



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria
Gestionale a.a. 2023/2024
Sessione di Laurea Novembre 2024

Valutazione di Query SQL tramite l'utilizzo di Large Language Models

Relatore:

Prof. Luca Cagliero

Candidato:

Daniele Sferlazzo (S317237)

Co-Relatore:

Prof. Laura Farinetti

Indice

1	Introduzione.....	1
1.1	Struttura della Relazione.....	2
2	Contesto ed Importanza	4
2.1	Large Language Models.....	4
2.2	Linguaggio SQL.....	6
2.3	Obiettivi della Ricerca.....	8
2.4	Domande di Ricerca	9
3	Revisione della Letteratura.....	12
3.1	Valutazione delle Query SQL	12
3.1.1	Errori Sintattici	12
3.1.2	Errori Semantici.....	13
3.1.3	Errori Logici	14
3.1.4	Punti di Attenzione.....	14
3.2	Utilizzo di Large Language Models	15
3.2.1	Introduzione e Contesto.....	15
3.2.2	Dataset e Metodologia	16
3.2.3	Esperimento ChatGPT-as-a-SQL-Assistant	17
3.2.4	Risultati Sperimentali.....	19
3.2.5	Conclusioni.....	19
4	Metodologia	20
4.1	Design della Ricerca	20
4.1.1	Origine del Dataset	20
4.1.2	Modelli di Machine Learning utilizzati	21
4.2	Descrizione degli errori utilizzati	23
4.2.1	Errori sintattici.....	24
4.2.2	Errori semantici	25
4.2.3	Errori Logici	26
4.2.4	Punti di Attenzione.....	27
4.2.5	INV Query Invalida	29
4.3	Descrizione del Dataset	29
4.3.1	Dati delle Esercitazioni	29
4.3.2	Dati di Supporto agli Esercizi	30
4.3.3	Selezione del Sub-Dataset	34

4.4 Addestramento dei Modelli	34
4.4.1 Implementazione dei Modelli	34
4.4.2 Prompt degli Errori	35
4.4.3 Prompt degli Esercizi	35
4.4.4 Interventi Correttivi	37
4.5 Tecniche di Classificazione degli Errori	37
5. Analisi dei Risultati	39
5.1 Prestazioni Generali dei Modelli.....	39
5.1.1 Confronto nel riconoscimento degli errori	40
5.1.2 Confronto nell'Accuratezza dei Voti	40
5.2 Analisi Specifica dei Modelli	41
5.2.1 ChatGPT	42
5.2.2 Gemma 2B	44
5.2.3 CodeStral.....	46
5.2.4 Llama 3.1 Sonar	48
6. Discussione	50
6.1 Confronto con la Letteratura	50
6.2 Implicazioni della Ricerca	52
6.2.1 ChatGPT	53
6.2.2 Gemma2B.....	54
6.2.3 CodeStral.....	55
6.2.4 Llama 3.1 Sonar	55
6.3 Costi di Inferenza.....	56
6.3.1 Comprensione dei Costi di Inferenza e Struttura dei Token	57
6.3.2 Confronto tra Modelli Open-Source e Modelli Proprietari.....	57
6.3.3 Considerazioni Sui Costi di Inferenza e Scelte di Implementazione	59
6.4 Limitazioni dello Studio	59
7. Conclusione.....	62
7.1 Sintesi dei principali risultati ottenuti	62
7.1.1 Accuratezza nella Valutazione delle Query SQL	62
7.1.2 Valutazione degli Errori Sintattici, Semantici e Logici.....	62
7.1.3 Prestazioni dei Modelli in con addestramento Few-Shot e Zero-Shot.....	63
7.1.4 Generosità e Severità dei Modelli.....	64
7.1.5 Identificazione dei Punti di Attenzione nelle Query	65
7.1.6 Contributi al Campo dell'Educazione	65
7.2 Contributi forniti alla Ricerca.....	66

7.2.1 Contributo Teorico.....	66
7.2.2 Contributo Pratico	67
7.2.3 Contributi alla Valutazione degli Errori	68
7.3 Principali Takeaways della Ricerca	69
7.4 Proposte di Ricerca Future	71
7.4.1 Applicazione di Altri Modelli LLM.....	71
7.4.2 Estensione del Set di Errori Considerati.....	71
7.4.3 Inclusione di Dati da Contesti Educativi Diversificati	72
7.4.4 Aumento del Dataset di Addestramento	72
7.4.5 Miglioramento dei Prompt e della Capacità di Addestramento	73
7.4.6 Adattamento alle Lingue Diverse	73
7.4.7 Integrazione di Modelli Ibridi.....	73
8. Appendici	74
8.1 Risultati del modello ChatGPT (primi 6 esercizi).....	74
8.2 Risultati del Modello Gemma2B (primi 6 esercizi)	76
8.3 Risultati del Modello CodeStral (primi 6 esercizi)	78
8.4 Risultati del Modello Llama 3.1 Sonar (primi 6 esercizi)	80
9. Bibliografia.....	82
10. Strumenti e Tecnologie Utilizzate	83
11. Ringraziamenti	84

