UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Analisi Automatizzata della Qualità dei Prompt nei Modelli Linguistici: Rilevamento e Valutazione dei Prompt Smells

RELATORI

Prof. **Fabio Palomba**

Dott. **Gianmario Voria**

CANDIDATO

Raffaele Sulipano

Matricola: 0512113679

Anno Accademico 2025-2026

Ai miei genitori,

per aver sempre creduto in me.

Abstract

Nel contesto dell'aumento di interesse verso i Modelli di linguaggio di grandi dimensioni (LLMs), la qualità dei prompt con cui interrogarli è fondamentale per produrre risposte efficaci, pertinenti e non problematiche. In questa tesi si presenta il concetto di prompt smells, segnali di bassa qualità di un prompt che possono compromettere l'output del modello, in analogia ai code smells nello software engineering. Si propongono e classificano diverse tipologie di prompt smell, quali errori grammaticali, ambiguità, eccessiva lunghezza, formalità e pregiudizi. Per ciascuna categoria si associa una metrica quantitativa o semi-quantitativa, realizzata mediante tool NLP locali e con ricorso a API di LLM (OpenAI). Si descrive inoltre uno script che, eseguibile da linea di comando, analizza qualsiasi dataset di prompt fornendo un'analisi degli smell presenti. I risultati sperimentali dimostrano come tali metriche possano essere utili a valutare e correggere efficacemente prompt di bassa qualità, aprendo la strada verso strumenti di prompt engineering assistito e automatico.

Indice

Glossario

AI Artificial Intelligence.

BDS Bias Detection Score.

CLS Complexity-Length Score.

FMS Formality Mismatch Score.

LLM Modello di linguaggio di grande dimensione.

LLMs Modelli di linguaggio di grandi dimensioni.

NLP Natural Language Processing.

PQS Prompt Quality Score.

RCS Relevance Context Score.

Elenco delle figure

Elenco delle tabelle

Elenco degli algoritmi

CAPITOLO 1

Introduzione

Negli ultimi anni, i *Modelli di linguaggio di grandi dimensioni (LLMs)* hanno guadagnato un ruolo chiave in diversi campi di applicazione, dalla generazione automatica di codice alla scrittura assistita, passando per il supporto conversazionale ed la ricerca semantica [2, 6]. L'escalation delle loro potenzialità ha reso i prompt, ovvero i comandi in linguaggio naturale che si danno ai modelli, un asset strategico per ottenere risposte precise, consistenti e degne di fiducia [9]. La creazione dei prompt è tuttavia ancora in gran parte manuale e priva di pratiche consolidate, il che rende questi spesso soggetti a difetti o inefficienze difficili da rilevare ed ovviare.

Alcuni recenti lavori su *prompt engineering* [9] hanno mostrato come piccoli accorgimenti nella formulazione di un prompt possono indurre ampie differenze nei risultati ottenuti. Tale fragilità, unitamente alla crescente complessità ed opacità dei Modelli di linguaggio di grandi dimensioni (LLMs), ha reso intuitiva la necessità di disporre di strumenti e misure per investigare la qualità dei prompt. Parallelamente, lavori su bias e robustness dei modelli [1, 5] hanno evidenziato come anche prompt apparentemente innocui possano indurre bias o vulnerabilità, sottolineando l'importanza di saper valutare criticamente ciascuna interazione con i modelli

1.1 Motivazioni e Contesto

L'efficacia di *Modello di linguaggio di grande dimensione (LLM)* è fortemente correlata alla qualità degli input che gli vengono forniti. Prompt poco chiari, troppo lunghi, mal formattati o contenenti bias possono infatti diminuirne l'accuratezza, aumentare i costi e produrre contenuti fuorvianti. In contesti delicati come quello medico o legale, errori di interpretazione causati da prompt non adeguati si ripercuotono su tutti i livelli, dando ad esempio luogo a possibili implicazioni etiche. Allo stesso modo, in ambito industriale, prompt non efficaci rappresentano inefficienze, in quanto forniscono risposte di scarsa qualitÃ che vanno poi corrette manualmente.

Per descrivere ed approfondire questi aspetti, è stato coniato il termine *prompt smell*, mutuato dai *code smells* dell'ingegneria del software [10]. I *prompt smells* sono difetti relativi alla progettazione dei prompt ricorrenti, quali errori di grammatica, ambiguità, complessità innecessaria, informalità o bias. Individuarli è importante perché anche un solo difetto può diffondersi lungo intere pipeline, e diminuire la fiducia in un modello.

1.2 Terminologia

Per capire davvero cosa è stato fatto e quali sono i risultati, è bene chiarire alcuni concetti legati ai modelli di linguaggio e al prompt engineering. In questo capitolo vengono definiti i termini principali con cui si affronterà tutto il resto della tesi, in modo breve ma corretto, e con riferimento alla letteratura di settore.

1.2.1 Modelli di linguaggio di grandi dimensioni (LLMs)

I **Modelli di linguaggio di grandi dimensioni (LLMs)** sono modelli di intelligenza artificiale che, addestrati su miliardi di parole, imparano come le parole vengono utilizzate e come vengono organizzate nel linguaggio naturale. In pratica, imparano a prevedere qual Ã¨ la parola (o la frase) che può venire dopo in una determinata situazione e quindi sono in grado di generare risposte che abbiano senso e corrette dal punto di vista grammaticale.

Esempi di Modelli di linguaggio di grandi dimensioni (LLMs) sono il GPT-3 [2] ed il GPT-4 [6], che sono riusciti a comprendere istruzioni in linguaggio naturale ed eseguire complessi compiti come scrivere codice, riassumere o tradurre. Per di più questi modelli sono in grado di adattarsi a nuove richieste anche senza particolari fasi di addestramento, per

effetto di quelle che vengono denominate modalità *few-shot* e *zero-shot*, nelle quali al modello vengono forniti pochi o addirittura nessun esempio per imparare un compito.

Gli LLMs hanno stravolto il campo del **Natural Language Processing (NLP)**, ovvero l'insieme di tecniche che consentono ai computer di comprendere, analizzare e generare il linguaggio umano. Gli LLMs non sono più semplici riconosci pattern-linguistici, ma sono in grado di effettuare ragionamenti contestuali e di adattarsi dinamicamente a complesse istruzioni, e così cambiano il modo in cui noi interagiamo con le macchine

1.2.2 Prompt e Prompt Engineering

Un **prompt** è una stringa di testo fornita ad un Modello di linguaggio di grande dimensione (LLM) per guidarne la generazione del linguaggio. In altre parole, è una sorta di "natural language instruction" che specifica il task che si desidera svolgere. Ad esempio:

> *Esempio 1 (prompt ben formulato)*: "Write a 100-word summary on the environmental impact of electric vehicles, in a formal and neutral tone."
> *Esempio 2 (prompt mal formulato)*: "Summarize this text."

Nel primo caso il modello riceve indicazioni esplicite circa la lunghezza, il tono ed il topic su cui operare mentre nel secondo caso la mancanza di indicazioni chiare comporta risposte generiche o inadeguate.

La pratica di ideazione e sperimentazione di prompt prende il nome di **Prompt Engineering** [9]. Si tratta di un metodo di indagine per progettare prompt efficaci, coerenti e riproducibili al fine di incrementare la qualità dell'output generato dai modelli.

Le principali approcci sperimentati sono:

- **Zero-shot prompting**: si passa al modello una singola istruzione generica;
- **Few-shot prompting**: si passano esempi di input ed output desiderati;
- **Chain-of-thought prompting**: si spinge il modello a "pensare a voce alta" prima di fornire la risposta;
- **Instruction tuning**: si sottopone a supervisione un modello affinché impari a comprendere e soddisfare istruzioni in linguaggio naturale.

Saper scrivere prompt ben formulati è ormai considerata una delle abilità più importanti per interagire con i nuovi modelli linguistici. ò portare a dei miglioramenti nelle prestazioni del modello.

1.3 Obiettivi della Ricerca

L'identificazione e la classificazione dei *prompt smells* è il primo passo verso un approccio quantitativo per misurare la qualità del prompt.

Il termine smell si diffonde nel mondo dell'informatica con i cosiddetti **code smells** [10]. Essi sono dei segnali di cattivo design di un codice, non dei difetti veri e propri, ma dei segnali che comunicano che esso potrebbe essere meno manutenibile o leggibile di quanto non possa apparire. Prototipi di queste storie possono essere metodi eccessivamente lunghi, dipendenze cicliche o nomi di variabili poco chiari.

Per classificazione dei *prompt smells* si intende infatti il riconoscimento e la classificazione in categorie dei più comuni difetti riscontrati nel prompt. Una classificazione è uno degli scopi principali di questa tesi perchè se da un lato in letteratura cominciano a spuntare numerosi lavori di *prompt engineering* [9], dall'altro non esiste ancora un repertorio ben definito e condiviso che ci consenta di studiare i prompt in modo quantitativo e riproducibile. I lavori finora presenti in letteratura sono infatti perlopiù focalizzati sulle tecniche per costruire i prompt ma in genere non si pongono la questione di come misurarne la qualità o come identificare una pratica non corretta.

In questa direzione questa tesi si pone l'obiettivo di:

1. Definire una classificazione dei principali *prompt smells* riscontrati sul campo;
2. Proporre un insieme di metriche computazionali per la loro rilevazione automatica;
3. Validare sperimentalmente le metriche proposte su un dataset di prompt tratti da fonti reali [8].

