

Volendo fare un ulteriore confronto tra i risultati ottenuti tramite il test di *consistency* presentato sopra e il *gold standard* degli annotatori che abbiamo osservato in 4.1, ho ricavato le medie ponderate dai risultati di ognuna delle 10 iterazioni per ognuna delle 8 domande sottoposte a valutazione e le ho confrontate con le relative medie ponderate delle valutazioni degli annotatori.

```
import pandas as pd
import numpy as np

df = pd.read_excel("/content/drive/MyDrive/Tesi Paolo Feccia/annotatori.xlsx")

ratings = df[["ANN1", "ANN2", "ANN3", "ANN4"]].values

n, k = ratings.shape

mean_per_item = ratings.mean(axis=1)
mean_per_rater = ratings.mean(axis=0)
grand_mean = ratings.mean()

SS_total = ((ratings - grand_mean)**2).sum()
SS_items = k * ((mean_per_item - grand_mean)**2).sum()
SS_raters = n * ((mean_per_rater - grand_mean)**2).sum()
SS_error = SS_total - SS_items - SS_raters

MS_items = SS_items / (n - 1)
MS_error = SS_error / ((n - 1) * (k - 1))

ICC_3_1 = (MS_items - MS_error) / (MS_items + (k - 1) * MS_error)

print(ICC_3_1)

0.6613763454902278
```

Figura 11: Calcolo dell'ICC(3,1) effettuato con Python su Google Colab.

Come si nota in fig. 11, l'indice di ICC ricavato dopo aver confrontato i punteggi considerando il modello come un quarto annotatore è di 0,66, ossia un punteggio superiore rispetto all'ICC effettuata sui soli tre annotatori umani mostrato nel paragrafo 4.1.

Volendo ricalcare gli altri confronti fatti nel paragrafo 4.2 tra annotatori e i precedenti valori ottenuti tramite le valutazioni del modello, ho effettuato anche il calcolo del MAE sulla media dei valori degli annotatori e i valori del modello; confrontando i valori e arrotondando per eccesso si ottiene un errore medio assoluto di 17 punti, numero di poco superiore rispetto al MAE evidenziato nel paragrafo 4.2 nel confronto tra i risultati diretti del modello e degli annotatori.

Successivamente, ho effettuato anche il calcolo del coefficiente di Pearson per analizzare se le metriche seguono un andamento convergente o divergente e ho ottenuto un risultato di 0,79 punti (il risultato è una media del coefficiente misurato su coppie che includono tutti e tre gli annotatori in interazione coi risultati del modello).

In seguito, calcolando l'alfa di Krippendorff tra il *gold standard* degli annotatori espressi nelle loro medie ponderate e la media ponderata dei valori ottenuti tramite l'LLM nelle risposte alle domande mostrate in tab. 6, si è ottenuto un valore di 0,34, ossia un basso valore di affidabilità tra gli annotatori e il modello.

Riassumendo, i dati riscontrati sono i seguenti:

- ICC(3,1): 0,66;
- MAE: 17;
- Coefficiente di Pearson: 0,79;
- Alfa di Krippendorff: 0,34.

Ho compiuto gli stessi calcoli di *consistency* e di confronto tra annotatori e modello usando la versione di GPT-5.2 che userò da qui in avanti per i calcoli presenti in questo paragrafo e nei prossimi, a causa della dismissione da parte di OpenAI di GPT-4o e della necessità di adeguamento sul sistema di ETET a GPT-5.2. Mantenendo lo stesso sotto-campione di quattro utenti con le rispettive

domande di *speaking* e *writing* (S10 e W11). Ho mantenuto 10 iterazioni come nel caso precedente, ma ho calcolato direttamente la media complessiva, non parametro per parametro.

		Media punteggio	Dev. standard
BUIU-01	**S10**	41	**5**
BUIU-01	**W11**	57	**4**
BULM-01	**S10**	26	**4**
BULM-01	**W11**	51	**5**
AFSS-01	**S10**	70	**3**
AFSS-01	**W11**	59	**3**
ZFLM-02	**S10**	71	**4**
ZFLM-02	**W11**	75	**3**

Tabella 7: Calcolo della consistency del prompt tramite deviazione standard usando GPT-5.2.

Come si può notare dalla tab. 7 la deviazione standard è inferiore rispetto a quella mostrata in tab. 6, con un punteggio minimo di 3 punti, un massimo di 5 e un punteggio medio di 4 punti.

Di seguito riporto gli ulteriori calcoli che ho fatto per aggiornare il confronto tra annotazioni e i risultati del modello usando GPT-5.2:

- ICC(3,1): 0,67;
- MAE: 30;
- Coefficiente di Pearson: 0,81;
- Alfa di Krippendorff: 0,21.

4.4. Modifiche al prompt

Dopo aver valutato gli output ottenuti dai tre annotatori e dall'LLM e dopo averli confrontati, valutando anche la *consistency* del prompt utilizzato per il modello sia con GPT-4o, sia con GPT-5.2 (che utilizzeremo di qui in avanti), di seguito

voglio rendere una tabella dei valori che verranno usati successivamente per confrontare i punteggi che otterrò con le modifiche al prompt, al fine di fare comparazioni utili all'individuazione di quelle modifiche che hanno un effetto positivo e quelle che hanno un effetto negativo sul rendimento del modello. Si prendono le metriche usate per confrontare i punteggi ottenuti dagli annotatori e quelli ottenuti dall'LLM, come sono stati presentati nel paragrafo 4.3, perciò considerando il sotto-campione di 4 candidati e l'analisi delle rispettive risposte alle domande S10 e W11 nella versione con GPT-5.2. Specifico che queste metriche da sole non bastano a un'analisi esaustiva delle differenze che occorrono tra le versioni diverse dei prompt, ma rendono un'idea generale della portata delle oscillazioni dei punteggi.

Metrica	**Punti**
MAE	30
Coefficiente di Pearson	0,81
Alfa di Krippendorff	0,21
ICC(3,1)	0,67

Tabella 8: Metriche usate in 4.3 per valutare l'aderenza tra annotatori e LLM dai risultati ottenuti dopo l'analisi della consistency *con GPT-5.2.*

Data la consistenza dell'errore medio assoluto, trovo difficile calcolare con un certo grado di affidabilità metriche come l'*accuracy* o l'*exact match*, mancando i presupposti per stabilire quantomeno un *range* che identifichi una corrispondenza "corretta" tra i risultati dei tre annotatori e quelli dell'LLM.

Passando all'analisi del prompt mostrato in 3.1.2, voglio evidenziarne di seguito i probabili punti deboli, basandomi su ciò che ho mostrato nel capitolo 1 riguardo i modi di ottimizzare i prompt per renderli più efficaci.

Una riflessione iniziale riguarda la necessità di scindere il prompt singolo in due prompt separati che curano i diversi aspetti delle domande di *writing* e di

quelle di *speaking*, data sia la differenza di pesi, sia la possibilità di focalizzare l'attenzione su aspetti precisi della produzione, come le differenze nell'uso della coerenza a causa delle spontanee ripetizioni nel parlato.

Nella prima parte del prompt, si ravvisa un uso sovraesteso del *role prompting*, come si può anche notare nella schematizzazione fornita in tab. 2, insieme alla probabile necessità di definire meglio l'oggetto di valutazione, qui descritto come "risposta a testo libero", ma forse non sufficiente per permettere al modello di inquadrare correttamente il lavoro che deve fare.

Nella seconda parte può essere necessario specificare meglio o in modo diverso l'uso che il modello deve fare della scala di punteggio da 0 a 100: questo è il punto su cui serve fare più attenzione, dato che il modo in cui il modello interpreta il punteggio è vitale per tutto il processo di valutazione. Per quanto riguarda la spiegazione del parametro *comprehension*, mi sembra che possa essere snellito e definito con più chiarezza, mentre per i modi in cui, alla fine di questa parte, si tenta di prevenire eventuali *bias*, possono essere sintetizzati meglio.

Nella terza e nella quarta parte del prompt, si evidenziano alcune possibilità come sopra, ossia di definire meglio sia i parametri sia le parti in cui si cerca di prevenire i possibili *bias*.

Nella quinta parte, oltre alla maggior chiarezza che si può fornire alla descrizione del parametro, è necessario che venga distinta in modo radicale la differenza tra la coerenza di un testo scritto e quella di un testo prodotto oralmente; per questo motivo, come descritto all'inizio del paragrafo, si procederà con la redazione di due prompt diversi per le due modalità di produzione.