1 Introduzione

Negli ultimi anni, l'impiego dei Large Language Models (LLM), come ChatGPT e altri, ha rivoluzionato una vasta gamma di settori, trasformando radicalmente l'assistenza virtuale, la generazione di contenuti e molte altre applicazioni digitali. La capacità di questi modelli di comprendere e generare testo con un'accuratezza e una coerenza eccezionali li rende strumenti particolarmente promettenti anche nel campo dell'educazione. Un'area di applicazione in crescita è l'automazione delle attività di valutazione e feedback per esercizi complessi, come quelli inerenti alle query SQL, dove l'intervento umano è tradizionalmente considerato essenziale a causa delle molteplici sfumature nella valutazione della correttezza di una query. Questa tesi si inserisce in tale contesto esplorando il potenziale dei LLM nella valutazione automatizzata delle query SQL, con l'obiettivo di supportare i docenti nella correzione delle risposte degli studenti e nel fornire un feedback immediato e accurato.

SQL, ovvero Structured Query Language, è uno dei linguaggi cardine per la gestione e l'interrogazione di database, con applicazioni che spaziano dall'ambito accademico a quello professionale, dove l'abilità di gestire database e di manipolare informazioni è cruciale. Nonostante la sua larga diffusione, l'insegnamento e la valutazione delle competenze SQL rappresentano sfide complesse. La varietà degli errori che gli studenti possono commettere è ampia e include errori sintattici (come una struttura non corretta della query) e errori più sottili, come quelli semantici e logici. Per esempio, una query può essere scritta correttamente dal punto di vista sintattico, ma può restituire un risultato errato perché non interpreta correttamente la logica richiesta. Data la natura articolata della valutazione SQL, l'automazione richiede non solo la capacità di identificare gli errori ma anche di fornire un feedback utile che supporti lo studente nell'approfondire la propria comprensione.

Questa ricerca si è quindi concentrata sull'analisi di diversi modelli LLM per valutarne l'efficacia nella correzione automatizzata di esercizi SQL, utilizzando un dataset strutturato attorno a errori comuni nelle query SQL. I modelli esaminati includono opzioni open-source, come Llama 3.1 Sonar, e modelli proprietari, tra cui ChatGPT di OpenAI. Per ogni modello sono stati presi in considerazione parametri come la precisione nel riconoscere e classificare gli errori e la capacità di offrire valutazioni coerenti con quelle di un docente umano, il tutto per comprendere quanto questi modelli possano rispecchiare le aspettative e le necessità educative in un contesto di apprendimento.

Nella prima parte dello studio sono stati analizzati gli errori più frequenti commessi dagli studenti nelle query SQL, con l'obiettivo di determinarne un sottoinsieme rappresentativo. Questo sottoinsieme è stato poi descritto in dettaglio, fornendo ai modelli sufficienti informazioni per distinguere un tipo di errore dall'altro, senza però superare il numero di token disponibili per i modelli.

I risultati ottenuti indicano che i LLM possiedono capacità promettenti, in particolare nell'individuare errori sintattici e nel fornire riscontri che risultano preziosi per il contesto educativo. Tuttavia, sono emerse alcune limitazioni: i modelli analizzati presentano difficoltà nella comprensione di errori semantici complessi e nella corretta interpretazione delle intenzioni sottese alla query, soprattutto quando queste implicano una logica articolata o presuppongono una profonda conoscenza del database. Tali limiti indicano la necessità di ulteriori sviluppi per rendere i LLM strumenti affidabili e autonomi nella valutazione di esercizi SQL, in modo da ridurre il bisogno di interventi correttivi umani.

Una componente rilevante di questa tesi ha riguardato, inoltre, il confronto tra l'accuratezza e i costi operativi dei modelli proprietari rispetto a quelli open-source. Tale comparativa ha consentito di evidenziare i trade-off associati all'uso dei diversi modelli, sia in termini di efficienza economica sia di prestazioni. I costi di inferenza, che dipendono principalmente dal numero di token necessari per analizzare e generare risposte a query SQL, sono un fattore determinante per l'adozione degli LLM. I modelli open-source si mostrano economicamente vantaggiosi in contesti con risorse limitate, ma spesso offrono prestazioni inferiori rispetto ai modelli proprietari, che, pur richiedendo un investimento maggiore, tendono a garantire risultati più accurati e affidabili.

Questo lavoro rappresenta un contributo significativo nell'ambito dell'educazione supportata dall'intelligenza artificiale, proponendo l'uso dei LLM come strumenti per l'automazione della valutazione delle query SQL. La ricerca fornisce una visione complessiva delle potenzialità e delle limitazioni di questi modelli, dimostrando che, pur richiedendo ancora ottimizzazioni, gli LLM possono rappresentare una risorsa preziosa per supportare l'apprendimento dell'SQL e offrire feedback tempestivo e personalizzato. Pur non sostituendo completamente l'intervento umano, l'integrazione di LLM nel processo di apprendimento delle query SQL offre agli studenti la possibilità di sperimentare e migliorare le proprie competenze in autonomia, riservando ai docenti un ruolo più consulenziale e mirato.