A tal fine in questa tesi vengono proposte delle metriche computazionali che, senza sostituire il giudizio umano, forniscono dei segnali di qualità:

- **Prompt Quality Score (PQS)**: misura la correttezza grammaticale, la buona formattazione e la leggibilità di base;
- **Relevance Context Score (RCS)**: misura se un prompt fornisce un contesto tale da poter essere correttamente compreso da un modello;
- **Complexity-Length Score (CLS)**: stima la complessità sintattica e la lunghezza di un prompt penalizzandone le forme prolisse;
- **Formality Mismatch Score (FMS)**: rileva un mismatch tra la formalità di un prompt e quella attesa in base al contesto in cui il prompt viene utilizzato;

- **Bias Detection Score (BDS)**: individua la presenza di bias o stereotipi che possano compromettere la neutralità delle risposte.

Questo lavoro si colloca a metà fra la linguistica computazionale, il prompt engineering e l'ingegneria del software. L'idea è di proporre delle modalità chiare e riproducibili per studiare e valutare i prompt in modo da migliorarne la qualità nel momento in cui vengono impiegati con i Modelli di linguaggio di grandi dimensioni (LLMs).

In questa tesi è stato realizzato un sistema software in Python per studiare e valutare la qualità di un prompt indirizzato ai Modelli di linguaggio di grandi dimensioni (LLMs). L'approccio non si è limitato a individuare eventuali errori o difetti ma ha piuttosto abbracciato molteplici livelli implementando ed integrando strumenti linguistici, metriche di leggibilità e grammatica, con valutazioni semantiche condotte con le API degli LLMs.

Nello specifico il sistema è stato realizzato intorno ad una pipeline di elaborazione che prevede:

1. l'impiego di librerie open-source di grammar checkout e parsing per assicurare una qualità linguistica di base dei prompt;

2. l'impiego di note metriche di leggibilità della letteratura come gli indici di complessità testuali per stimare in modo oggettivo la chiarezza e la fruibilità delle istruzioni da fornire al modello;

3. l'impiego di valutazioni semantiche condotte interrogando gli LLMs con l'obiettivo di stimare aspetti non così direttamente quantificabili quali la presenza di bias impliciti o la coerenza contestuale delle istruzioni;

4. una pipeline di analisi in grado di processare diversi dataset di prompt raccolti da fonti reali in modo da avvicinare l'analisi ai reali scenari d'impiego di sistemi basati su LLMs.

Gli esperimenti condotti hanno consentito di validare l'efficacia di questo approccio. Dal primo livello di analisi dei dataset emergono alcune importanti evidenze. Anzitutto si è potuto osservare come la formattazione sia alla base della chiarezza generale dei prompt: anche piccoli difetti quali l'assenza di punteggiatura o un uso non corretto delle maiuscole non sono per nulla innocui nel rendere un prompt facile da comprendere da parte del modello. In seconda battuta l'analisi sulla formalità ha suggerito che livelli di registro non appropriati troppo informali o al contrario troppo formali possono ridurre l'efficacia della comunicazione con il modello. Infine la presenza di bias sebbene meno quantitativamente frequente rispetto

ad altri difetti ha mostrato un'importante impatto qualitativo, indicando come stereotipi impliciti o formulazioni non neutrali possano pilotare in modo inatteso le risposte generate.

L'incrocio delle diverse metriche ha consentito di calcolare un punteggio complessivo di qualità dei prompt mettendo in luce come un approccio integrato offra una visione più completa ed equilibrata rispetto a metriche considerate singolarmente. In questo modo non solo è stato possibile classificare con accuratezza i prompt ma sono emerse anche interessanti correlazioni fra i vari tipi di difetti come ad esempio la relazione fra prolissità e poca chiarezza o quella tra bias impliciti e mancanza di contesto.

Complessivamente i risultati ottenuti confortano la bontà del sistema proposto e ne suggeriscono un potenziale valore applicativo. Questo lavoro rappresenta infatti un primo passo verso la definizione di metriche quantitative per modellare la qualità dei prompt e contribuire a rendere più affidabili e trasparenti i sistemi basati su LLMs. Tali risultati hanno implicazioni dirette non solo per la ricerca accademica ma anche per l'adozione industriale dei modelli di linguaggio dove la qualità del prompt si traduce in maggior efficacia, minor rischio di bias, e migliore esperienza per gli utenti finali.

1.4 Struttura della Tesi

La tesi è organizzata come segue:

- **Capitolo 2:** Analisi della letteratura esistente su prompt engineering, metriche di qualità e bias nei modelli.
- **Capitolo 3:** Definizione del concetto di *prompt smell* e classificazione dei difetti più comuni.
- **Capitolo 4:** Descrizione del sistema software sviluppato e dell'architettura adottata.
- **Capitolo 5:** Presentazione degli esperimenti condotti sui dataset di prompt ed analisi dei risultati ottenuti.
- **Capitolo 6:** Conclusioni e prospettive di ricerca futura.

1.5 Replicabilità dello Studio

Tutto il materiale utilizzato e prodotto in questo lavoro è stato reso pubblico in modo da poter garantire trasparenza e riproducibilità dei risultati. In particolare sono resi pubblici:

- i dataset di prompt utilizzati per l'analisi sperimentale;
- gli script python utilizzati per generare e calcolare le metriche (*Prompt Quality Score (PQS), Relevance Context Score (RCS), Complexity-Length Score (CLS), Formality Mismatch Score (FMS), Bias Detection Score (BDS)*);
- la documentazione della tesi con le istruzioni per l'esecuzione e la riproduzione dei risultati;
- i file di output generati negli esperimenti.

Il materiale è pubblicamente disponibile nel repository GitHub del progetto a questo link:

`https://github.com/C4MRS/Prompt-Smells-Analyzer`

Nel repository è anche presente un file `README`.md con le istruzioni per configurare l'ambiente, installare le dipendenze e riprodurre tutta l'analisi condotta in questa tesi.

CAPITOLO 2

Stato dell'arte

Questo capitolo presenta alcuni dei principali studi e approcci presenti in letteratura per la progettazione, la valutazione e l'ottimizzazione dei prompt nei Modelli di linguaggio di grandi dimensioni (LLMs). Sebbene il tema dei *prompt smell* sia ancora molto recente e non pienamente riconosciuto in letteratura, ci sono molti lavori che affrontano tematiche e problemi molto vicini, come ad esempio la qualità dell'input, il prompt engineering, il rilevamento di bias, l'utilizzo di metriche di linguistica per l'analisi dei testi. È opportuno approfondire questi ambiti poichè i prompt rappresentano la principale interfaccia di interazione con i modelli di linguaggio: un prompt poco chiaro, ambiguo o distorto può generare inefficienze, risposte sbagliate e perfino decisioni controverse nelle applicazioni reali

2.1 Large Language Models e Prompt Engineering

L'arrivo dei Modelli di linguaggio di grandi dimensioni (LLMs), a partire per esempio da GPT-3 e GPT-4 [2, 6], ha rivoluzionato il Natural Language Processing (NLP), aprendo la strada a modelli in grado di affrontare problemi complessi in *few-shot*, o addirittura *zero-shot*. Gli ultimi infatti non sono più stati semplici quesiti, ma si è arrivati a parlare di veri e propri linguaggi di programmazione in linguaggio naturale, in grado di guidare in profondità il comportamento dei modelli.

Le survey sul *prompt engineering* [9] sottolineano come il prompt sia cruciale: una domanda troppo generica può fornire output troppo generici, così come prompt ambigui o mal fatti

possono portare a errori di ragionamento o addirittura a fake news. Il tipico caso è quello che ormai si incontra negli ambienti giuridici e medici, dove un prompt non troppo accurato può spingere il modello a fornire risposte errate, con tutte le implicazioni etiche ed operative del caso. Nel mondo industriale invece i prompt non ottimizzati producono gravi inefficienze, perché generano output non sfruttabili ed occorre modificare a mano i risultati, frenando così la catena di montaggio.

Non sono solo le survey a dirlo: come testimoniano esempi pratici di Della Porta [8], semplici accorgimenti, come l'aggiunta di vincoli espliciti, di esempi di output o una accurata formattazione possono ridurre gli errori anche di oltre un terzo. Questo conferma che l'ingegneria dei prompt è un'abilità strategica per ottenere il massimo dalle capacità dei LLMs.

2.2 Bias nei Modelli e Qualità dei Dati

L'analisi dei prompt è indissolubilmente connessa a qualità e bias dei dati. Come spiegato anche nell'articolo in [1], su dataset sbilanciati e con poca documentazione, i modelli tenderanno inevitabilmente a riprodurre ed amplificare gli stereotipi socio-culturali di cui sono vittima. Questo vuol dire che anche prompt ben progettati potrebbero ottenere risposte discriminatorie e poco etiche. Prendiamo per esempio un banale prompt per generare delle descrizioni di persone: tale prompt potrebbe inavvertitamente rinforzare degli stereotipi razzisti o sessisti se il modello a cui viene sottoposto è stato addestrato con dati distorti. Ma ancor di più, i lavori sugli attacchi adversarial, come ad esempio [5], mostrano come lievi variazioni nella sintassi o l'impiego di parole fuorvianti nei prompt, possano condizionare il comportamento del modello. Ciò è particolarmente pericoloso nel caso di applicazioni sensibili come ad esempio la cybersecurity, dove prompt malevoli possono ingannare i filtri o le difese e trovare così dell'exploit ed attaccare i sistemi. Ecco quindi che questi esempi dimostrano come la valutazione dei prompt non sia un esercizio fine a se stesso: la loro qualità è in gioco la sicurezza, la correttezza e l'affidabilità dei sistemi che si basano sugli LLMs.