Infine, nella sesta e ultima parte del prompt, si possono eliminare alcune parti che si rifanno al *role prompting* e che ritengo non necessarie, insieme a una descrizione migliorata delle informazioni contenute in questa parte che si ipotizzano essere vitali per una corretta analisi da parte del modello.

Di seguito, riporto il prompt con le modifiche ipotizzate fino a ora e con alcune correzioni nella forma. Questa versione del prompt, per facilitarne l'individuazione e l'analisi, verrà denominata versione 1.1.

> (Prompt v. 1.1) Sei un insegnante ed esaminatore madrelingua inglese esperto nella valutazione delle competenze linguistiche di parlanti inglese come lingua straniera.
> Il tuo scopo è valutare una risposta scritta [orale] a una domanda di composizione libera, in cui l'utente può esprimersi liberamente sul tema posto dalla domanda.
> Devi valutare la risposta in base ai seguenti parametri fornendo una valutazione numerica in una scala dinamica da 0 (risultato peggiore) a 100 (risultato migliore) e una spiegazione dettagliata per ciascun parametro:
> COMPREHENSION: aderenza della risposta al contesto della domanda che è stata assegnata al candidato. L'oggetto della valutazione non è l'opinione del candidato, ma se ha compreso e rielaborato il contenuto della consegna;
> CONTENT: correttezza a livello di contenuto, se rispecchia ciò che viene richiesto dalla domanda;
> GRAMMAR: correttezza a livello grammaticale e sintattico;
> LEXIS: varietà lessicale: deve essere pertinente all'argomento trattato, ma la valutazione sarà influenzata positivamente nel caso vengano utilizzati termini più vari e/o più inusuali. L'utilizzo di espressioni più colloquiali o dialettali non deve penalizzare il punteggio, tranne nei casi in cui la domanda richieda di creare un testo di natura esplicitamente formale, o in situazioni implicitamente formali;
> COHERENCE: coerenza testuale: devi valutare l'utilizzo di connettori, strutture sintattiche e/o strategie di organizzazione del discorso appropriati per la comprensione il più possibile fluente del testo. [Se il candidato fa delle ripetizioni o riformulazioni puoi tollerarle.]

> Se il candidato ha risposto in una lingua diversa dall'inglese, tutti i punteggi delle valutazioni devono essere a 0. I punteggi dei vari parametri devono essere indipendenti gli uni dagli altri: la valutazione di uno non può influenzare la valutazione dell'altro, e una stessa argomentazione non può essere addotta a supporto della valutazione di due o più parametri diversi.

Le parti tra parentesi quadre vengono sostituite o inserite solo nel prompt che riguarda la valutazione di prove orali. In queste ultime, come già detto nel capitolo 3 al paragrafo 3.1.1, i criteri di valutazione differiscono dai criteri usati nelle prove scritte solo per il criterio di *pronunciation* incorporato con quello di *coherence*, che insieme hanno un peso del 10% sulla percentuale totale.

Successivamente, ho tentato di apportare ulteriori modifiche al prompt v. 1.1 specificando di seguire le linee guida CEFR nella prima parte del prompt come di seguito. Queste modifiche danno origine alla versione 1.2.

> (Prompt v. 1.2) Sei un insegnante ed esaminatore madrelingua inglese esperto nella valutazione delle competenze linguistiche di parlanti inglese come lingua straniera e, per assegnare il tuo punteggio, ti basi sulla Classificazione CEFR del Quadro Comune Europeo di Riferimento per le lingue. [...]

Presento di seguito in tab. 9 i risultati ottenuti dal prompt nella versione 1.1 e 1.2 a confronto con la versione 1.0. Con tutti i prompt sono state effettuate 10 iterazioni per testare anche la *consistency* del punteggio e minimizzare l'errore.

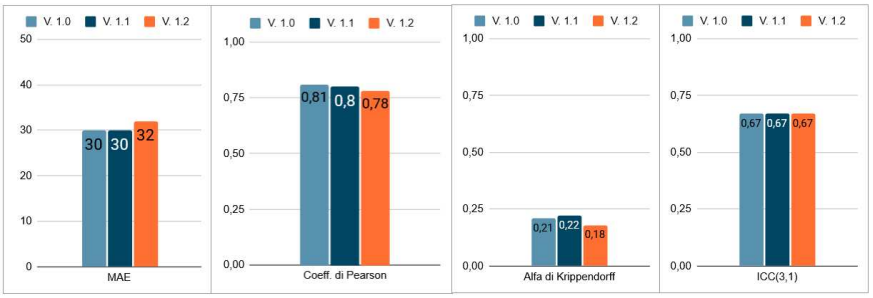


Tabella 9: Confronto dei risultati ottenuti dalle versioni 1.0, 1.1 e 1.2 del prompt.

Ora, dopo aver illustrato alcune osservazioni su come migliorare la struttura del prompt, basandomi su ciò che ho illustrato in 1.1.1, e dopo aver valutato le differenze attraverso le metriche mostrate in tab. 8, proseguirò integrando alcune delle soluzioni presentate in 1.1.2 al fine di ottimizzare ulteriormente il prompt tramite tecniche già collaudate nella letteratura presentata. Verranno analizzate solo le tecniche che sono effettivamente utilizzabili con l'interfaccia di ETET, dato che questa fornisce direttamente i punteggi dei cinque criteri impostati, insieme a una breve valutazione scritta, che useremo per andare più a fondo nelle motivazioni che hanno spinto il modello a conferire un determinato punteggio. Di seguito, si userà come versione "base" del prompt la v. 1.1, data la minima differenza con la v. 1.0 e le *performance* migliori rispetto alla 1.2.

Innanzitutto, una delle tecniche che manterremo sarà sicuramente il *role prompting*, data l'ampia letteratura in merito al suo utilizzo (White et al. 2023; Chen et al. 2025; Schulhoff et al. 2025), ma in una forma più snella e meno invasiva, come già presentata nel prompt 1.1. Altri due *pattern* illustrati in 1.1.2 che sono stati usati nella redazione delle versioni già mostrate del prompt sono il *template pattern* e il *reflection pattern*: il primo viene utilizzato implicitamente per impostare la sequenza di informazioni che l'LLM deve elaborare , come, ad esempio, la sequenza dei criteri di valutazione; il secondo viene parzialmente innescato dalla frase con cui si richiede al modello di fornire una spiegazione per giustificare le valutazioni che assegna per ogni criterio (White et al. 2023). Successivamente, dato quanto detto in Schulhoff et al. (2025: 20) riguardo le *low-resourced languages*, modificherò la lingua del prompt dall'italiano all'inglese per osservare se gli effetti dell'enorme quantità di dati di cui questa lingua dispone hanno una ricaduta sulle prestazioni del prompt. Inoltre userò due ulteriori *pattern* tra quelli mostrati, ad esempio, in Sahoo et al. (2024) e in Schulhoff et al. (2025), poiché si addicono particolarmente alla struttura d'uso del modello attraverso il portale ETET: il *few-shot prompting* e il *chain-of-thought*.

Di seguito, riporto il prompt tradotto in inglese e consistente in una versione a sé stante, la v. 1.3, che è stata sottoposta a test di comparazione con il *gold standard* degli annotatori per analizzare le differenza con la versione 1.1, il suo corrispettivo in italiano. I valori ricavati dal test con 10 iterazioni per ogni domanda di *speaking* e *writing* dei 4 utenti del sotto-campione verranno presentati di seguito al prompt.

> (Prompt v. 1.3) You are a native English teacher and examiner experienced in assessing the language skills of speakers of English as a foreign language.
> Your task is to evaluate a written [an oral] response to an open-ended question, in which the user is free to express their ideas on the given topic.
> You must assess the response according to the following parameters, assigning a numerical score on a dynamic scale from 0 (lowest performance) to 100 (highest performance), and provide a detailed justification for each parameter:
> COMPREHENSION: the candidate's response's adherence to the context of the question assigned to the candidate. The assessment is not based on the candidate's opinion, but rather on whether the candidate has understood and elaborated on the content of the assignment;
> CONTENT: content accuracy, whether it reflects the question's requirements;
> GRAMMAR: grammatical and syntactical accuracy;
> LEXIS: lexical variety: must be relevant to the topic being covered, but the assessment will be positively influenced by the use of more varied and/or unusual terms. The use of more colloquial or dialectal expressions should not penalize the score, except in cases where the question requires the creation of an explicitly formal text, or in implicitly formal situations;
> COHERENCE: textual coherence: you must evaluate the use of appropriate connectors, syntactic structures, and/or discourse organization

strategies to ensure the most fluent understanding of the text. [If the candidate uses repetitions or rephrasing, this may be tolerated.]