1.1 Struttura della Relazione

Questa tesi è organizzata in sei capitoli principali, ognuno dei quali contribuisce a costruire un quadro completo del progetto di ricerca. Di seguito viene fornita una panoramica dettagliata di ciascun capitolo.

Capitolo 2: Contesto ed Importanza

Vengono analizzati i principali usi e le caratteristiche delle tecnologie trattate in questa ricerca, come gli algoritmi di Machine Learning e il linguaggio SQL. Successivamente, sono esposte le principali domande di ricerca e i relativi obiettivi.

Capitolo 3: Revisione della Letteratura

Viene fornita una revisione critica della letteratura esistente relativa ai database relazionali, SQL, e all'uso di LLM nell'elaborazione del linguaggio SQL. Si esplorano le tecniche tradizionali di valutazione delle query SQL e si esaminano i recenti progressi nell'uso di modelli di machine learning per compiti simili. La revisione della letteratura fornisce il contesto teorico necessario e giustifica la necessità della presente ricerca.

Capitolo 4: Metodologia

Il capitolo descrive in dettaglio la metodologia utilizzata per condurre la ricerca. Viene spiegato il processo di raccolta e preparazione dei dati, la progettazione del sistema basato su LLM e le tecniche utilizzate per addestrare e testare i modelli.

Capitolo 5: Analisi dei Risultati

Vengono descritti i risultati ottenuti durante la fase di valutazione dei modelli, analizzando sia le prestazioni generali, sia quelle specifiche dei vari modelli e dei rispettivi metodi di addestramento utilizzati.

Capitolo 6: Risultati e Discussione

Il sesto capitolo presenta i risultati ottenuti dall'implementazione e dal testing del sistema. Viene fornita un'analisi dettagliata delle prestazioni del sistema in termini di precisione, tempo di valutazione e qualità del feedback. I risultati vengono confrontati con quelli ottenuti tramite metodi tradizionali di valutazione delle query SQL. Inoltre, si discutono le implicazioni dei risultati, le limitazioni del sistema e le potenziali aree di miglioramento.

Capitolo 7: Conclusioni e Lavori Futuri

L'ultimo capitolo riassume i principali contributi della ricerca, evidenziando come i LLM possano essere utilizzati efficacemente per la valutazione automatizzata delle query SQL. Vengono discusse le conclusioni tratte dai risultati e vengono proposte direzioni per futuri lavori di ricerca. Questo capitolo riflette anche sulle implicazioni pratiche della ricerca per l'educazione e la formazione in SQL, nonché per l'industria dei database.

2 Contesto ed Importanza

2.1 Large Language Models

Negli ultimi anni, l'avanzamento dell'intelligenza artificiale (AI) e del machine learning ha reso possibile lo sviluppo di modelli di linguaggio di grandi dimensioni, noti come Large Language Models (LLM). Questi modelli, tra cui spiccano GPT-3 di OpenAI, BERT di Google e altri, si basano su reti neurali avanzate e sono stati addestrati su vasti corpus di testo provenienti da fonti eterogenee. La loro capacità di comprendere e generare testo in maniera accurata e contestuale li ha resi strumenti potenti, impiegati in vari ambiti come l'assistenza virtuale, la generazione di contenuti e l'automazione di processi complessi.

Storia e Sviluppo dei LLM

La storia degli LLM ha radici nelle prime reti neurali per l'elaborazione del linguaggio naturale, ma il settore ha conosciuto una rivoluzione con l'avvento dei modelli basati su trasformatori. Inizialmente, i modelli di linguaggio erano limitati dalla quantità di dati e dalle capacità computazionali, con parametri molto meno sofisticati rispetto a quelli attuali. Il rilascio di BERT (Bidirectional Encoder Representations from Transformers) da parte di Google nel 2018 ha segnato un cambiamento epocale, introducendo una comprensione bidirezionale del contesto. Ciò ha permesso di interpretare il significato di una parola in relazione alle altre parole circostanti in una frase. Successivamente, OpenAI ha introdotto GPT-3 nel 2020, con un modello dotato di 175 miliardi di parametri, che ha dimostrato straordinarie capacità nella generazione di testo coerente e nella comprensione delle richieste dell'utente, avvicinandosi sempre più a risposte di tipo umano.

Architettura dei LLM

I LLM utilizzano principalmente l'architettura del trasformatore, una struttura di rete neurale che sfrutta un meccanismo di attenzione di tipo multi-head. Questo permette al modello di analizzare una frase tenendo conto del contesto di tutte le parole e di stabilire relazioni di significato tra di esse, anche a lunga distanza. Il trasformatore rappresenta una delle innovazioni più significative per l'elaborazione del linguaggio naturale, poiché consente una comprensione profonda e flessibile del testo, riducendo la necessità di processi sequenziali e aumentando l'efficienza computazionale.