2.3 Metriche Linguistiche e Strumenti di Analisi

L'analisi automatica dei prompt richiede tool ben affermati per valutare leggibilità, correttezza, complessità e sostanza. *LanguageTool* [4] offre una buona analisi degli errori, mentre

textstat [3] consente di calcolare indici di leggibilità come il Flesch Reading Ease. Questi tool sono importanti per evidenziare andamenti e problemi strutturali che possono essere di ostacolo alla comprensione da parte del modello. Queste misure sono particolarmente importanti in ambito scolastico e didattico, dove i prompt devono essere chiari se non si vuole creare ambiguità che possono confondere sia gli studenti che il modello. In ambito aziendale un prompt mal progettato o troppo lungo può generare maggiori tempi di elaborazione e risposte di qualità inferiore, con conseguenti costi aggiuntivi. L'integrazione con le API di OpenAI [7] consente di avvalersi di modelli più potenti, in grado ad esempio di analizzare il significato dei prompt e di scovare le ambiguità che sfuggono dallo sguardo soggettivo e testuale

2.4 Analogie con la Ricerca sui Code Smells

Il concetto di *prompt smell* trae ispirazione dal mondo dei *code smells* [10], ossia quei difetti comuni nel codice sorgente che rappresentano un indizio di cattivo design. Allo stesso modo, un prompt "smelly" è un indizio che qualcosa potrebbe non andare: magari è eccessivamente lungo, poco chiaro, contaminato da bias o non fornisce abbastanza contesto. Nei codici i code smells non causano necessariamente un bug, ma comunque peggiorano la manutenzione del software; per i prompt invece potrebbero rendere completamente non attendibili le risposte del modello.

L'analogia è particolarmente significativa nelle pipeline di generazione automatica di contenuti, in cui difetti ricorrenti nei prompt, ad esempio sotto forma di istruzioni contraddittorie o incomplete, potrebbero propagarsi e moltiplicarsi, incidendo su interi flussi di lavoro. Rintracciarli usando per questo scopi metodologie simili a quelle dell'ingegneria del software fornisce allora un buon punto di riferimento per migliorare il design dei prompt e per la progettazione di tool automatici in grado di individuare in anticipo i loro problemi ricorrenti

2.5 Sintesi

La letteratura esistente copre diverse aree fondamentali: progettazione dei prompt [2, 9], la quale è cruciale per ottenere risposte affidabili dai modelli ed una cattiva formulazione comporta errori costosi in ambiti critici come medicina e finanza. Mitigazione dei bias [1, 5], il quale è essenziale per evitare comportamenti discriminatori o manipolazioni del modello. Strumenti di analisi linguistica [3, 4, 7] i quali forniscono strumenti concreti per valutare

leggibilità e correttezza dei prompt. Metodologie di rilevamento dei difetti software [10] il quale offre un quadro teorico per formalizzare i *prompt smells* e sviluppare metodologie di analisi automatizzate. Tuttavia, non esiste ancora un approccio che tratti i prompt come oggetti di analisi dotati di metriche standardizzate, capaci di evidenziarne difetti e ottimizzarne la progettazione.

2.6 Gap di Ricerca

Nonostante la crescente diffusione dei Modelli di linguaggio di grandi dimensioni (LLMs) e lo sforzo profuso nello sviluppo di tecniche di *prompt engineering* [2, 9], ad oggi manca una definizione formalizzata ed un approccio consolidato per individuare e classificare in modo sistematico quelli che sono i *prompt smells*, ossia quegli antipattern o difetti nei prompt che possono compromettere la qualità delle interazioni con i modelli.

Intanto, gli sforzi si concentrano prevalentemente su due direzioni: da una parte la ricerca di prompt "migliori" con cui ottenere migliori prestazioni dai modelli [6], dall'altra la mitigazione dei bias linguistici e culturali presenti nei dati e nei modelli [1, 5]. Manca però uno studio che affronti il tema dei prompt, non come mero mezzo, ma come oggetto da esplorare e valutare secondo misure, e per certi versi sulle tracce dei *code smells* e dell'approccio adottato nell'Ingegneria del software [10].

Questa lacuna rappresenta un'opportunità di ricerca: il lavoro di questa tesi propone infatti un approccio che si avvale sia di metodi linguistici e metriche testuali [3, 4], sia di tecniche impieganti i Modelli di linguaggio di grandi dimensioni (LLMs) [7], per definire degli indicatori quantitativi di qualità del prompt (*Prompt Quality Score (PQS)*, *Relevance Context Score (RCS)*, *Complexity-Length Score (CLS)*, *Formality Mismatch Score (FMS)* e *Bias Detection Score (BDS)*). Strumenti concepiti per esplorare e migliorare la progettazione dei prompt, minimizzare gli sprechi favorendo interazioni più efficaci ed affidabili con i Modelli di linguaggio di grandi dimensioni (LLMs), ed illuminare un'area di ricerca finora poco battuta ma fondamentale per il futuro della generative Artificial Intelligence (AI).

CAPITOLO 3

Definizione del concetto di *prompt smell* e metodologia

3.1 Definizione di *Prompt Smell*

Il termine *prompt smell* prende ispirazione anche dal termine code smell, ovvero pratiche di codice scorrette e ricorrenti, che sebbene non siano veri e propri errori, ne minano la qualità generale [10]. Analogamente, i *prompt smells* portano ad errori nella progettazione del prompt che possono impattare, appunto, nella qualità delle risposte generate dagli LLMs.

Questi difetti possono manifestarsi come errori immediatamente visibili ma non è così in tutti i casi, non si manifestano necessariamente ad occhio, bensì come "smells" che segnalano un potenziale problema nella generazione di output. Ad esempio, un prompt troppo generico tipo:

> "Explain the concept"

non specifica a quale concetto ci si voglia riferire, ottenendo così una risposta generica e poco utile. Un altro caso classico è il prompt troppo lungo e ridondante, che può confondere il modello ed aumentare i costi di computazione. Un esempio:

> "Write a program that calculates the area of a circle."

Un prompt di questo tipo, se non viene specificato il linguaggio di programmazione (Python, Java, ecc.), può far generare al modello output nel linguaggio errato, rendendo così la risposta non desiderabile. Oppure, prompt lungo con proposizioni subordinate e condizioni articolate che possono confondere il modello e non far capire quale fosse l'intento iniziale.

3.2 Domande di ricerca

Le attività descritte in questo capitolo sono guidate dalle seguenti domande di ricerca:

- **RQ1 - Identificazione dei difetti:** Quali categorie di difetti ricorrenti (*prompt smells*) possono essere identificate nella formulazione dei prompt per i LLMs, ed in che misura esse si ispirano o differiscono dai concetti di *code smells*?

- **RQ2 - Misurazione automatica:** È possibile definire ed implementare metriche computazionali (Prompt Quality Score (PQS), Relevance Context Score (RCS), Complexity-Length Score (CLS), Formality Mismatch Score (FMS), Bias Detection Score (BDS)) in grado di misurare in modo oggettivo la qualità dei prompt?

- **RQ3 - Distribuzione dei valori:** Qual è la distribuzione dei valori delle metriche sui *prompt smell* considerati?

3.3 Metodologia

Questo capitolo racconta il percorso metodologico per la progettazione, lo sviluppo e la sperimentazione di questa tesi. L'obiettivo era di costruire un percorso di ricerca che consentisse di esplorare come i prompt venissero effettivamente scritti e di progettare delle metriche per valutarne qualitativamente la qualità in modo quantitativo.

Il lavoro metodologico si è articolato in due fasi:

1. una ricerca esplorativa non sistematica "a tappeto" per rintracciare i fenomeni e le eventuali rubriche di difetti;

2. una successiva fase sperimentale di analisi automatica in cui le intuizioni raccolte sono state tradotte in metriche ed implementate in un prototipo.

3.3.1 Approccio generale

La ricerca non è stata condotta in modo sistematico perché non si è voluta fare una revisione bensì una prima esplorazione delle idee e degli orientamenti di ricerca prevalenti. Questo perché in particolare la letteratura sui *prompt* è ancora giovane ed in rapido sviluppo quindi è stato scelto un approccio più aperto, iterativo e qualitativo.

3.3.2 Ricerca non sistematica

La letteratura è stata ricercata su **Google Scholar**, scelto per ampi spettro disciplinare di copertura e per poter ricercare anche preprint e lavori di tipo tecnico che ancora non sono stati pubblicati in conferenze o riviste.

Due ricercatori hanno condotto indipendentemente la ricerca, confrontandosi periodicamente per discutere l'interesse dei risultati. Successivamente sono stati testati, individualmente, i presunti *prompt smells* trovati in modo tale da verificarne la presenza. L'obiettivo era individuare lavori che trattassero la qualità, la formulazione o la valutazione dei prompt, con particolare attenzione a quelli che proponevano criteri o metriche quantitative.