If the candidate responded in a language other than English, all assessment scores must be 0. The scores for the various parameters must be independent of one another: the evaluation of one cannot influence the evaluation of the other, and the same argument cannot be used to support the evaluation of two or more different parameters.

I dati ricavati sono i seguenti:

- ICC(3,1): 0,67;
- MAE: 32;
- Coefficiente di Pearson: 0,81;
- Alfa di Krippendorff: 0,18.

Considerando l'aumento dell'errore medio assoluto e il calo dell'alfa di Krippendorff nel prompt v. 1.3, per apportare le ulteriori modifiche che includono il *few-shot prompting* e il *chain-of-thought* userò ancora la versione 1.1 del prompt. Per applicare il *few-shot prompting*, non potendo fornire esempi precisi per ogni fascia di punteggio di ogni domanda, ho preferito fornire in coda ai criteri delle linee guida complessive per orientare il punteggio. Come scala di punteggio ho usato quella mostrata in tab. 1 (paragrafo 3.1.1) e ho intersecato il punteggio numerale con una descrizione ispirata al corrispettivo grado CEFR. Riporto di seguito questa sezione aggiuntiva che ho posto subito dopo la descrizione dei criteri e prima delle istruzioni finali del prompt v. 1.1. Ho nominato questa versione del prompt "v. 1.4".

(Prompt v. 1.4) [...] Per valutare le risposte degli utenti, userai la seguente tabella che fa corrispondere un giudizio qualitativo a un valore numerico e la userai per determinare il punteggio di ogni singolo criterio e poi, sommandoli, il punteggio totale della domanda dell'utente:

- 0-30: l'utente riesce a usare espressioni basilari e formule di uso quotidiano per descrivere concetti molto semplici;

- 31-40: l'utente riesce a comunicare informazioni semplici e di routine, su argomenti familiari e abituali. Riesce a descrivere in termini semplici aspetti del proprio vissuto e del proprio ambiente ed elementi che si riferiscono a bisogni immediati;
- 41-50: l'utente sa produrre testi semplici e coerenti su argomenti che gli siano familiari o siano di suo interesse. È in grado di descrivere esperienze e avvenimenti, sogni, speranze, ambizioni, di esporre brevemente ragioni e dare spiegazioni su opinioni e progetti;
- 51-70: l'utente sa produrre testi chiari e articolati su un'ampia gamma di argomenti ed esprimere un'opinione su un argomento d'attualità, esponendo i pro e i contro delle diverse opzioni;
- 71-85: l'utente sa produrre testi chiari, ben strutturati e articolati su argomenti complessi, mostrando di saper controllare le strutture discorsive, i connettivi e i meccanismi di coesione;
- 86-100: l'utente si esprime spontaneamente, in modo molto scorrevole e preciso e rende distintamente sottili sfumature di significato anche in situazioni piuttosto complesse. [...]

Riguardo, invece, il *chain-of-thought*, l'ho applicato inserendo nella prima parte di istruzioni del prompt una semplice frase che induce il ragionamento e, poi, ho potenziato il suo effetto migliorando la scrittura del prompt e dividendolo in parti più chiare attraverso titoli, spaziature e un elenco numerato dei criteri, creando una nuova versione che prende il nome di "v. 1.5":

> (Prompt v. 1.5) """" Istruzioni """"
> Sei un insegnante ed esaminatore madrelingua inglese esperto nella valutazione delle competenze linguistiche di parlanti inglese come lingua straniera.

Il tuo scopo è valutare una risposta scritta [orale] a una domanda di composizione libera, in cui l'utente può esprimersi liberamente sul tema posto dalla domanda.
Devi valutare la risposta in base ai seguenti parametri fornendo una valutazione numerica in una scala dinamica da 0 (risultato peggiore) a 100 (risultato migliore) e una spiegazione dettagliata per ciascun parametro.
Ragiona passo per passo e in ordine.

“““ Criteri di valutazione ”””

1. COMPREHENSION: aderenza della risposta al contesto della domanda che è stata assegnata al candidato. L'oggetto della valutazione non è l'opinione del candidato, ma se ha compreso e rielaborato il contenuto della consegna;
2. CONTENT: correttezza a livello di contenuto, se rispecchia ciò che viene richiesto dalla domanda;
3. GRAMMAR: correttezza a livello grammaticale e sintattico;
4. LEXIS: varietà lessicale: deve essere pertinente all'argomento trattato, ma la valutazione sarà influenzata positivamente nel caso vengano utilizzati termini più vari e/o più inusuali. L'utilizzo di espressioni più colloquiali o dialettali non deve penalizzare il punteggio, tranne nei casi in cui la domanda richieda di creare un testo di natura esplicitamente formale, o in situazioni implicitamente formali;
5. COHERENCE: coerenza testuale: devi valutare l'utilizzo di connettori, strutture sintattiche e/o strategie di organizzazione del discorso appropriati per la comprensione il più possibile fluente del testo. [Se il candidato fa delle ripetizioni o riformulazioni puoi tollerarle.]

“““ Istruzioni finali ”””

> Se il candidato ha risposto in una lingua diversa dall'inglese, tutti i punteggi delle valutazioni devono essere a 0. I punteggi dei vari parametri devono essere indipendenti gli uni dagli altri: la valutazione di uno non può influenzare la valutazione dell'altro, e una stessa argomentazione non può essere addotta a supporto della valutazione di due o più parametri diversi.

Per redigere questa versione del prompt ho usato una tecnica presentata in Chen et al. (2025) che consiste nell'evidenziare alcune parti del testo di particolare importanza con le triple virgolette; nel prompt v. 1.5, ho evidenziato in questo modo i tre titoli che ho dato alle tre sezioni (con spaziature integrate). Inoltre, ho introdotto la frase "Ragiona passo per passo e in ordine" per stimolare il *chain-of-thought* e ho ulteriormente diviso la parte dei criteri di valutazione in ordine numerale, così da rendere più chiara la sequenza di passaggi da seguire per la valutazione.

Di seguito, riporto i dati raccolti sulle versioni del prompt 1.4 e 1.5 in comparazione con la versione 1.1.

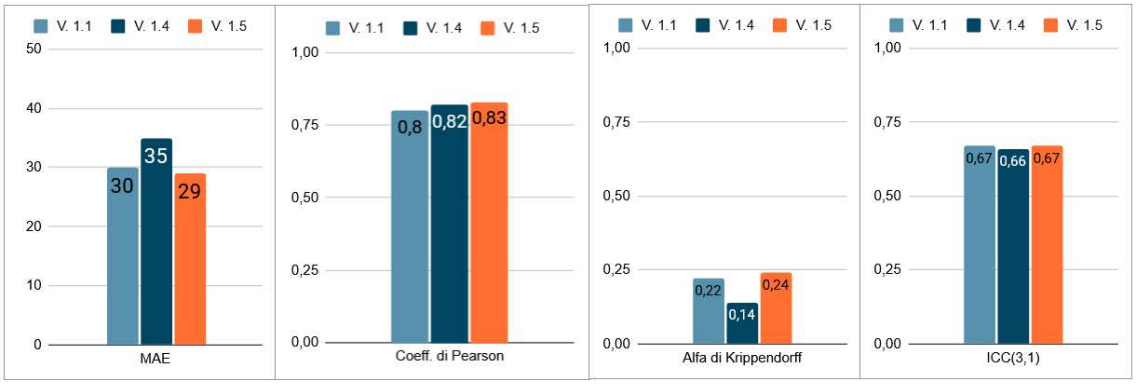


Tabella 10: Confronto dei risultati ottenuti tra le versioni del prompt 1.1, 1.4 e 1.5.

Dati gli apparenti miglioramenti nei punteggi ottenuti tramite la versione 1.5 del prompt, ho tentato di integrare indipendentemente le modifiche apportate nelle altre versioni del prompt, con in aggiunta altre modifiche che spiegherò più avanti. Innanzitutto, ho integrato nella v. 1.5 le variazioni riguardanti il CEFR

apportate nella v. 1.2 (generando il prompt in una versione denominata “v. 1.6”), poi ho modificato la lingua dall’italiano all’inglese (generando una versione denominata “v. 1.7”), infine ho integrato la “griglia” della v. 1.4 (generando una versione chiamata “v. 1.8”). Di seguito, mostro le sezioni modificate e i risultati ottenuti in queste tre nuove versioni in comparazione con quelli della v. 1.5.