Metodologie di Addestramento

L'addestramento degli LLM è un processo complesso e costoso, che richiede dataset testuali di dimensioni straordinarie e risorse computazionali avanzate come GPU o TPU. Utilizzando l'unsupervised learning, i modelli vengono addestrati per prevedere parole mancanti o completare frasi. Questo li dota di una capacità intrinseca di prevedere il flusso naturale del linguaggio e di rispondere in modo coerente. L'ottimizzazione avviene su miliardi di parametri

attraverso iterazioni prolungate, con l'obiettivo di minimizzare l'errore nella previsione delle parole e migliorare la capacità complessiva del modello di interpretare il linguaggio naturale.

Applicazioni Pratiche dei LLM

Gli LLM trovano applicazione in una vasta gamma di settori, dimostrando un impatto significativo nel migliorare le interazioni tra uomo e tecnologia e automatizzando operazioni complesse. Nell'assistenza virtuale, questi modelli alimentano chatbot e assistenti digitali, rispondendo a domande complesse, assistendo gli utenti e risolvendo problemi in tempo reale, per esempio nel servizio clienti. Nella traduzione automatica, i LLM hanno elevato la qualità delle traduzioni, riducendo sensibilmente errori e ambiguità linguistiche e supportando la comunicazione tra lingue diverse. La scrittura automatizzata rappresenta un'altra area di applicazione: i modelli generano articoli, rapporti, riassunti e contenuti creativi, velocizzando il lavoro di scrittura e assicurando una qualità professionale. In ambito aziendale e di marketing, gli LLM vengono usati per l'analisi dei sentimenti e il riconoscimento delle emozioni nei dati, interpretando con precisione le reazioni dei clienti e supportando le decisioni strategiche, contribuendo a costruire esperienze più personalizzate e reattive.

Sfide e Limiti dei LLM

Nonostante i progressi, i LLM presentano ancora sfide significative. L'addestramento di modelli di grandi dimensioni richiede ingenti risorse di dati e computazionali, limitando l'accesso a tali tecnologie solo alle grandi organizzazioni. Gli LLM, inoltre, possono generare risposte plausibili ma non sempre accurate, fenomeno noto come "allucinazione" del modello, il che rappresenta una criticità per applicazioni che richiedono precisione assoluta. Un'altra limitazione è la presenza di bias nei dati di addestramento, che può portare a risposte inappropriate o parziali, influenzate da pregiudizi insiti nei dataset. Infine, la complessità dell'architettura e la scarsa trasparenza nei processi decisionali ostacolano la spiegabilità dei modelli, rendendo difficile interpretare il modo in cui il modello giunge a determinate conclusioni.

Futuro dei LLM e Innovazioni in Corso

Il futuro dei LLM appare promettente, con l'implementazione di tecniche come l'apprendimento multi-task e continuo, l'integrazione con dati strutturati e il miglioramento dell'efficienza energetica. La riduzione dell'impatto ambientale è diventata una priorità, con la ricerca di metodi di addestramento più sostenibili. Le innovazioni puntano anche a incrementare la flessibilità dei modelli, adattandoli a contesti e lingue diverse, e a introdurre architetture ibride che combinano LLM con altri tipi di AI per rispondere a domande specifiche e ad esigenze particolari.

Utilizzo dei LLM nella Ricerca

Nel contesto della presente ricerca, i Large Language Models vengono applicati per valutare la correttezza e identificare le tipologie di errore nelle query SQL, un compito complesso ma cruciale nel campo dell'educazione tecnica. SQL è un linguaggio di query fondamentale nella

gestione dei dati, ma la sua complessità richiede un apprendimento preciso, dove gli errori possono facilmente compromettere i risultati. L'uso degli LLM consente di automatizzare la valutazione delle query, identificando errori sintattici, semantici e logici e fornendo suggerimenti specifici per la correzione. Questo approccio non solo contribuisce ad alleggerire il carico di lavoro dei docenti, ma rappresenta anche un prezioso strumento formativo per gli studenti, aiutandoli a comprendere meglio le strutture e la logica delle query SQL e a migliorare le loro competenze in modo mirato.

In sintesi, gli LLM si stanno affermando come strumenti versatili e potenti, capaci di trasformare diversi ambiti, dall'assistenza clienti alla ricerca scientifica, fino all'educazione. I loro progressi e le loro applicazioni lasciano intravedere un futuro in cui l'interazione con le macchine diventa sempre più naturale ed efficiente, ampliando le possibilità di impiego e rendendo l'AI un supporto fondamentale nelle attività quotidiane e professionali.

2.2 Linguaggio SQL

Nel mondo moderno, i database rappresentano uno degli elementi fondamentali per la gestione e l'organizzazione delle informazioni in una vasta gamma di settori, tra cui finanza, sanità, e-commerce e molti altri. Tra questi, spiccano per importanza e diffusione i database strutturati, componenti cruciali dell'infrastruttura informatica moderna. Utilizzati per organizzare, archiviare e gestire dati in modo efficiente e accessibile, e caratterizzati da una struttura rigorosa e predeterminata, questi database memorizzano i dati in tabelle con righe e colonne, dove ogni colonna rappresenta un attributo dei dati e ogni riga una singola voce o record.