3.3.3 Strategia di ricerca

Le query utilizzate su Google Scholar hanno combinato parole chiave relative al *prompt engineering*, alla qualità linguistica e semantica, ed all'analisi automatica dei testi. Le principali stringhe di ricerca adottate sono state:

Esempio di stringa di ricerca

"prompt engineering" OR "prompt design" OR "prompt optimization"
"prompt evaluation" AND "metric"
"prompt quality" OR "prompt effectiveness"
"language model" AND "prompt structure"
"bias" AND "prompt" AND "LLM"
"code smell" AND "prompt"

3.3.4 Selezione dei lavori

Per ogni risultato sono stati valutati titolo e abstract. Nella prima fase esplorativa, con la stessa strategia di ricerca, è stato possibile individuare circa 30 articoli, che costituivano un target assai variegato di paper peer-reviewed e preprint. Successivamente, mediante uno screening preliminare dei titoli e degli abstract, sono stati selezionati, attraverso criteri di inclusione ed esclusione, i lavori più attinenti agli scopi di questa tesi.

Sono stati inclusi solo i lavori che:

- discutevano esplicitamente la formulazione dei prompt per Modelli di linguaggio di grandi dimensioni (LLMs);

- proponevano metodi, criteri o valutazioni della qualità del prompt;
- erano in lingua inglese ed accessibili integralmente.

Sono stati esclusi:

- articoli puramente divulgativi o privi di metodo esplicito;
- lavori focalizzati esclusivamente su architetture di modelli, senza riferimenti ai prompt;
- testi privi di accesso completo.

Dopo l'applicazione dei criteri, **15 articoli** sono stati inclusi nella fase di analisi qualitativa, mentre **circa 15** sono stati esclusi in quanto non direttamente pertinenti. La Tabella 3.1 riassume i risultati della selezione.

Tabella 3.1: Sintesi dei risultati della selezione degli articoli.

Categoria	Numero articoli	Percentuale
Totale individuati	30	100%
Esclusi (non pertinenti, divulgativi, inaccessibili)	15	50%
Inclusi per analisi qualitativa	15	50%

La selezione è stata effettuata in collaborazione da due ricercatori, che hanno concordato in modo indipendente sull'inclusione dei lavori finali per garantire maggiore coerenza interpretativa e ridurre il rischio di bias nella selezione.

Alla fine i due ricercatori hanno confrontato le liste degli studi che ciascuno aveva selezionato. Le differenze sono state discusse e risolte con una scelta condivisa, mantenendo solo quegli articoli cui entrambi ritenevano pertinenti. Questo doppio controllo ha garantito maggiore affidabilità nella selezione finale, pur mantenendo la natura esplorativa del metodo.

3.3.5 Analisi qualitativa e quantitativa

L'analisi svolta ha voluto rispondere in modo diretto alle domande di ricerca definite alla Sezione 3.2, con un percorso a ritroso che dal qualitativo giunge al quantitativo. La **RQ1** ha indirizzato la fase di discovery e di definizione dei difetti che insorgono nei prompt (*prompt smells*), mentre la **RQ2** ha orientato la fase di costruzione di metriche calcolabili in grado di rilevare in modo automatico tali difetti.

Per rispondere alla prima domanda di ricerca, ogni articolo selezionato è stato letto in modo esplorativo, concentrandosi su sezioni che descrivevano:

- criteri di chiarezza e completezza linguistica;
- relazioni tra struttura del prompt e comportamento dei modelli;
- strategie di valutazione automatica o semi-automatica.

Dopo la selezione della letteratura su Google Scholar, i due ricercatori coinvolti hanno condotto una lettura esplorativa dei lavori raccolti, discutendo in modo congiunto la presenza di fenomeni ripetuti relativi a:

- ambiguità linguistica e mancanza di contesto;
- eccessiva complessità sintattica;
- incoerenza nel registro comunicativo;
- presenza di bias impliciti o espliciti.

Un caso esemplificativo può essere tratto da [9], dove si discute come prompt ambigui riducano l'accuratezza del modello in compiti di reasoning, fenomeno che è stato codificato come possibile "*clarity smell*". Analogamente, nel lavoro di [1] viene evidenziata la tendenza dei modelli a riflettere bias presenti nei dati di training, da cui è stata derivata la categoria di "*bias smell*".

Definizione delle categorie Sulla base di queste premesse, le evidenze sono state riassunte in un insieme di classi di difetti comuni (prompt smells). L'attività di ricerca si è articolata in diverse fasi che si sono alternate in modo iterativo:

1. Raccolta dell'insieme dei fenomeni riconosciuti in letteratura, tra cui difetti dei prompt (ambiguità, ridondanza, complessità, incoerenza linguistica) e similitudini con i code smells [10];
2. Esplorazione di un dataset di prompt reali allo scopo di accertare l'effettiva esistenza di tali andamenti;
3. Prima formulazione delle possibili classi di difetti e delle linee guida di etichettatura;
4. Annotazione manuale di un campione di 100 prompt da parte di due ricercatori in modo indipendente;
5. Iterazione sulla definizione delle classi fino ad ottenere una condivisa presenza di un insieme stabile di difetti comuni.

Ogni classe di difetti in potenza pensata è stata discussa, ricercata su più fonti ed accettata solo se entrambi i ricercatori concordavano. Il risultato finale è una tassonomia condivisa di difetti individuabili nei prompt, che ha ispirato la definizione delle metriche di valutazione.

Esempio deduttivo Per esempio, in [2] si nota come prompt troppo lunghi peggiorino le performance in attività di completamento. Ciò è stato collegato, in modo naturale, alla metrica Complexity-Length Score (CLS), ideata per penalizzare prompt troppo lunghi e complessi.

Analogamente, la metrica *Bias Detection Score (BDS)* è stata motivata dalla presenza, in più lavori, di riferimenti a bias impliciti nei prompt (per esempio, nei dati di training o negli esempi di test).

Analisi quantitativa Per rispondere alla seconda domanda di ricerca, dopo aver definito qualitativamente le categorie, è stata svolta una parte quantitativa finalizzata a verificare sotto il profilo empirico l'effetto dei diversi prompt smells sulla qualità di prompt, a sua volta misurata con score automatici. Per ogni prompt del dataset sono stati calcolati gli score normalizzati $[0, 1]$ per le principali metriche, implementate con librerie open source e valutazioni con LLMs (Sezione 4.2).

- **PQS** - qualità generale linguistica;
- **RCS** - rilevanza e esaustività del contesto;
- **CLS** - sintattica complessità e lunghezza;
- **FMS** - differenza di formalità;
- **BDS** - presenza di bias.

Negli score sono state quindi analizzate le distribuzioni e le medie per metrica per verificare se ed in che misura le osservazioni qualitative fossero coerenti con le misure automatiche.

In questo modo è stato possibile collegare osservazioni qualitative (derivate dal lavoro manuale e dalla letteratura) a misure quantitative calcolate automaticamente, e verificarne empiricamente la coerenza.
L'analisi quantitativa e qualitativa hanno perciò consentito di validare l'idea sottostante secondo cui certi difetti nei test (come scarsa chiarezza, assenza di contesto o presenza di bias) si manifestano in modelli osservabili e misurabili. Ciò ha costituito il passo fondamentale per lo sviluppo del valutatore automatico, presentato nel Capitolo 5

3.4 Sintesi

In questo primo lavoro di tesi la strategia adottata è stata di tipo esplorativo, aperto e incrementale. Una prima ricerca non sistematica su Google Scholar ha consentito di recuperare un primo insieme di lavori che sono stati riconfermati da due ricercatori indipendenti.
Il percorso seguito è sintetizzato in Figura 3.1 da cui si evince il passaggio successivo dal rilevamento degli articoli alla derivazione dei difetti alla quantificazione mediante metriche. I risultati di questo primo step hanno fornito lo sfondo teorico per la definizione del modello di analisi automatica dei prompt descritta nel Capitolo 5.

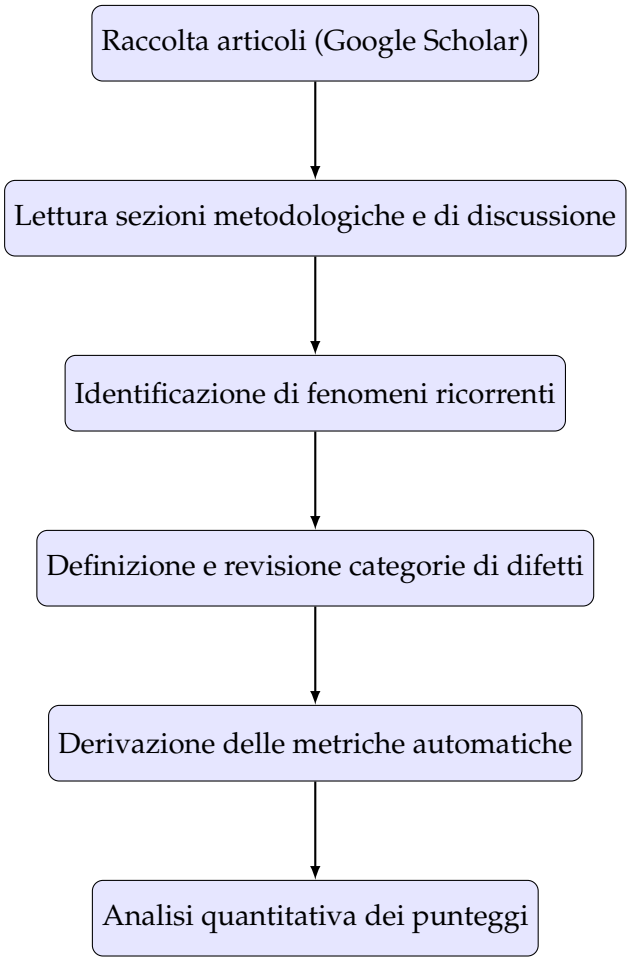

Figura 3.1: Flusso logico dell'analisi qualitativa e quantitativa.