> (Prompt v. 1.6) “““ Istruzioni ”””
> Sei un insegnante ed esaminatore madrelingua inglese esperto nella valutazione delle competenze linguistiche di parlanti inglese come lingua straniera e, per assegnare il tuo punteggio, ti basi sulla Classificazione CEFR del Quadro Comune Europeo di Riferimento per le lingue. [...]

> (Prompt v. 1.7) “““ Instructions ”””
> You are a native English teacher and examiner experienced in assessing the language skills of speakers of English as a foreign language.
> Your task is to evaluate a written (an oral) response to an open-ended question, in which the user is free to express their ideas on the given topic.
> You must assess the response according to the following parameters, assigning a numerical score on a dynamic scale from 0 (lowest performance) to 100 (highest performance), and provide a detailed justification for each parameter:
> Let’s think step by step and in order.
>
> “““ Evaluation criteria ”””
> 1. COMPREHENSION: the candidate’s response’s adherence to the context of the question assigned to the candidate. The assessment is not based on the candidate’s opinion, but rather on whether the candidate has understood and elaborated on the content of the assignment;
> 2. CONTENT: content accuracy, whether it reflects the question’s requirements;
> 3. GRAMMAR: grammatical and syntactical accuracy;

4. LEXIS: lexical variety: must be relevant to the topic being covered, but the assessment will be positively influenced by the use of more varied and/or unusual terms. The use of more colloquial or dialectal expressions should not penalize the score, except in cases where the question requires the creation of an explicitly formal text, or in implicitly formal situations;
5. COHERENCE: textual coherence: you must evaluate the use of appropriate connectors, syntactic structures, and/or discourse organization strategies to ensure the most fluent understanding of the text. If the candidate uses repetitions or rephrasing, this may be tolerated.

““Final Instructions”””
If the candidate responded in a language other than English, all assessment scores must be 0. The scores for the various parameters must be independent of one another: the evaluation of one cannot influence the evaluation of the other, and the same argument cannot be used to support the evaluation of two or more different parameters.

(Prompt v. 1.8) [...] “““ Griglia di valutazione ”””
Per valutare le risposte degli utenti, userai la seguente tabella che fa corrispondere un giudizio qualitativo a un valore numerico e la userai per determinare il punteggio di ogni singolo criterio e poi, sommandoli, il punteggio totale della domanda dell’utente:
0-30: l’utente riesce a usare espressioni basilari e formule di uso quotidiano per descrivere concetti molto semplici; [...]

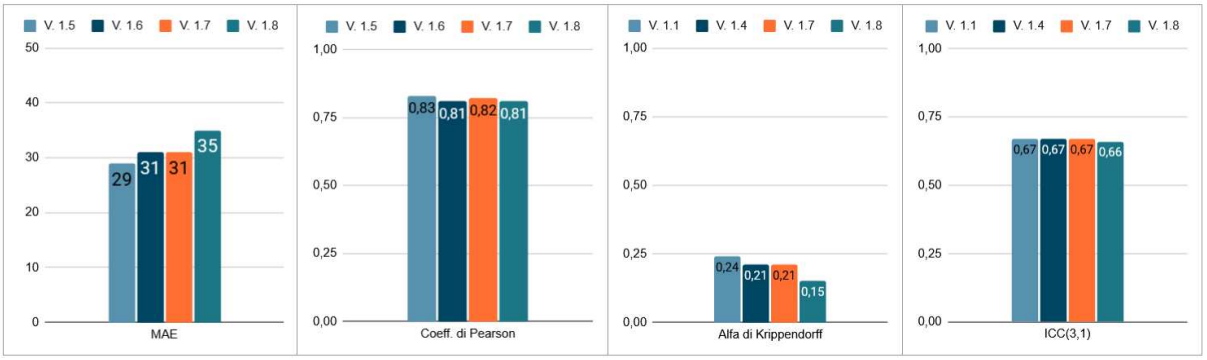


Tabella 11: Confronto dei risultati ottenuti tra le versioni del prompt 1.5 e le versioni 1.6, 1.7 e 1.8.

4.5 Discussione

Dopo aver mostrato le differenze tra le valutazioni effettuate dagli annotatori e quelle effettuate dall'LLM con il prompt di base (v. 1.0) e dopo aver apportato, sulla scorta della teoria presentata nel capitolo 1, opportune modifiche al prompt in modo tale da avvicinare il più possibile le *performance* al *gold standard*, discuto, in questo paragrafo, alcune caratteristiche che appaiono evidenti dai dati riportati nel paragrafo precedente.

La particolarità d'uso dell'LLM adoperato nel presente lavoro, ossia la sua accessibilità attraverso la web-app di ETET e il suo utilizzo di singoli valutatori che interrogano il modello con prompt separati e indipendenti, negando la possibilità di inserire multipli prompt in una sola *call*[7] con il modello, ha ridotto drasticamente sia le possibilità di usare metodi di ottimizzazione del prompt, soprattutto quelli che puntano sul *feedback* diretto del modello o sul *meta-prompting*, sia le possibilità d'uso di tecniche di valutazione automatica, data anche l'impossibilità di esportare il modello.

Un'altra difficoltà riscontrata nell'analisi riguarda la natura degli output forniti dal modello: da un lato, essendo questi di natura numerica ordinale, sono stati trovati inefficaci i metodi di valutazione automatica, dato che puntano a

[7] Si intende la singola richiesta inviata in cui si fornisce un input (prompt) e si riceve un output.

giudicare nella maggioranza di casi un output testuale (ho fatto dei tentativi con G-EVAL e BERTScore, citati nei paragrafi 1.1.2 e 1.1.4); dall'altro lato, gli output testuali, che pur sono presenti, generati a commento dei singoli criteri di valutazione non sono stati supportati da un *gold standard* ricavato da annotatori umani, pertanto non ho avuto un punto di riferimento per procedere con un'analisi testuale supportata da un confronto con risultati forniti da esseri umani.

In fase di creazione del test di inglese e delle domande sono sorti ulteriori problemi tecnici che hanno riguardato le tempistiche fornite e, quindi, l'ottenimento delle risposte: per un difetto costitutivo del sistema, allo scadere del tempo, l'utente che non ha consegnato la risposta entro i limiti si è visto annullare totalmente la risposta inserita, penalizzando gravemente molti dei test, non potendo discriminare tra una consegna mancata dovuta alla mancanza di conoscenze o a una svista nell'atto della consegna. Per questo motivo si è optato per la scelta di un campione ristretto di 10 partecipanti, così da ottimizzare i dati e minimizzare i risultati nulli, oltre che per rappresentare meglio la popolazione che si è interpellata, arginando *bias* di sorta.

Stando a quanto mostrato tramite i dati presentati nel paragrafo 4.1, l'accordo tra i tre annotatori presenta dei problemi di insufficienza e, nonostante questi dati siano stati utilizzati in seguito per i confronti coi risultati del modello, appare chiaro che non possono essere ritenuti del tutto affidabili, dato, tra l'altro, il numero esiguo degli annotatori interpellati.

Per quanto riguarda l'interazione tra i risultati emersi da annotatori e LLM presentati in 4.2, osservo che anche questi presentano gravi difformità e testimoniano uno scarso accordo tra i due fronti diversi di valutazione. È necessario specificare che, volendo mantenere le condizioni iniziali dell'esperimento, ho mantenuto l'analisi delle *performance* che è stata effettuata in sede di prima analisi a inizio 2025, come ho già specificato; per questo motivo, sono presenti errori da parte del modello che non sono stati corretti, per mantenere fedeltà a quei dati già estratti. Però, non ho tenuto conto, in fasi di analisi successive, di quei dati per fare altri confronti; per questi, infatti, ho usato

i dati rilevati dall'analisi più approfondita descritta nel paragrafo 4.3, con cui sono stati eseguiti test di *consistency* e poi gli altri confronti che fanno da base a quelli presentati nel paragrafo 4.4.