Storia e Evoluzione di SQL

SQL, acronimo di Structured Query Language, è stato sviluppato negli anni '70 presso IBM da Donald D. Chamberlin e Raymond F. Boyce. La prima implementazione commerciale di SQL fu rilasciata nel 1979 da Oracle Corporation. Negli anni successivi, SQL è diventato il linguaggio standard per la gestione dei database relazionali, grazie alla sua facilità d'uso e alla potenza espressiva. L'American National Standards Institute (ANSI) ha adottato SQL come standard nel 1986, seguito dall'International Organization for Standardization (ISO) nel 1987. Nel corso degli anni, SQL ha subito diverse revisioni e aggiornamenti per includere nuove funzionalità e miglioramenti, rendendolo un linguaggio sempre al passo con le esigenze dell'industria.

Principali Caratteristiche di SQL

SQL offre una vasta gamma di funzionalità che lo rendono uno strumento indispensabile per la gestione dei dati. Tra le sue principali caratteristiche vi sono:

- **Interrogazione dei Dati:** Permette di estrarre informazioni specifiche dai database utilizzando comandi come SELECT, WHERE e JOIN.
- **Manipolazione dei Dati:** Consente di inserire (INSERT), aggiornare (UPDATE) ed eliminare (DELETE) dati dalle tabelle.
- **Definizione dei Dati:** Permette di definire la struttura del database attraverso comandi come CREATE, ALTER e DROP.
- **Controllo dei Dati:** Fornisce strumenti per gestire i permessi di accesso e garantire la sicurezza dei dati tramite i comandi GRANT e REVOKE.
- **Transazioni:** Supporta le operazioni transazionali, garantendo che le modifiche ai dati siano eseguite in modo atomico e coerente.

Importanza di SQL nell'Industria

SQL è fondamentale per una vasta gamma di applicazioni industriali. In ambito finanziario, le banche e le istituzioni finanziarie utilizzano SQL per gestire transazioni e analizzare dati finanziari. Nel settore sanitario, SQL è impiegato per gestire le cartelle cliniche elettroniche e facilitare la ricerca medica. Nell'e-commerce, SQL aiuta a gestire inventari, ordini e dati dei clienti, migliorando l'efficienza operativa e l'esperienza del cliente. La sua robustezza e versatilità lo rendono un componente chiave per qualsiasi sistema informativo aziendale.

Questo tipo di database utilizza il modello relazionale, introdotto negli anni '70, che permette di stabilire relazioni logiche tra diverse tabelle attraverso l'uso di chiavi primarie e chiavi esterne. Structured Query Language (SQL) è il linguaggio standard per la gestione e manipolazione di database relazionali e viene utilizzato da milioni di sviluppatori e analisti in tutto il mondo per interrogare, aggiornare e gestire dati strutturati.

Sfide Comuni nell'Apprendimento di SQL

Nonostante la sua potenza, l'apprendimento di SQL può presentare diverse sfide. Gli studenti e i nuovi sviluppatori spesso faticano a comprendere la logica delle query complesse e la manipolazione dei dati. Errori comuni includono l'uso improprio delle clausole JOIN, l'incapacità di ottimizzare le query e la difficoltà nel garantire l'integrità referenziale. La necessità di padroneggiare sia la sintassi del linguaggio che la logica del modello relazionale richiede tempo e pratica.

L'efficienza e l'accuratezza delle query SQL sono cruciali per assicurare che i sistemi informativi operino correttamente e che le decisioni aziendali siano basate su dati precisi e tempestivi. Tuttavia, scrivere query SQL corrette può essere complesso e soggetto a errori, specialmente per chi è alle prime armi. Errori nelle query possono portare a risultati incompleti, errati o addirittura alla compromissione dell'integrità dei dati.

SQL nel Contesto di Big Data e Cloud Computing

Con l'avvento dei Big Data e del cloud computing, SQL ha mantenuto la sua rilevanza evolvendosi per soddisfare le nuove esigenze. Strumenti come Apache Hive e Google BigQuery utilizzano sintassi SQL per consentire l'interrogazione di enormi volumi di dati distribuiti su cluster di server. Le piattaforme cloud offrono servizi di database relazionali scalabili, come Amazon RDS e Microsoft Azure SQL Database, che permettono alle aziende di gestire e analizzare grandi quantità di dati senza dover gestire l'infrastruttura sottostante.

Il Futuro del Linguaggio SQL

Il processo di insegnamento e valutazione delle query SQL richiede una comprensione approfondita sia della sintassi del linguaggio che della logica sottostante alle operazioni sui dati. Identificare e correggere gli errori nelle query SQL è una parte fondamentale del processo di apprendimento, poiché permette agli studenti di comprendere meglio le strutture dei dati e le relazioni tra essi. Inoltre, una valutazione accurata delle query SQL è cruciale per garantire che i database funzionino correttamente e che le informazioni recuperate siano affidabili e precise.