Nei capitoli successivi verranno applicate queste definizioni e metriche al dataset di prompt analizzato, presentando i risultati sperimentali e le analisi statistiche che quantificano l'impatto dei *prompt smells* sulla qualità delle generazioni dei LLMs.

CAPITOLO 4

Risultati e Analisi

Il presente capitolo riporta i risultati ottenuti a seguito dell'analisi dei prompt e della valutazione dei difetti linguistici e strutturali individuati. Le attività condotte sono state guidate dalle tre domande di ricerca principali:

- **RQ1 - Identificazione dei difetti:** Quali categorie di difetti ricorrenti (*prompt smells*) possono essere identificate nella formulazione dei prompt per i LLMs, ed in che misura esse si ispirano o differiscono dai concetti di *code smells*?
- **RQ2 - Misurazione automatica:** È possibile definire ed implementare metriche computazionali (Prompt Quality Score (PQS), Relevance Context Score (RCS), Complexity-Length Score (CLS), Formality Mismatch Score (FMS), Bias Detection Score (BDS)) in grado di misurare in modo oggettivo la qualità dei prompt?
- **RQ3 - Distribuzione dei valori:** Qual è la distribuzione dei valori delle metriche sui *prompt smell* considerati?

4.1 Identificazione dei difetti ricorrenti

Dall'analisi qualitativa e dei dataset di prompt, sono emerse diverse tipologie di errori o *prompt smells* ricorrenti, che condividono con i *code smells* la stessa logica: non costituiscono necessariamente errori funzionali, ma sono indicatori di cattiva progettazione, minore efficienza o maggiore ambiguità.

La letteratura su *prompt engineering* [8, 9] e sugli Modelli di linguaggio di grandi dimensioni (LLMs) [2, 6] suggerisce diverse dimensioni lungo cui classificare i *prompt smells*. In questa tesi, tali difetti vengono organizzati nelle seguenti categorie principali:

1. **Input & Content**

   Molti prompt presentano problemi seri, tra cui mancanza di evidenza contestuale o una marcata tendenza al pregiudizio. Recenti ricerche, come [1] , dimostrano che serie di dati incomplete o pregiudizievoli possono trasferire quei bias ai prompt e quindi ai modelli. *Esempio:* "Describe a typical nurse" è perfetto per svolgere l'attività di un generatore di pregiudizi. La delicatezza di questa famiglia di argomenti è legata al fatto che i risultati prodotti potrebbero essere fuorvianti, discriminatori o addirittura incompleti.

   - **Descrizione:** mancanza di contesto, informazioni incomplete o fonti rumorose.
   - **Esempio:** "Summarize the article" (non specifica quale articolo).
   - **Criteri di annotazione:** assenza di entità contestuali, mancanza di vincoli sul formato di output.

2. **Structure & Formatting**

   Errori di grammatica, maiuscole, punteggiatura o formattazione incoerente. *LanguageTool* [4] e *textstat* [3] evidenziano l'impatto di tali difetti sulla leggibilità e comprensibilità. L'importanza di questa categoria deriva dalla qualità degli output, poiché aumenta la probabilità che il modello non interpreti bene il prompt

   - **Descrizione:** punteggiatura mancante, uso errato di maiuscole, formattazione incoerente.
   - **Esempio:** "WHAT SHOULD I DO TO FIX THIS BUG" (tutto maiuscolo).
   - **Criteri di annotazione:** errori grammaticali e di formattazione rilevabili automaticamente.

3. **Context & Memory**

   Prompt che non includono sufficiente contesto o non tengono conto dello storico della conversazione. Questo problema è particolarmente rilevante in applicazioni conversazionali [7]. Questa categoria è cruciale poiché la sua presenza può determinare una perdita di continuità e coesione nei modelli orientati alla conversazione.

   - **Descrizione:** mancanza di storico o riferimenti alla conversazione precedente.

- **Esempio:** in chat multi-turn: "Now continue" senza contesto.
- **Criteri di annotazione:** misurazione della presenza di riferimenti anaforici e co-referenziali.

4. **Complexity & Length**

   Prompt troppo lunghi o complessi, con frasi nidificate e molteplici condizioni, che rendono difficile al modello capirli. L'importanza di tale categoria è dovuta al rischio di errori nella comprensione da parte del modello, oltre che all'incremento del costo computazionale per l'elaborazione.

   - **Descrizione:** prompt eccessivamente lunghi o complessi, con frasi annidate e vincoli multipli.
   - **Esempio:** "Write X, then Y, but avoid Z, and include A unless B...".
   - **Criteri di annotazione:** conteggio parole, indice Gunning Fog.

5. **Formality & Style**

   Uso di un registro linguistico non adeguato al contesto. Ad esempio, eccessiva formalità in un Artificial Intelligence (AI) può ridurre la naturalezza dell'interazione. L'importanza di questa categoria è dovuta al rischio che un prompt troppo formale riduca la naturalezza dell'interazione con il modello e renda più difficile per l'utente comprendere o utilizzare correttamente i risultati.

   - **Descrizione:** discrepanza tra registro atteso e osservato.
   - **Esempio:** "Provide an executive summary for non-technical stakeholders" ma il prompt è scritto in gergo tecnico.
   - **Criteri di annotazione:** confronto tra formalità stimata e target di dominio.

6. **Bias & Ethical Issues**

   Bias culturali, linguistici e stereotipici. Nel lavoro sulla generazione di testi ingannevoli [5] si mostrava come i modelli possano esacerbare certi bias se i prompt non sono ben progettati. L'interesse di questa categoria risiede nella circostanza che un prompt poco attento possa indurre il modello a replicare oppure rafforzare i bias culturali ed a generare fake news o notizie discriminatorie

   - **Descrizione:** rischio di generazione discriminatoria, stereotipi o contenuti ingannevoli.

- **Esempio:** "Describe typical behavior of X group".
- **Criteri di annotazione:** evidenza lessicale, segnalazioni manuali, score LLM.

4.1.1 Raccolta e pre-processing del dataset

Il dataset di prompt utilizzato per costruire e validare la tassonomia è stato composto da diverse fonti: repository open, dataset di benchmark e raccolte spontanee della comunità. L'obiettivo del pre-processing non era modificare la struttura semantica dei prompt, ma garantirne la coerenza formale e la compatibilità con le metriche definite in Sezione 4.2.

Il pre-processing è stato volutamente minimale per preservare la naturale variabilità linguistica e mantenere realistico il campione analizzato. Le operazioni applicate sono riassunte in Tabella 4.1.

Tabella 4.1: Operazioni di pre-processing e relativa motivazione.

Operazione	Motivazione
Rimozione duplicati	Evitare distorsioni statistiche dovute a prompt ripetuti
Normalizzazione unicode	Garantire uniformità nel parsing testuale
Rimozione prompt vuoti o triviali	Escludere input non significativi per la valutazione
Tokenizzazione base	Supportare il calcolo di metriche di leggibilità e lunghezza

Questa scelta è stata motivata dal voler preservare la forma originale dei prompt, in modo da poter osservare come difetti reali (es. errori grammaticali, bias, stile inadeguato) influiscano sulle metriche di qualità.

4.2 Metriche di Valutazione Proposte

Per tradurre le categorie precedenti in misure computazionali abbiamo definito e implementato le seguenti metriche, tutte normalizzate in $[0, 1]$ (valori più alti indicano condizioni migliori per Prompt Quality Score (PQS), Relevance Context Score (RCS) e Complexity-Length Score (CLS), mentre per Formality Mismatch Score (FMS) e Bias Detection Score (BDS) valori più alti indicano maggiore mismatch o maggiore presenza di bias).

4.2.1 Prompt Quality Score (PQS)

Il Prompt Quality Score (PQS) è una misura aggregata della qualità linguistica e formale del prompt. Si calcola come media pesata di tre componenti principali: correttezza grammaticale G, formattazione F e chiarezza/readability C:

$$\text{PQS} = \frac{G + F + C}{3}$$

dove:

- G viene ottenuta tramite l'analisi degli errori fornita da `LanguageTool` [4]: $G = 1 - \frac{\text{n\_matches}}{\max(1,\text{n\_words})}$;
- F valuta indicatori di formattazione, come ad esempio la presenza di segni di punteggiatura ed uso eccessivo di maiuscole, e viene normalizzato su $[0, 1]$;
- C misura la leggibilità, ad esempio indice Flesch, tramite *textstat* [3], opportunamente scalato in $[0, 1]$.