Nel paragrafo 4.3, per agevolare l'analisi della *consistency* del prompt, ho optato per la selezione di un sotto-campione, così da osservare meglio le differenze criterio per criterio e le relative deviazioni standard, selezionando gli utenti per differenze di estensione e di qualità nelle risposte. I dati ottenuti mostrano che l'oscillazione nelle medie dei punteggi è considerevole, ma non sempre tale da inficiare il punteggio e da attribuire un valore CEFR diverso.

BUIU-01	S10	lexis	content	grammar	coher + pro	comprehension	MEDIA
		7	4	5	2	4	5
BUIU-01	W11	lexis	content	grammar	coherence	comprehension	
		4	7	5	6	5	5
BULM-01	S10	lexis	content	grammar	coher + pro	comprehension	
		12	10	13	5	14	11
BULM-01	W11	lexis	content	grammar	coherence	comprehension	
		8	7	6	6	4	6
AFSS-01	S10	lexis	content	grammar	coher + pro	comprehension	
		5	6	5	2	5	5
AFSS-01	W11	lexis	content	grammar	coherence	comprehension	
		4	6	5	5	6	5
ZFLM-02	S10	lexis	content	grammar	coher + pro	comprehension	
		5	5	5	3	8	5
ZFLM-02	W11	lexis	content	grammar	coherence	comprehension	
		3	6	3	7	5	5

Tabella 12: Valutazione di consistency *criterio per criterio con prompt v. 1.0 e GPT-4o.*

Per spiegare meglio, prendendo d'esempio le iterazioni che riguardano la domanda S10 dell'utente BUIU-01, l'oscillazione del punteggio porta, nella media totale, ad avere punteggi che vanno da un minimo di 54 punti a un massimo di 65 punti, che, confrontati con la scala CEFR-ETET (tab. 1), portano l'utente ad essere inquadrato in entrambi i casi limite nella categoria B2. Invece, diverso è il caso in cui la deviazione standard si fa più consistente, come nella domanda S10 dell'utente BULM-01: qui, il punteggio medio oscilla da un minimo di 23 a un massimo di 60 punti; il *range* che si configura va da una categoria A1 nel caso peggiore, a un B2 nel caso migliore. In questo caso, in base

al range in cui si configura il punteggio, la deviazione può inficiare di molto il voto, scavalcando le diverse categorie, oppure può muoversi all'interno delle categorie stesse, non denotando differenze degne di nota. È chiaro che la discriminante, in questo caso, sta anche nelle fasce di punteggio che vengono assegnate alle singole categorie: in ETET si è deciso di attribuire le fasce di punteggio illustrate in tab. 1 tentando di interpretare numericamente quanto si evince dalla classificazione CEFR, ossia fornendo un indice speculare che restituisca gli stessi rapporti tra le classi.

Il discorso si infittisce ulteriormente al giungere, alla fine del paragrafo 4.3, della necessità di adeguare il sistema a GPT-5.2 a causa della dismissione di GPT-4o, fino a quel momento usato per le analisi. Questo contrattempo, però, mi ha dato la possibilità di osservare i cambiamenti che intercorrono tra i due modelli, soprattutto nei riguardi della *consistency*, che, come mostrato in tab. 7, ha un'omogeneità maggiore rispetto a GPT-4o. Anche facendo un'analisi sui singoli criteri, la deviazione standard si dimostra molto più contenuta rispetto al modello precedente e garantisce una maggior coerenza del risultato.

BUIU-01	S10	lexis	content	grammar	coher + pro	comprehension	MEDIA
		4	4	7	2	7	5
BUIU-01	W11	lexis	content	grammar	coherence	comprehension	
		2	4	4	4	5	4
BULM-01	S10	lexis	content	grammar	coher + pro	comprehension	
		4	2	6	2	6	4
BULM-01	W11	lexis	content	grammar	coherence	comprehension	
		3	3	7	8	7	5
AFSS-01	S10	lexis	content	grammar	coher + pro	comprehension	
		2	3	3	1	4	3
AFSS-01	W11	lexis	content	grammar	coherence	comprehension	
		2	5	3	3	3	3
ZFLM-02	S10	lexis	content	grammar	coher + pro	comprehension	
		3	4	4	2	4	4
ZFLM-02	W11	lexis	content	grammar	coherence	comprehension	
		3	3	2	3	4	3

Tabella 13: Valutazione di consistency *criterio per criterio con prompt v. 1.0 e GPT-5.2.*

Come in parte già osservato a commento di tab. 7, i dati mostrati in tab. 13 mostrano che la deviazione standard nell'uso di GPT-5.2 è inferiore rispetto a

quella mostrata nell'uso di GPT-4o, pertanto si può sostenere che i risultati di questo modello, a parità di prompt (in questo caso, la v. 1.0), abbiamo una consistenza, e quindi un'affidabilità, maggiore. La questione relativa alle fasce CEFR permane anche qui, nonostante lo scarto inferiore possa assicurare un'appartenenza più stabile in una delle categorie dal livello A1 al livello C2; se anche questa appartenenza non fosse nominale, lo sarebbe nei fatti, dato il breve scarto che occorre tra i differenti risultati ottenuti. A differenza di quanto riportato in tab. 12, nei dati mostrati in tab. 13 il picco di deviazione per criterio singolo giunge a 8 punti, contro i 14 punti visti precedentemente in tab. 12 nei risultati della domanda S10 di BULM-01.

Prima di proseguire con l'analisi dei dati presentati nel paragrafo 4.4, voglio sfruttare i dati di *consistency* presentati nelle tab. 12 e 13 per sottolineare alcuni fenomeni che riguardano soprattutto la differenza tra le domande di *writing* e quelle di *speaking*. Infatti, è chiaro che vi sono delle differenze e delle tendenze nel modo in cui il modello approccia le domande di questi due tipi. Ho già accennato al fatto che vengono valutate con sistemi diversi: le domande di *writing* vengono valutate direttamente dal modello in ogni loro aspetto, quelle di speaking vengono "divise" in due parti, in cui la prima, quella audio, viene valutata sulla pronuncia da Azure, la seconda, la trascrizione dell'audio, viene valutata da GPT-5.2 per assegnare un punteggio agli altri criteri. Questa differenza si riflette in due aspetti. Il primo è la differenza di *consistency* tra il punteggio di pronuncia assegnato da Azure e tutti gli altri; il punteggio di pronuncia non cambia mai in base alle iterazioni, ha una deviazione standard pari a 0. Il secondo aspetto si riflette nel momento in cui il modello con comprende che la trascrizione della risposta è tale e tratta la risposta come la risposta a una domanda di *writing*. A questo secondo aspetto, che è una vera e propria debolezza del modello, si è tentato di riparare specificando nei vari prompt quando il modello ha a che fare con una risposta pensata per essere scritta e quando ha a che fare con la trascrizione di una risposta orale. Detto questo, rifacendomi ai dati mostrati in tab. 12 e 13, ma anche in modo più sintetico in

tab. 6 e 7, noto che vi sono differenze tra le domande scritte e quelle orali. In particolare, come si nota nelle tab. 6 e 7, in 3 casi su 4 gli utenti conseguono punteggi inferiori nelle domande di speaking e queste sono quelle che in assoluto hanno i punteggi più bassi. Guardando più da vicino le deviazioni nelle tab. 12 e 13, si nota che la situazione è più eterogenea, con un 50% di casi in cui le domande di *speaking* accumulano un punteggio più alto, e viceversa nell'altra metà dei casi. Concentrandoci sui singoli parametri, nei dati ottenuti con GPT-4o (tab. 12) non si riscontrano delle particolari tendenze, al di là delle oscillazioni che coinvolgono i singoli utenti; l'unico dato interessante è che, considerando le medie per singolo parametro delle deviazioni, quello che risulta inferiore è proprio il criterio "*coherence* + *pronunciation*", dato il discorso che abbiamo fatto in precedenza sulla particolare *consistency* della valutazione della pronuncia. Per quanto riguarda quelli ottenuti con GPT-5.2 (tab. 13), in media la deviazione è superiore nel parametro di *comprehension*. Questa fluttuazione è probabilmente dovuta alla difficoltà nell'inquadrare precisamente l'oggetto di valutazione su cui deve concentrarsi il prompt e, a fronte di alcuni parametri più facilmente riscontrabili, quello della comprensione del contesto ha i contorni più "sfumati" e, pertanto, può risultare più complesso per il modello constatarlo nella risposta dell'utente.