Tradizionalmente, la valutazione delle query SQL e l'identificazione degli errori sono stati compiti affidati a esperti umani, un processo che può essere dispendioso in termini di tempo e soggetto a bias. Con l'avanzare della tecnologia, l'automazione di questi processi è diventata una priorità per migliorare l'efficienza e ridurre i costi. In questo contesto, i Large Language Models (LLM) e gli algoritmi di machine learning rappresentano una soluzione promettente.

2.3 Obiettivi della Ricerca

La ricerca si propone di esplorare la possibilità e la fattibilità di un sistema basato su Large Language Models (LLM) per valutare la correttezza delle query SQL e identificare i diversi tipi di errori che possono verificarsi durante la loro scrittura, fornendo così un supporto efficace agli esperti di dominio.

1. Supporto nella Valutazione delle Query SQL

Uno degli obiettivi principali della ricerca è quello di fornire supporto nel processo di valutazione delle query SQL agli esperti di dominio. Attualmente, la verifica della correttezza delle query SQL è spesso eseguita manualmente da esperti, un processo che può essere dispendioso in termini di tempo e soggetto a bias umani. L'introduzione di un sistema automatizzato basato su LLM come supporto ad un esperto di dominio può ridurre significativamente il tempo necessario per valutare le query e aumentare la precisione della valutazione.

2. Classificazione degli Errori nelle Query SQL

Un altro obiettivo cruciale è sviluppare un sistema che non solo identifichi se una query SQL è corretta o meno, ma che sia anche in grado di classificare il tipo di errore presente nella query. Gli errori nelle query possono essere di diversi tipi, tra cui errori sintattici, errori logici, semantici o punti di attenzione. Un sistema che può classificare questi errori fornirà un feedback più dettagliato agli utenti, aiutandoli a comprendere meglio le loro aree di debolezza e come correggerle.

3. Analisi Dettagliata degli Errori

La ricerca mira anche a fornire un'analisi dettagliata degli errori nelle query SQL. Non basta sapere che una query è errata; è importante capire perché è errata. Utilizzando i LLM, il sistema può analizzare il contesto della query e fornire spiegazioni dettagliate su cosa è andato storto. Questo approccio non solo aiuta a correggere l'errore corrente, ma fornisce anche una comprensione più profonda dei principi SQL, che può prevenire errori futuri.

4. Miglioramento dell'Educazione e della Formazione in SQL

L'utilizzo dei LLM per la valutazione delle query SQL ha il potenziale di migliorare significativamente l'educazione e la formazione in SQL. Uno degli obiettivi della ricerca è la valutazione di uno strumento didattico che possa essere utilizzato da studenti e professionisti per apprendere SQL in modo più efficace. Fornendo feedback immediato e dettagliato sulle query, il sistema può aiutare gli studenti a capire meglio i concetti fondamentali di SQL e migliorare le loro competenze pratiche.

5. Confronto dei risultati di analisi di tipo Few-Shot e Zero-Shot Learning

La ricerca prevede l'analisi dei risultati dei modelli sia in un contesto zero-shot che in uno few-shot. Mentre nel zero-shot learning viene presentata al modello soltanto la query errata, nel few-shot learning vengono fornite anche delle query contenenti gli errori più frequenti commessi dagli studenti per quell'esercizio.

6. Confronto con Metodi Tradizionali

Infine, uno degli obiettivi della ricerca è confrontare le prestazioni del sistema basato su LLM con i metodi tradizionali di valutazione delle query SQL. Questo confronto includerà metriche come la precisione e l'accuratezza dei risultati. Dimostrare la superiorità o l'efficacia comparabile del sistema basato su LLM rispetto ai metodi tradizionali sarà cruciale per la sua adozione pratica.

2.4 Domande di Ricerca

L'obiettivo di questa ricerca è valutare se i LLM possano essere strumenti utili sia in contesti educativi che nell'automazione della valutazione. Queste domande di ricerca costituiscono

l'ossatura di questo studio, orientando l'esplorazione di come i LLM possano essere utilizzati nella valutazione e nell'insegnamento delle query SQL

Domanda 1: I LLM possono automatizzare efficacemente la valutazione delle query SQL?

L'automazione della valutazione delle query SQL è da tempo una sfida, soprattutto a causa della complessità e della varietà di errori che possono emergere. Questi errori includono errori sintattici, logici e semantici, ciascuno dei quali richiede livelli diversi di interpretazione e correzione.

Per affrontare questa domanda, è essenziale considerare la struttura di base dei LLM. Questi modelli, addestrati su enormi quantità di dati, inclusi linguaggi di programmazione, hanno dimostrato una notevole capacità di generare risposte simili a quelle umane, riconoscere modelli e interpretare il linguaggio naturale. Tuttavia, le query SQL richiedono più di una semplice identificazione di modelli: richiedono una comprensione profonda delle strutture logiche, degli schemi dei database e dei risultati attesi della query.