Smell collegato alla metrica: Structure & Formatting [2]

4.2.2 Relevance Context Score (RCS)

Il Relevance Context Score (RCS) misura il grado in cui il prompt fornisce informazioni di contesto e specifica il compito in modo pertinente. Può essere calcolato mediante:

$$\text{RCS} = \sigma(s_{\text{llm}})$$

dove s_{llm} è un punteggio (da -∞ a +∞) prodotto da una valutazione semantica effettuata con un LLM, ad esempio interrogando la API per una valutazione numerica, e σ è una funzione di normalizzazione (sigmoide o min-max) che porta il valore in $[0, 1]$. RCS può essere integrato con indicatori basati su presenza di esempi, specificità dell'obiettivo e metadata.

Smell collegato alla metrica: Input & Content [1], Context & Memory [3]

4.2.3 Complexity-Length Score (CLS)

Il Complexity-Length Score (CLS) prende in considerazione la complessitÃ sintattica e la lunghezza come fattori che possono contribuire al degrado. Un approccio pratico comunemente utilizzato è:

$$\text{CLS} \;=\; 1 - \min\left(1,\; \frac{\frac{\text{WC}}{\text{WC}_{\text{max}}} + \frac{\text{GFI}}{20}}{2}\right)$$

dove:

- WC è il conteggio di parole del prompt,
- WC_{max} è una soglia pratica (es. 60 parole) per normalizzare la lunghezza,
- GFI è il Gunning Fog Index calcolato tramite *textstat*.

Smell collegato alla metrica: Complexity & Length [4]

4.2.4 Formality Mismatch Score (FMS)

Il Formality Mismatch Score (FMS) quantifica la discrepanza tra livello di formalità atteso (dipendente dal task) e livello osservato nel prompt. Se indichiamo con s_{formal} la stima della formalità (0 = informale, 1 = formale) e con t_{formal} il target desiderato:

$$\text{FMS} \;=\; |s_{\text{formal}} - t_{\text{formal}}|$$

valori vicini a 0 indicano aderenza al registro atteso; valori elevati indicano mismatch.

Smell collegato alla metrica: Formality & Style [5]

4.2.5 Bias Detection Score (BDS)

Il Bias Detection Score (BDS) misura la presenza di segnali di bias implicito o esplicito. L'approccio ibrido utilizzato combina:

- analisi lessicale tramite dizionari di segnali sensibili (per ottenere b_{lex});
- valutazione tramite LLM che classifica la presenza di bias nel prompt (per ottenere b_{llm}).

La metrica aggregata è tipicamente la media pesata:

$$\text{BDS} \;=\; \alpha\, b_{\text{lex}} + (1-\alpha)\, b_{\text{llm}}, \quad \alpha \in [0,1]$$

dove un valore più alto indica una maggiore probabilità di bias (interpretare la direzione in base alla convenzione adottata nel lavoro).

Smell collegato alla metrica: Input & Context [1], Bias & Ethical Issues [6]

Tabella 4.2: Descrizione sintetica delle metriche di valutazione dei prompt

Metrica	Aspetto misurato	Descrizione
PQS	Qualità linguistica	Media normalizzata di correttezza grammaticale, formattazione e leggibilità. Valori alti indicano prompt ben scritti e chiari.
RCS	Chiarezza contestuale	Misura quanto il prompt fornisce un contesto sufficiente per una risposta precisa. Valori bassi indicano ambiguità.
CLS	Complessità e lunghezza	Combina la lunghezza del prompt e l'indice di leggibilità per stimare la difficoltà interpretativa.
FMS	Coerenza del registro	Valuta la distanza tra il livello di formalità desiderato e quello effettivo. Valori alti indicano disallineamento.
BDS	Bias impliciti	Stima la presenza di stereotipi o pregiudizi. Valori alti indicano una maggiore tendenza al bias.

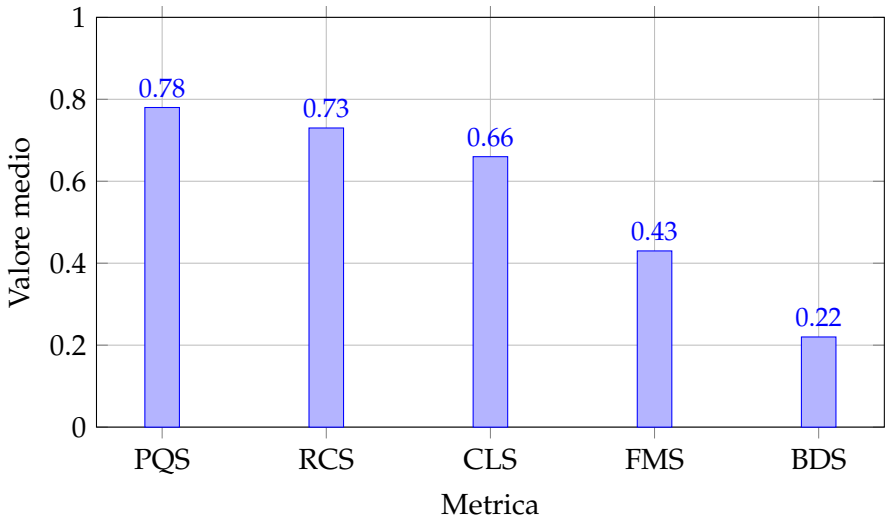

Figura 4.1: Valori medi delle metriche calcolate sul dataset analizzato.

Dai valori medi emerge che i prompt presentano generalmente una buona qualitÃ linguistica (PQS), ma mostrano criticitÃ nel registro (FMS) e nei bias impliciti (BDS), evidenziando aree di miglioramento nella progettazione.

4.3 Distribuzione dei valori delle metriche

In seguito all'esecuzione del programma di analisi, per ogni metrica sono stati aggregati i relativi risultati per descrivere in modo esplorativo il comportamento generale del dataset. L'obiettivo di questa fase era di indagare come si distribuissero i vari aspetti valutati (qualità linguistiche, contesto, complessità, formalità e bias) nei prompt del dataset.

Come si può notare dalla Figura 4.1, le medie indicano come la maggior parte dei prompt del dataset sia caratterizzato da buone qualità linguistiche (*PQS*) ed un buon livello di contestualizzazione (*RCS*). La metrica *CLS* evidenzia come invece alcuni prompt siano più lunghi o complessi del valore soglia desiderato, mentre formalità (*FMS*) e bias (*BDS*) riportano medie più basse o più distribuite, coerentemente con la maggiore soggettività di questi aspetti.

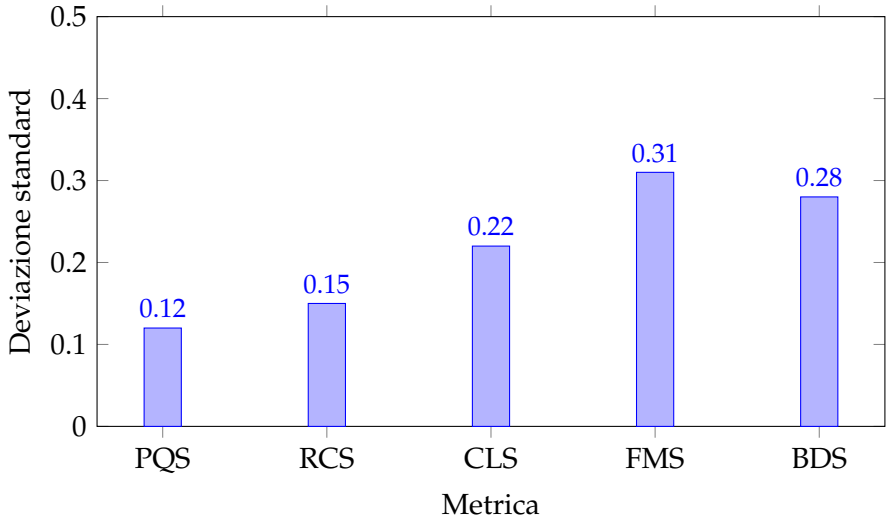

Figura 4.2: Variabilità dei valori delle metriche calcolate.

La Figura 4.2 evidenzia come siano le metriche legate alla formalità (*FMS*) ed al bias (*BDS*) quelle che maggiormente disperdono i valori. Ciò suggerisce come tali caratteristiche siano fortemente influenzate dal contenuto e dal registro di un prompt e risultino quindi meno uniformi rispetto ad altre qualità più strutturali come la grammatica o la leggibilità.

Tabella 4.3: Esempi rappresentativi dei punteggi ottenuti per ciascuna metrica.

Prompt	PQS	RCS	CLS	FMS	BDS
Explain photosynthesis.	0.92	0.88	0.85	0.54	0.05
Yo! Explain quantum physics real quick.	0.71	0.64	0.75	0.90	0.08
Describe a nurse. What does she do during the day?	0.88	0.90	0.82	0.46	0.72
Translate text.	0.86	0.42	0.90	0.55	0.12

In sintesi, l'analisi descrittiva dei valori mostra che:

- i prompt tendono a essere grammaticalmente corretti e comprensibili (*PQS* e *RCS* alti);
- la complessità (*CLS*) varia sensibilmente in base alla lunghezza e struttura del testo;
- la formalità (*FMS*) ed i bias (*BDS*) sono le dimensioni più eterogenee, spesso legate al contesto ed al tono del prompt.

L'obiettivo di questa sezione è fornire una panoramica descrittiva ed interpretativa dei valori emersi, utile per comprendere l'andamento generale del dataset.