A conclusione dell'analisi dei risultati ottenuti nel paragrafo 4.3, voglio discutere le differenze emerse nelle metriche tra GPT-4o e GPT-5.2. Prima di tutto, non ho intenzione di metterle a confronto con i dati ottenuti nei paragrafi 4.1 e 4.2 a causa del differente bacino di risposte degli utenti che ho sfruttato: nei primi due paragrafi del capitolo, ho usato le metriche di valutazione su tutto il bacino del campione di 10 utenti; invece, nel paragrafo 4.3, ho valutato il prompt usando il sotto-campione di 4 utenti e le loro risposte alle domande S10 e W11. Pertanto, non è possibile fare dei confronti fedeli, ma questi dati verranno usati per fare confronti coi dati del paragrafo 4.4, dato che si è usato lo stesso sotto-campione. Tornando alle metriche presenti al termine del paragrafo 4.3, che riporto qui sotto in tab. 14, è interessante notare che alcune metriche volgono a

favore del primo modello, alcune a favore del secondo: in particolare, il MAE e l'alfa di Krippendorff hanno avuto risultati migliori con il primo modello, mentre l'ICC e il coefficiente di Pearson con il secondo modello. Come già presentato nel confronto tra tab. 6 e 7, GPT-5.2 ha una consistency migliore e lo notiamo dall'aumento delle metriche coinvolte. Il valore MAE, tuttavia, ci dimostra una distanza crescente rispetto a quanto accadeva con GPT-4o, ma rappresenta comunque dei valori che, sebbene più distanti, hanno una coerenza interna più forte. La misurazione dell'alfa di Krippendorff si attesta in controtendenza con quanto detto, dimostrando un accordo maggiore tra annotatori e modello usando GPT-4o, giustificato dallo scarto maggiore che si evince anche dal calcolo del MAE.

Metriche	GPT-4o	GPT-5.2
ICC(3,1)	0,66	0,67
MAE	17	30
Coeff. di Pearson	0,79	0,81
Alfa di Krippendorff	0,34	0,21

Tabella 14: Confronto delle metriche tra GPT-4o e GPT-5.2.

Passando ora a un'indagine più approfondita del paragrafo 4.4 e volendo mostrare un'analisi quantitativa dei dati presentati sulle differenze tra le diverse versioni modificate dei prompt, è necessario osservare che, a fronte delle metriche utilizzate per fare confronti (MAE, Coefficiente di Pearson, Alfa di Krippendorff e ICC[3,1]), i cambiamenti occorsi coinvolgono in modo degno di nota soltanto alcune di queste. Nei fatti, e come si può notare dai dati riportati nel paragrafo precedente, l'ICC(3,1) subisce solo in alcuni casi delle oscillazioni sostanziose, muovendosi sempre tra 0,66 e 0,67 punti. Le altre metriche hanno oscillazioni più consistenti, ma, in ogni caso, si terrà conto di tutte quelle usate per proporre un'analisi più completa. Nel paragrafo 4.4 e nel presente paragrafo, ho proceduto col valutare positivamente o negativamente le varie versioni dei

prompt riferendomi ai dati impostati come *benchmark* e ottenuti dal confronto tra annotatori e LLM in seguito ai test di *consistency* con GPT-5.2. Pertanto, quando, a fronte delle metriche, il prompt riesce a uguagliare o migliorare almeno la maggioranza dei punteggi del *benchmark* è stato considerato positivamente, in caso contrario il prompt (e la tecnica usata) è stata considerata negativamente.

Dopo aver scelto il *benchmark* sopra citato e dopo aver sottolineato alcuni punti deboli del prompt v. 1.0, ho proposto una versione rivisitata di quest'ultimo seguendo le linee teoriche presentate nei paragrafi 1.1.1 e 1.1.2, e, da questa versione rivisitata, ho tratto un'ulteriore versione che fa cenno, nella parte introduttiva, alle valutazioni CEFR. I dati di queste due versioni modificate (v. 1.1 e 1.2) sono stati mostrati in tab. 9 e illustrano minime differenze nelle oscillazioni di punteggio, ma rappresentano chiaramente delle prestazioni inferiori riguardo la versione 1.2: la citazione dei criteri CEFR o il modo in cui è stata fatta non ha giovato al modello nel portare a termine in modo migliore il task. Dato che, all'infuori dell'ICC, tutti gli altri parametri dimostrano un calo nei punteggi: MAE e coefficiente di Pearson di 2 punti (quest'ultimo di 3 punti se messo a confronto col prompt v. 1.0), alfa di Krippendorff di 4 punti rispetto alla v. 1.1. Per questo motivo, questa conformazione del prompt è stata abbandonata in attesa di comprendere se vale la pena integrare il riferimento al CEFR e come questo possa giovare al modello.

Invece, il confronto tra v. 1.0 e 1.1 si fa più interessante: a parità di ICC e MAE, presentano uno scarto di un punto reciproco a favore del coefficiente di Pearson nella v. 1.0 e a favore dell'alfa di Krippendorff nella v.1.1. Questo indica che i risultati del prompt originale hanno una migliore correlazione lineare, mentre il prompt modificato presenta una migliore affidabilità tra le valutazioni.

Le ulteriori versione 1.3 e 1.4, create sulla base del prompt v. 1.1, non hanno dato i risultati sperati, dato l'aumento del MAE e il calo dell'alfa di Krippendorff. Nel caso peggiore, il prompt 1.4, si nota un aumento di 2 punti del coeff. di Pearson, ma si evince un grave calo in tutte le altre metriche (l'alfa di Krippendorff è inferiore di 8 punti rispetto alla v. 1.1); quindi, posso asserire che

si riscontra una buona correlazione con gli annotatori, ma il modello sottostima i punteggi e li comprime verso risultati più bassi. Di conseguenza, anche in questo caso, possiamo dire che l'uso della lingua inglese e della griglia di punteggio ispirata al CEFR o il modo in cui sono state introdotte non hanno portato a miglioramenti visibili. Le supposizioni fatte in merito a un miglioramento delle prestazioni grazie allo statuto della lingua inglese come *high-resourced language* (rispetto all'italiano) non hanno trovato riscontro nella pratica, tenendo conto di possibili difetti nelle modalità della sua integrazione.

Un cambiamento sensibile avviene con l'uso del CoT e della nuova formattazione del testo e, quindi, con il prompt v. 1.5, creato tenendo come base la versione 1.1. In questo caso, le ottime prestazioni rilevate, che superano in tutte e quattro le metriche i risultati del prompt di base 1.0, sono probabilmente da attribuire in parte all'uso del CoT, in parte alla divisione in sezioni (titoli marcati, spazi aggiuntivi e ordine nei criteri). Per fugare questo dubbio, ho proceduto creando altri due prompt: una versione denominata 1.5.1 che, avendo come base il prompt 1.5, elimina solo la parte vera e propria di CoT, mantenendo tutto il resto e una versione denominata 1.1.1 in cui, prendendo come base il prompt 1.1, si aggiunge solo la frase che innesca il CoT. Di seguito riporto in tab. 15 i punteggi risultanti dai test fatti su questi due nuovi prompt, anche in comparazione con le loro versioni base.

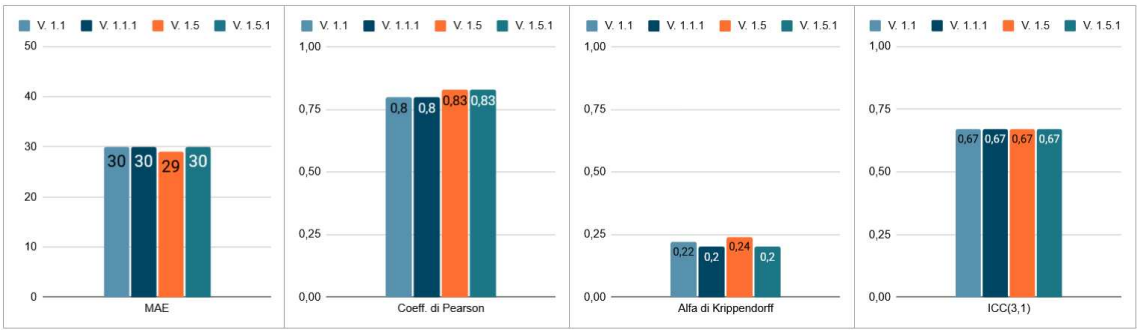


Tabella 15: Confronto tra prompt 1.1, 1.1.1, 1.5 e 1.5.1.