L'ipotesi alla base di questa domanda di ricerca è che gli LLM possano essere strumenti efficaci per la valutazione delle query SQL, in particolare per l'identificazione degli errori sintattici e per offrire un feedback di base. In sistemi di valutazione automatica, per esempio, gli LLM potrebbero fornire un feedback in tempo reale agli studenti o agli sviluppatori, aiutandoli a individuare gli errori senza la necessità di un istruttore umano. Tuttavia, possono emergere delle difficoltà nel discernere questioni più complesse, come gli errori logici, che spesso richiedono la comprensione dell'intento del programmatore piuttosto che una semplice verifica della sintassi SQL.

Domanda 2: I LLM possono classificare accuratamente i diversi tipi di errori nelle query SQL?

Le query SQL presentano spesso diverse categorie di errori, tra cui errori sintattici, semantici e logici. La seconda domanda di ricerca si concentra sulla capacità degli LLM di differenziare tra questi tipi di errori e fornire classificazioni accurate.

1. **Errori sintattici:** I LLM, addestrati ampiamente su codice e testi, sono in grado di identificare con efficacia questo tipo di errori, individuando rapidamente la mancanza di punti e virgola, virgole mal posizionate o sintassi SQL invalide.
2. **Errori semantici:** Identificare gli errori semantici è più complesso per i modelli, poiché richiede una comprensione del rapporto tra le tabelle, della logica dietro la query e dell'intento del programmatore.
3. **Errori logici:** Gli errori logici sono forse i più complessi da rilevare. Essi derivano dall'applicazione errata della logica nella query. Rilevare questi errori richiede una

comprensione approfondita delle operazioni SQL, delle relazioni tra tabelle e degli obiettivi della query.

I Large Language Models, nella loro forma attuale, eccellono nella classificazione degli errori sintattici e possono gestire alcune problematiche semantiche, soprattutto quando queste si riferiscono a modelli comuni nelle query SQL. Tuttavia, la loro capacità di rilevare e classificare gli errori logici rimane un'area che richiede miglioramenti. Una delle ragioni di questa limitazione è che gli errori logici sono spesso molto contestuali, richiedendo al modello di interpretare l'intento del programmatore, che non è sempre evidente basandosi solo sulla query.

Domanda 3: I LLM possono fornire valutazioni simili a quelle dei docenti e coerenti con il contesto fornito?

I modelli di linguaggio, come ChatGPT e altri LLM, sono progettati per comprendere il linguaggio naturale e generare risposte simili a quelle umane. Tuttavia, la sfida in questo contesto è se possano essere abbastanza sofisticati da offrire valutazioni che non siano solo tecnicamente accurate, ma anche pedagogicamente utili. Ad esempio, un modello potrebbe identificare un errore sintattico ma non tener conto del fatto che lo studente ha dimostrato una comprensione avanzata della logica della query, elemento che potrebbe influenzare positivamente la valutazione complessiva assegnata da un docente umano.

L'ipotesi alla base di questa domanda è che i LLM, con un adeguato addestramento e un contesto fornito in maniera dettagliata, possano avvicinarsi alla valutazione umana, sia in termini di accuratezza sia di coerenza con il contesto educativo. Tuttavia, i risultati della sperimentazione hanno evidenziato che, sebbene i LLM riescano a replicare gran parte del processo valutativo, presentano ancora delle difficoltà nell'assegnare il massimo punteggio (100), soprattutto quando devono discernere tra una soluzione completamente corretta e una che si avvicina solo parzialmente alla risposta attesa. In particolare, i modelli tendono a essere più severi rispetto ai docenti umani, assegnando voti più bassi anche in situazioni in cui un docente potrebbe essere più indulgente.

Questa difficoltà nella valutazione può essere legata alla mancanza di un contesto emotivo e pedagogico che guida il processo valutativo umano, e che i LLM non possono replicare facilmente. La sfida per il futuro sarà dunque quella di migliorare la capacità di questi modelli di tenere conto del contesto educativo complessivo, non solo degli errori tecnici.

3 Revisione della Letteratura

3.1 Valutazione delle Query SQL

La letteratura accademica ha ampiamente esplorato le sfide connesse alla valutazione delle query, mettendo in luce le difficoltà incontrate da studenti e professionisti nel formulare query corrette. L'analisi delle query include la diagnosi di errori sintattici, semantici e logici, con lo scopo di migliorare sia le competenze dei programmatori che l'affidabilità delle applicazioni basate su database.

3.1.1 Errori Sintattici

Gli errori sintattici sono generalmente i più facili da identificare e correggere, grazie alla natura rigida delle regole sintattiche del linguaggio e al supporto fornito dai Database Management Systems (DBMS). Come discusso da Ahadi et al. (2015), gli errori sintattici si verificano quando una query non segue la grammatica e le regole del linguaggio SQL. Questi errori possono includere il mancato utilizzo di parole chiave corrette, errori di digitazione, o la mancata chiusura di parentesi, tra gli altri. I DBMS come PostgreSQL sono in grado di intercettare tali errori e fornire messaggi di errore specifici che aiutano l'utente a individuare e correggere rapidamente il problema.