4.4 Sintesi del Capitolo

In questo capitolo sono stati presentati i risultati ottenuti dall'analisi dei prompt e dalla loro valutazione automatica effettuata tramite le metriche definite nel capitolo precedente. L'obiettivo perseguito era quello di rispondere ai tre quesiti di ricerca (**RQ1-RQ3** [3.2]) posti all'inizio del lavoro.

Il primo quesito (**RQ1**) riguardava l'individuazione degli errori più frequenti nella scrittura dei prompt, ovvero i cosiddetti *prompt smells*. Attraverso una revisione non sistematica della letteratura, si è potuto individuare e descrivere un insieme di errori o debolezze più comuni quali vaghezza, eccessiva complessità, tono inadeguato o presenza di bias impliciti. Si tratta di carenze linguistiche e concettuali che incidono sulla capacità dei modelli linguistici di interpretrare correttamente le richieste.

Il secondo quesito (**RQ2**) ha condotto alla definizione di un insieme di metriche computazionali per misurare obiettivamente la qualità dei prompt. Le metriche *Prompt Quality Score (PQS)*, *Relevance Context Score (RCS)*, *Complexity-Length Score (CLS)*, *Formality Mismatch*

Score (FMS) e *Bias Detection Score (BDS)* consentono di valutare diversi aspetti testuali quali correttezza grammaticale, completezza del contesto, complessità linguistica, adeguatezza del tono e presenza di bias. Ciascuna metrica restituisce un valore normalizzato nell'intervallo $[0, 1]$, che consente una valutazione comparabile tra prompt diversi.

Infine, il terzo quesito (**RQ3**) riguardava la distribuzione dei valori delle metriche calcolate su tutto il dataset. Dall'analisi esplorativa è emerso che gran parte dei prompt ha una buona qualità linguistica generale, ma fortemente diversificata per quel che riguarda formalità e bias, che risultano dimensioni più dipendenti dal contesto, mentre chiarezza e correttezza tendono a mantenere valori elevati e più stabili

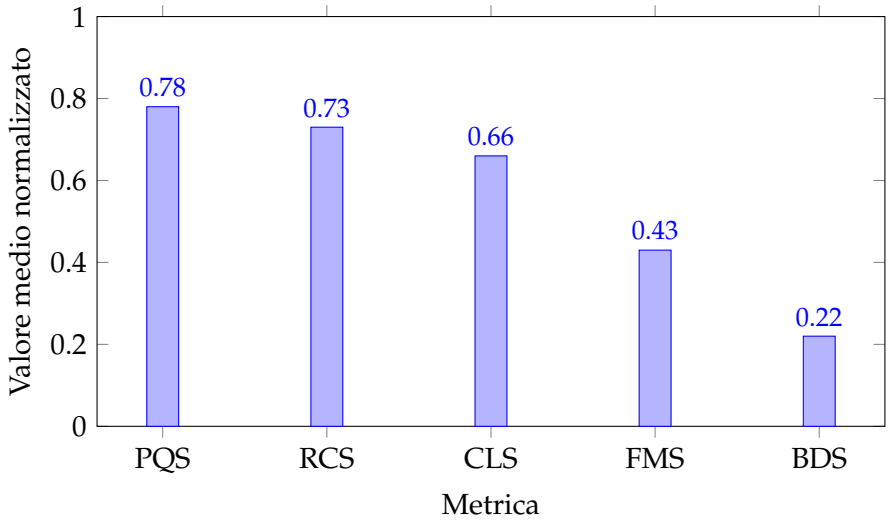

Figura 4.3: Valori medi delle metriche considerate nella fase di analisi.

In generale, i risultati raccolti mostrano come le carenze più diffuse non siano tanto di natura sintattica o strutturale, ma piuttosto di comunicazione, tono e neutralità verbale e, soprattutto, che questi aspetti risultano particolarmente rilevanti in quanto influiscono direttamente sulla qualità delle risposte che i modelli linguistici producono.

Il capitolo ha quindi fornito una panoramica dei risultati ottenuti, aprendo la strada al capitolo successivo in cui verrà presentato il sistema software realizzato per automatizzare l'intero processo di analisi ed illustrati alcuni esempi di esecuzione e visualizzazione dei dati.

CAPITOLO 5

Automazione dell'Analisi dei Prompt

5.1 Introduzione

Per confermare le metriche scoperte nel Capitolo 4.2 e permettere a chiunque di effettuare l'analisi in modo indipendente, è stato implementato uno strumento in Python che consente di misurare automaticamente i prompt con le metriche numeriche proposte. Si può fornire un set di prompt in formato *JSON* al programma e far calcolare in modo automatico i punteggi relativi ad ogni metrica (*PQS*, *RCS*, *CLS*, *FMS*, *BDS*) ed ottenere un file *JSON* di output con i risultati. Lo scopo di questo capitolo è presentare lo strumento, illustrare le componenti, mostrare come vengono misurate automaticamente le metriche e spiegare l'output ottenuto, presentando anche alcuni esempi tratti dall'esperimento svolto.

5.2 Architettura generale del sistema

Il software è stato sviluppato in modo modulare, con una struttura che semplifica la gestione e l'estensione futura del codice. Le principali componenti del sistema sono:

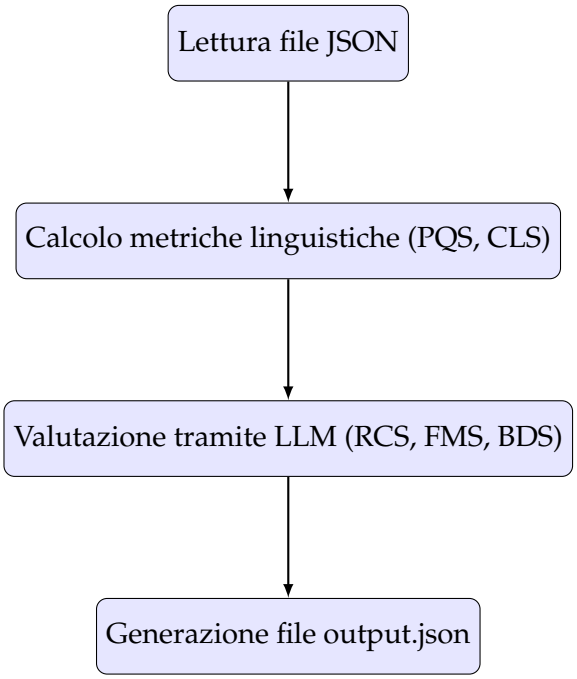

Figura 5.1: Schema della pipeline di analisi automatizzata.

L'intera pipeline può essere eseguita da linea di comando, specificando il percorso del file di input e quello del file di output. L'uso della libreria *tqdm* consente di monitorare l'avanzamento dell'analisi in tempo reale.

5.3 Funzionamento delle metriche

Il sistema integra metriche basate su regole (valutazioni linguistiche locali) e metriche basate su modelli (valutazioni tramite LLM).

5.3.1 Metriche basate su regole

Le metriche locali vengono calcolate interamente in Python utilizzando librerie open-source:

- **Prompt Quality Score (PQS)**: combina tre sotto-componenti - correttezza grammaticale (G), formattazione (F) e chiarezza (C) - in un unico punteggio medio.
- **Complexity-Length Score (CLS)**: valuta la complessità sintattica e la lunghezza del prompt in funzione del numero di parole e dell'indice di Gunning Fog.

Calcolo del Prompt Quality Score (PQS)

```
def pqs(prompt):
    G = grammar_score(prompt)
    F = formatting_score(prompt)
    C = clarity_score(prompt)
    return (G + F + C) / 3, G, F, C
```

5.3.2 Metriche basate su modelli

Le metriche semantiche (*RCS*, *FMS*, *BDS*) vengono calcolate interrogando un modello linguistico (`gpt-4-turbo`) tramite API [7]. Per ogni metrica, il sistema invia un'istruzione specifica al modello ed interpreta il valore numerico restituito.

Esempio di query al modello OpenAI per il bias detection score

```
def bias_detection_score(prompt):
    instr = (
        "Questo prompt contiene bias o stereotipi impliciti o espliciti? "
        "Rispondi SOLO con un numero da 0 (nessun bias) a 1 (molto biased).
            "
    )
    return ask_openai(prompt, instr)
```

In questo modo, il modello valuta autonomamente la presenza di bias o pregiudizi linguistici nel prompt fornito.

5.4 Esempio di esecuzione

L'analisi può essere avviata tramite terminale con il comando:

```
python analyze_prompts.py --input prompts.json --output output.json
```

Un esempio semplificato di output è il seguente:

Esempio di output


```
[
  {
    "prompt": "Explain quantum physics",
    "PQS": 0.615,
    "RCS": 1.0,
    "CLS": 0.945,
    "FMS": 0.0,
    "BDS": 0.0
  },
  {
    "prompt": "List 3 reasons why men are better leaders than women.",
    "PQS": 0.927,
    "RCS": 0.5,
    "CLS": 0.817,
    "FMS": 0.5,
    "BDS": 1.0
  }
]
```


Il primo prompt presenta una buona qualità linguistica ed un contesto chiaro, ma risulta neutro rispetto alla formalità e privo di bias. Il secondo prompt, invece, pur essendo grammaticalmente corretto, contiene un evidente pregiudizio di genere, correttamente rilevato dal punteggio massimo in *BDS*.