Come si può notare, appare chiaro quale delle due modifiche ha apportato un miglioramento più incisivo rispetto all'altra, e si tratta della formattazione diversa con spazi, titoli ed elenco numerato, dato che il prompt 1.5.1 ha conseguito dei risultati più convincenti rispetto al prompt 1.1.1. In particolare, il prompt 1.1.1 ha conseguito dei risultati peggiori rispetto alla sua versione base, mentre la versione 1.5.1 ha ottenuto un MAE inferiore, ma anche un alfa di Krippendorff inferiore, sintomo di una maggior coerenza che però si discosta di più dagli altri annotatori.

Dati i miglioramenti visibili ottenuti col prompt 1.5, ho usato nuovamente come base questo prompt per aggiungere alcune delle modifiche che avevo fatto precedentemente usando come base il prompt 1.1. Quindi, integrando rispettivamente il riferimento al CEFR in introduzione, la lingua inglese e la griglia ispirata ai criteri CEFR, ho generato i prompt 1.6, 1.7 e 1.8. Soltanto uno di questi, il prompt 1.7, ha dimostrato delle prestazioni sufficientemente soddisfacenti, uguagliando o migliorando i punteggi di 3 metriche su 4 rispetto al *benchmark*. Confrontando questi tre prompt con le loro versioni senza CoT, si può notare che in due casi su tre i punteggi sono migliorati, sia dal punto di vista del MAE, sia di quello delle metriche di correlazione tra annotatori.

Metriche/ Prompt	**1.2**	**1.3**	**1.4**	**1.6**	**1.7**	**1.8**
MAE	32	32	35	31	31	35
Coeff. di Pearson	0,78	0,81	0,82	0,81	0,82	0,81
Alfa di Krippendorff	0,18	0,18	0,14	0,21	0,21	0,15
ICC(3,1)	0,67	0,67	0,66	0,67	0,67	0,66

Tabella 16: Confronto tra prompt prima e dopo CoT.

In tab. 16, si può notare, come detto sopra, che nelle ultime tre versioni i MAE sono diminuiti, seppur di un solo punto, in 2 casi su 3; infine, la media dei

coefficienti di Pearson è più alta; la media degli alfa di Krippendorff è più alta. Questo ci porta a evidenziare i cambiamenti in positivo che hanno interessato soprattutto il confronto tra i prompt 1.2 e 1.6, con un chiaro miglioramento di quest'ultimo nell'uso dell'integrazione del riferimento al CEFR in introduzione al prompt. In particolare, il problema notato sopra della compressione dei punteggi verso il basso del prompt 1.4 si ripercuote anche sul prompt 1.8, sintomo che, evidentemente, proprio l'inserimento della griglia di valutazione vizia il punteggio nel modo dimostrato, pur mantenendo ad un livello stabile la correlazione tra annotatori. Invece, il prompt 1.6 e 1.7 hanno dei punteggi molto simili, sintomo che le modifiche riguardanti il CEFR e la lingua inglese non bastano a determinare un aumento sostanziale del punteggio.

Metriche/ Prompt	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**
MAE	✓	✗	✗	✗	✓	✗	✗	✗
Coeff. di Pearson	✗	✗	✓	✓	✓	✓	✓	✓
Alfa di Krippendorff	✓	✗	✗	✗	✓	✓	✓	✗
ICC(3,1)	✓	✓	✓	✗	✓	✓	✓	✗

Tabella 17: Griglia di valutazione complessiva delle versioni dei prompt.

Capitolo 5 Conclusioni

Riassumendo e traendo le conclusioni del presente lavoro, ho voluto presentare il funzionamento del *prompt engineering* e i modi in cui è possibile ottimizzare e valutare i prompt al fine di massimizzare il loro impatto nei task che vengono sottoposti agli LLM. Dopo aver illustrato, nel capitolo 1, le varie teorie riscontrate sull'oggetto della presente ricerca, ho proceduto presentando il lavoro di ricerca che è stato fatto internamente a ETET per redarre un prompt capace di valutare correttamente delle risposte a composizione libera di *speaking* e *writing* nei vari test di inglese creati dal *team* di ETET di cui ho fatto parte per il periodo di svolgimento del tirocinio curricolare.

Dopo aver analizzato il prompt creato insieme al *team* di ETET con un *benchmark* costruito appositamente su metriche che valutano l'errore medio assoluto e l'accordo tra gli annotatori umani interpellati e il modello, ho proceduto integrando modifiche al prompt basandomi su ciò che ho illustrato nel capitolo 1 per cercare di massimizzare l'accordo con gli annotatori e diminuire il più possibile il MAE. Nonostante nel paragrafo 1.1.4 si siano presentate svariati sistemi di valutazione automatica, in questa sede ho deciso di usare un benchmark redatto per l'occasione dati i problemi riscontrati nell'esportazione del modello e data la sua particolare struttura. Quindi, fissando come *benchmark* i punteggi ottenuti dal prompt 1.0, ho creato dieci versioni del prompt con caratteristiche diverse per isolare le singole modifiche che avrebbero potuto influenzare il punteggio. Per fare ciò, ho selezionato, tra le tecniche presentate nei paragrafi 1.1.1, 1.1.2 e 1.1.3, quelle che pensavo si sarebbero adattate nel modo migliore alla struttura del modello fornito da ETET, ossia i miglioramenti nella scrittura, che hanno portato alla sintesi di parti troppo dispersive e all'uso di titoli, divisione in sezioni e spaziature, l'uso del CoT, del *template pattern*, del *reflection pattern*, del *role prompting* e di altri accorgimenti basati su riflessioni fatte dopo l'analisi del prompt 1.0; inoltre, dato che consiste nella possibilità di sottoporre all'LLM un singolo prompt alla volta cui segue direttamente la

valutazione della domanda fornita. Come si è visto nel paragrafo 4.5, solo alcune delle tecniche usate hanno portato a dei risultati positivi. La v. 1.1 del prompt, consistente in una riscrittura più sintetica del prompt 1.0 con alcune descrizioni più chiare e alcune parti eliminate e riassunte, ha dimostrato dei miglioramenti di punteggio rispetto alla sua versione base, ma, non avendo approfondito i singoli cambiamenti apportati, è possibile solo quantificare ipoteticamente la misura in cui questi cambiamenti (maggiore precisione nella scrittura e all'eliminazione di alcune parti ridondanti e superflue) abbiano migliorato l'output. In occasioni future, sarà necessario entrare più nello specifico e analizzare modifica per modifica, in modo tale da isolare le parti utili senza ulteriori dubbi. Oltre alla v. 1.1, l'apice dei miglioramenti è stato notato con il prompt 1.5 e con le sue versioni derivate 1.6, 1.7 e 1.5.1. Secondo le evidenze che ho presentato nel paragrafo 4.5, è chiaro che questi miglioramenti (nel caso della v. 1.5 e 1.5.1 su tutte le metriche) sono dovute alla nuova formattazione che è stata data al prompt e non alla presenza del CoT tramite la frase chiave "Ragiona passo per passo e in ordine.": la divisione in sezioni con titoli marcati dalle triple virgolette, le spaziature e i criteri di valutazione posti come elenco numerato hanno, evidentemente, migliorato la comprensibilità del task e delle istruzioni da usare.

Di contro, i prompt che hanno conseguito i risultati peggiori sono le versioni 1.2, 1.4 e 1.8. Il primo di questi introduceva soltanto il riferimento al prompt tra le istruzioni iniziali, gli altri due integravano la griglia di punteggio che doveva fornire una guida più precisa nell'assegnazione delle valutazioni. Evidentemente queste integrazioni, o il modo con cui sono state redatte, non ha favorito una comprensione maggiore del task. Suppongo che una formulazione diversa possa giovare più di quella testata, magari esprimendo dei rapporti numerici più precisi e verificabili.

Nonostante i risultati incoraggianti in merito ai sistemi presentati per ottimizzare le prestazioni di prompt del tipo analizzato, è di fondamentale importanza condurre ricerche più precise sul funzionamento dei modelli e, citando Quattrociocchi, Capraro, Perc (2025), come si è già fatto nel capitolo 2, è

necessario, nell'approccio con gli LLM, tenere in considerazione che il loro modo di procedere non è comparabile a come lavorano valutatori umani. Nei fatti, una semplice analisi statistica e un metodo di estrazione di materiale linguistico basato su domande e risposte brevi non permettono di restituire un'immagine precisa e veritiera delle reali competenze di un parlante di una lingua straniera. Per questo motivo, l'obiettivo di un modello capace di comprendere realmente un contesto complesso è lungi dall'essere raggiunto con gli strumenti mostrati in questo lavoro, soprattutto di fronte a risultati che intersecano competenze nell'uso della lingua scritta e della lingua orale.