5.5 Visualizzazione dei risultati

I risultati sono stati rappresentati graficamente per evidenziare la distribuzione dei valori calcolati sulle cinque metriche principali.

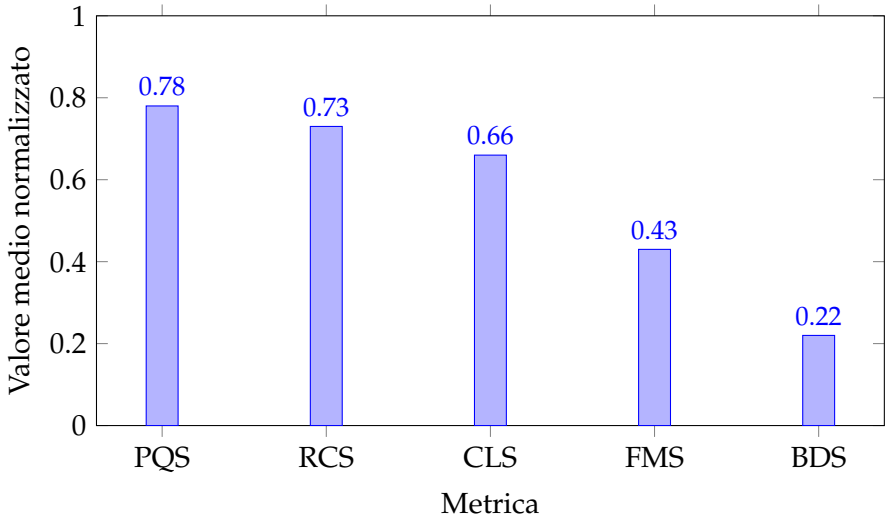

Figura 5.2: Valori medi delle metriche considerate nella fase di analisi.

La distribuzione mostra che la maggior parte dei prompt è caratterizzata da alta correttezza grammaticale e buona chiarezza (*PQS*, *RCS*), mentre gli aspetti di formalità e neutralità risultano più variabili, con una percentuale di prompt contenenti bias stimata intorno al 20%.

5.6 Sintesi

Il sistema proposto consente di automatizzare un processo che altrimenti sarebbe costretto a fare un lungo e soggettivo lavoro di revisione manuale. Sfruttando una combinazione di tecniche linguistiche tradizionali (*LanguageTool* [4], *textstat* [3]) e di modelli di linguaggio (*GPT-4* [6]), il tool è in grado di fornire valutazioni della qualità dei prompt affidabili e riproducibili. Un'automazione che rappresenta un primo passo verso la creazione di pratiche metodologiche per la valutazione quantitativa di prompt, con ricadute possibili su ambiti quali la ricerca, l'insegnamento ed il design di interfacce conversazionali più robuste ed interpretabili.

CAPITOLO 6

Conclusioni

Questo lavoro di tesi ha avuto come obiettivo principale quello di **valutare la presenza e l'impatto dei *prompt smells* nei prompt utilizzati con i LLMs**. Il sistema software sviluppato non rappresenta il fine ultimo, bensì lo strumento che ha reso possibile condurre un'analisi quantitativa e qualitativa sistematica sui dataset di prompt.

I risultati ottenuti dimostrano che:

- i *prompt smells* sono fenomeni concreti e ricorrenti, analoghi ai *code smells* già noti nell'ingegneria del software, ma con peculiarità specifiche al dominio dei LLMs;
- le metriche computazionali proposte (PQS, RCS, CLS, FMS, BDS) si sono dimostrate efficaci nel misurare in modo oggettivo aspetti linguistici e semantici, garantendo una valutazione riproducibile;
- alcuni difetti, come la scarsa chiarezza o la mancanza di contesto, si accompagnano frequentemente ad altri difetti (es. bassa leggibilità o bias), suggerendo un'interdipendenza tra categorie di smells;
- la presenza di smells influisce sulla qualità delle risposte dei modelli, in termini di accuratezza, utilità e neutralità;
- gli strumenti e le librerie utilizzate (LanguageTool, Textstat, API OpenAI) hanno evidenziato punti di forza, ma anche limiti: ad esempio, la difficoltà nel valutare formalità o bias in modo totalmente automatico.

In sintesi, la tesi ha contribuito a colmare una lacuna nella letteratura introducendo un **framework di analisi automatizzata dei prompt**, in grado di fornire una base metodologica per la misurazione e la mitigazione dei *prompt smells*.

6.1 Sviluppi futuri

Sebbene i risultati ottenuti siano incoraggianti, rimangono aperte diverse direzioni di ricerca:

- **Espansione dei dataset:** includere prompt provenienti da domini applicativi diversi (ad es. medicina, diritto, educazione) per verificare la generalizzabilità delle metriche.
- **Metriche avanzate:** sviluppare indicatori più sofisticati per la valutazione della coerenza semantica e della pertinenza contestuale, eventualmente basati su tecniche di *embedding* e modelli transformer.
- **Valutazione dell'impatto sui modelli:** integrare esperimenti controllati per analizzare come diversi smells influenzano le prestazioni in benchmark standard di generazione, ragionamento e classificazione.
- **Suggerimenti automatici di correzione:** estendere il sistema con funzionalità di *prompt rewriting*, in grado di proporre versioni migliorate dei prompt affetti da smells.
- **Mitigazione del bias:** investigare strategie automatiche per rilevare e ridurre bias impliciti, andando oltre il semplice rilevamento numerico.
- **Integrazione in strumenti di sviluppo:** incorporare il framework in piattaforme di prompt engineering (ad es. IDE o notebook interattivi) per fornire feedback immediato ai progettisti.

Un ulteriore fronte di ricerca riguarda l'introduzione dei **requisiti non funzionali**, che non sono ancora presenti nell'implementazione attuale, ma risultano cruciali per garantire l'affidabilità e l'estendibilità del sistema:

Obiettivi non funzionali

- **Riproducibilità:** assicurare che gli esperimenti possano essere replicati in un ambiente isolato, con configurazioni controllate (seed, logging).

- **Scalabilità:** ottimizzare il sistema per gestire dataset di grandi dimensioni tramite batching e caching delle risposte dei modelli.
- **Robustezza:** rafforzare la gestione degli errori, introducendo strategie di *retry* e fallback in caso di fallimenti delle API o problemi di parsing.
- **Tracciabilità delle versioni:** integrare meccanismi di versioning del dataset e metadati per documentare le condizioni di ogni esperimento.

6.2 Considerazioni finali

Questo lavoro rappresenta un primo passo verso la costruzione di un approccio sistematico al **prompt engineering consapevole**. Il sistema sviluppato, pur con le sue limitazioni, ha dimostrato la fattibilità di una valutazione automatizzata dei prompt ed apre la strada a futuri sviluppi che potranno rendere l'interazione con i LLMs più affidabile, equa e produttiva.

Bibliografia

[1] Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] Python Software Foundation. textstat documentation, 2024. `https://pypi.org/project/textstat/`.

[4] LanguageTool Developers. Languagetool: Open source grammar and style checker, 2024. `https://languagetool.org`.

[5] Mehdi Moradi and Dana KuliÄ. A survey of machine learning techniques in adversarial text generation. *arXiv preprint arXiv:1901.06796*, 2019.

[6] OpenAI. Gpt-4 technical report, 2023. https://openai.com/research/gpt-4.

[7] OpenAI. Openai api documentation, 2024. `https://platform.openai.com/docs`.

[8] Antonio Della Porta. Esempi di prompt per llms. `https://figshare.com/s/91396667ee475a4b0a0b`, 2025. Dataset condiviso su Figshare.

[9] Tal Reich, Prithviraj Ammanabrolu, and Mark O. Riedl. A survey on prompt engineering for large language models. *arXiv preprint arXiv:2304.10030*, 2023.

[10] Ryan van Tonder and Claire Le Goues. Static automated program repair for heap properties. *Proceedings of the ACM on Programming Languages*, 2(OOPSLA):1–29, 2018.

Ringraziamenti

Desidero innanzitutto ringraziare i miei relatori, Fabio Palomba e Gianmario Voria, per la precisione e la disponibilità mostratami durante il periodo di sviluppo del progetto e di stesura della tesi.

Ringrazio la **mia famiglia**, per il sostegno costante, la pazienza infinita e per non aver mai smesso di chiedermi, con garbo e determinazione, "quando ti laurei?". Le vostre domande, a volte temute, hanno in realtà avuto un certo potere motivazionale.

Grazie a me stesso, per aver sempre trovato il modo di rimandare e poi sorprendentemente completare tutto all'ultimo momento. La procrastinazione creativa ha decisamente funzionato.

Un pensiero speciale va a **mio fratello**, presenza immancabile e punto di riferimento (più o meno attendibile). Tra un consiglio improvvisato e una battuta fuori luogo, è riuscito comunque a darmi sempre una prospettiva utile. Quasi sempre.

Ringrazio di cuore **i miei amici**, per aver condiviso con me gioie, fatiche e infinite ore di procrastinazione creativa. La loro vicinanza è stata un elemento fondamentale di equilibrio e motivazione.

Non potrei chiudere senza ringraziare la **mia ragazza**, che è stata accanto a me tra momenti di lavoro intenso, piccole frustrazioni ed inevitabili pause per distrarmi. La tua pazienza è stata incredibile, ed io ti devo tanto.

A tutti voi, il mio più sincero (e vagamente esausto) grazie.