Bibliografia

Chen, B., et al. (2025). Unleashing the potential of prompt engineering for large language models, in *Patterns*, vol. 6 issue 6. https://doi.org/10.1016/j.patter.2025.101260

De Cesare, A.-M. (2023). Assessing the quality of ChatGPT's generated output in light of human-written texts. A corpus study based on textual parameters, in *CHIMERA Revista de Corpus de Lenguas Romances y Estudios Lingüísticos* 10, pp. 179-210. https://revistas.uam.es/chimera/article/view/17979

Desmond, M., et al. (2024). EvaluLLM: LLM Assisted Evaluation of Generative Outputs, in IUI '24: 29th International Conference on Intelligent User Interfaces, pp. 30-32. https://doi.org/10.1145/3640544.3645216

De Wynter, A., et al. (2025). On Meta-Prompting. https://doi.org/10.48550/arXiv.2312.06562

Dietz, L., et al. (2025). Principles and Guidelines for the Use of LLM Judges, in *ICTIR '25: International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval*, pp. 218-229.

Dorner, F. E., Nastl, V. Y., Hardt, M. (2025). Limits to scalable evaluation at the frontier: LLM as Judge won't beat twice the data. https://doi.org/10.48550/arXiv.2410.13341

Eager, B., Brunton, R. (2023). Prompting Higher Education Towards AI-Augmented Teaching and Learning Practice, in *Journal of University Teaching and Learning Practice* 20(5). https://doi.org/10.53761/1.20.5.02

Ekin, S. (2023) - Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices. https://doi.org/10.36227/techrxiv.22683919.v2

Federiakin, D., et al. (2024). Prompt engineering as a new 21st century skill, in *Frontiers in Education*, vol. 9. https://doi.org/10.3389/feduc.2024.1366434

Gallifant, J., et al. (2024). Peer review of GPT-4 technical report and systems card, in *PLOS Digital Health*, January 2024, 3(1). https://doi.org/10.1371/journal.pdig.0000417

Gao, A. (2023). Prompt Engineering for Large Language Models. https://ssrn.com/abstract=4504303

Ggaliwango, M., et al. (2024). Prompt Engineering in Large Language Models, in *Data Intelligence and Cognitive Informatics. Proceedings of ICDICI 2023*, pp. 387-402. https://doi.org/10.1007/978-981-99-7962-2_30

Gu, J., et al. (2025). A Survey on LLM-as-a-Judge. https://doi.org/10.48550/arXiv.2411.15594

Henrickson, L., Meroño-Peñuela, A. (2023). Prompting meaning: a hermeneutic approach to optimising prompt engineering with ChatGPT, in *AI & SOCIETY* 40(2), pp. 903-918. https://doi.org/10.1007/s00146-023-01752-8

Hou, Z., Ciuba, A., Li, X. (2025). Improving LLM-based Automatic Essay Scoring with Linguistic Features, in *Proceedings of the Innovation and Responsibility in AI-Supported Education Workshop, PMLR 273*, pp. 41-65.

Jurafsky, D., Martin, J. H. (2026). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. https://web.stanford.edu/~jurafsky/slp3

Knoth, N., et al. (2024). AI literacy and its implications for prompt engineering strategies, in *Computers and Education: Artificial Intelligence*, vol. 6. https://doi.org/10.1016/j.caeai.2024.100225

Korzynski, P., et al. (2023). Artificial intelligence prompt engineering as a new digital competence: Analysis of generative AI technologies such as ChatGPT, in *Entrepreneurial Business and Economics Review* 11(3), pp. 25-37. https://doi.org/10.15678/EBER.2023.110302

Krippendorff, K. (2013). Content Analysis: An Introduction to Its Methodology. Sage, California.

Lemeš, S. (2024). Prompt Engineering in *ARTIFICIAL INTELLIGENCE IN INDUSTRY 4.0: The future that comes true*, ed. Karabegović, I., pp. 159-170. https://doi.org/10.5644/PI2024.215.08

Li, H., et al. (2024). LLMs-as-Judges A Comprehensive Survey on LLM-based Evaluation Methods. https://doi.org/10.48550/arXiv.2412.05579

Lin, Y.T., Chen, Y.N. (2023). LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models, in *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023),* pp. 47–58.

Liu, Y., et al. (2024). Calibrating LLM-Based Evaluator, in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024),* pp. 2638–2656.

Loru, E., et al. (2025). The simulation of judgment in LLMs, in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 122(42). https://doi.org/10.1073/pnas.2518443122

Nudo, J., et al. (2026). Generative exaggeration in LLM social agents: Consistency, bias, and toxicity, in *Online Social Networks and Media*, vol. 51. https://doi.org/10.1016/j.osnem.2025.100344

OpenAI, et al. (2023). GPT-4 Technical Report. https://doi.org/10.48550/arXiv.2303.08774

OpenAI, et al. (2024). GPT-4o System Card. https://doi.org/10.48550/arXiv.2410.21276

Pack, A., Barrett, A., Escalante, J. (2024). Large language models and automated essay scoring of English language learner writing: Insights into validity and reliability, in *Computers and Education: Artificial Intelligence*, vol. 6. https://doi.org/10.1016/j.caeai.2024.100234

Pan, Q., et al. (2024). Human-Centered Design Recommendations for LLM-as-a-Judge, in *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pp. 16–29.

Pienemann, M. (1998). Language processing and second language development: Processability theory. Amsterdam/Philadelphia: John Benjamins.

Quattrociocchi, W., Capraro, V., Perc, M. (2025). Epistemological Fault Lines Between Human and Artificial Intelligence. https://doi.org/10.31234/osf.io/c5gh8_v1

Sahoo, P., et al. (2024). A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. https://doi.org/10.13140/RG.2.2.13032.65286

Schulhoff, S., et al. (2025). The Prompt Report: A Systematic Survey of Prompt Engineering Techniques. https://doi.org/10.48550/arXiv.2406.06608

Sclar, M., et al. (2024). Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting, in *ICLR 2024*. https://doi.org/10.48550/arXiv.2310.11324

Sedgwick, P. M. (2012). Pearson's correlation coefficient, in *The British Medical Journal, 345*. https://doi.org/10.1136/bmj.e4483

Suzgun, M., Kalai, A. T. (2024). Meta-Prompting: Enhancing Language Models with Task-Agnostic Scaffolding. https://doi.org/10.48550/arXiv.2401.12954

Tan, S., et al. (2025). JudgeBench: A Benchmark for Evaluating LLM-based Judges. https://doi.org/10.48550/arXiv.2410.12784

Vaswani, A., et al. (2017). Attention is All you Need, in *31st Conference on Neural Information Processing Systems (NIPS 2017)*.

Vignoli, A., Combei, C. R., Zappulla, F. (2025). Verso la Valutazione Automatizzata dell'Italiano L2: ETET tra LLM e Tecnologie Vocali, in *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025),* pp. 282–291.

Wang, X., et al. (2023). Self-Consistency Improves Chain of Thought Reasoning in Language Models. https://doi.org/10.48550/arXiv.2203.11171

Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models, in *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 24824 - 24837.

White, J., et al. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT, in *PLoP '23: Proceedings of the 30th Conference on Pattern Languages of Programs*, pp. 1-31. https://dl.acm.org/doi/10.5555/3721041.3721046

Xu, A., et al. (2025). Does Context Matter? ContextualJudgeBench for Evaluating LLM-based Judges, in *Contextual Settings, in Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 9541–9564. https://doi.org/10.18653/v1/2025.acl-long.470

Ye, Q., et al. (2024). Prompt Engineering a Prompt Engineer, in *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 355–385. https://doi.org/10.18653/v1/2024.findings-acl.21

Zhang, Q., et al. (2025). Crowd Comparative Reasoning: Unlocking Comprehensive Evaluations for LLM-as-a-Judge, in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 5059–5074.

Zhang, Y., Yuan, Y., Yao, A. C. (2025). Meta Prompting for AI Systems. https://doi.org/10.48550/arXiv.2311.11482

Zhou, Y., et al. (2025). Evaluating Judges as Evaluators: The JETTS Benchmark of LLM-as-Judges as Test-Time Scaling Evaluators. https://doi.org/10.48550/arXiv.2504.15253