Tipologie di Errori Sintattici

Taipalus et al. (2018) hanno classificato gli errori sintattici in diverse categorie, tra cui:

- **Ambiguous Database Object (SYN-1):** Errori derivanti dall'uso di nomi ambigui per oggetti del database, che possono confondere il DBMS.
- **Undefined Database Object (SYN-2):** Riferimenti a oggetti del database inesistenti, che portano a errori di sintassi.
- **Data Type Mismatch (SYN-3):** Errori causati dall'uso di operatori o funzioni inappropriati per i tipi di dati delle colonne.
- **Illegal Aggregate Function Placement (SYN-4):** Utilizzo improprio delle funzioni aggregate in clausole non consentite.
- **Illegal or Insufficient Grouping (SYN-5):** Errori relativi all'implementazione del raggruppamento che non rispettano gli standard SQL.
- **Common Syntax Error (SYN-6):** Errori nella struttura della query o nella sua sintassi.

Queste categorizzazioni hanno permesso di identificare con precisione le aree in cui gli studenti e i programmatori tendono a commettere errori, fornendo una base per lo sviluppo di strumenti di insegnamento e di valutazione più mirati.

3.1.2 Errori Semantici

Gli errori semantici, a differenza di quelli sintattici, sono più difficili da individuare poiché una query può essere sintatticamente corretta ma semanticamente errata. Come evidenziato da Ahadi et al., gli errori semantici si manifestano quando il risultato di una query non corrisponde all'intenzione del programmatore, pur essendo privo di errori sintattici. Questo tipo di errore è particolarmente insidioso perché non viene automaticamente rilevato dai DBMS, che eseguono la query senza segnalare problemi, restituendo però un risultato che potrebbe essere incompleto o completamente errato.

Tipologie di Errori Semantici

La classificazione di Taipalus et al. (2018) degli errori semantici include:

- **Inconsistent Expression (SEM-1):** Espressioni che causano risultati vuoti o che includono tutte le righe della tabella, contrariamente alle intenzioni dell'utente.
- **Inconsistent Join (SEM-2):** Join che, a causa di condizioni errate, restituiscono risultati incompleti o non corretti.
- **Missing Join (SEM-3):** L'assenza di clausole JOIN necessarie, che porta a risultati incompleti.
- **Duplicate Rows (SEM-4):** La presenza di righe duplicate nel risultato della query, indicante un errore nella proiezione o nella condizione di join.
- **Redundant Column Output (SEM-5):** Inclusione di colonne inutili nel risultato, che non aggiungono valore all'output desiderato.

Classificazione di Ahadi et al.

Ahadi et al. hanno effettuato una classificazione degli errori semantici riscontrati, suddividendoli in base alla loro parte di codice:

- **WHERE:** condizioni mancanti, riscontrate principalmente in query semplici, self-join, subquery correlate e JOIN.
- **FROM:** errore nella selezione delle tabelle e assenza di self-join.
- **HAVING:** mancanza di clausole GROUP BY e HAVING o utilizzo di colonne errate.
- **ORDER BY:** mancanza della clausola quando richiesta o utilizzo di colonne errate.
- **SELECT:** colonne mancanti o superflue nella clausola, riscontrate principalmente in query semplici.
- **GROUP BY:** mancanza della clausola o utilizzo di colonne errate.

Ahadi et al. hanno scoperto che l'errore più comune era l'omissione, in cui gli studenti dimenticavano di includere parti cruciali della query, come le clausole JOIN o GROUP BY.

Hanno anche confrontato i loro risultati con le categorizzazioni degli errori di studi precedenti, come quelli di Reisner e Welty, confermando che molti degli errori osservati potevano essere classificati come "omissioni".

3.1.3 Errori Logici

Taipalus et al. introducono anche una categoria separata di errori, denominata errori logici, che si riferisce a problemi concettuali nella costruzione della query. Questi errori non sempre compromettono la correttezza sintattica o semantica della query, ma indicano una comprensione errata della logica del database o delle specifiche del problema.

Tipologie di Errori Logici

Alcuni degli errori logici identificati includono:

- **Operator Error (LOG-1):** Utilizzo errato di operatori di confronto o logici, come l'uso di OR al posto di AND, che può alterare significativamente il risultato della query.
- **Join Error (LOG-2):** Esecuzione di join su tabelle sbagliate o l'omissione di join necessari, che può portare a risultati corretti solo apparentemente.
- **Nesting Error (LOG-3):** Problemi con la logica dei raggruppamenti e delle subquery, spesso dovuti a difficoltà nella gestione delle parentesi e della logica booleana.
- **Expression Error (LOG-4):** Errori nell'uso delle espressioni, come l'uso di colonne sbagliate o l'omissione di espressioni necessarie.
- **Projection Error (LOG-5):** Errori nella selezione delle colonne da proiettare, inclusa la mancanza di colonne o l'inclusione di colonne non necessarie.
- **Function Error (LOG-6):** Riguardanti un utilizzo errato di funzioni aggregate e dei loro parametri

3.1.4 Punti di Attenzione

I punti di attenzione, chiamati complicazioni da Taipalus et al., si riferiscono a query che, pur essendo corrette, sono inutilmente complesse o ridondanti. Questi punti non influiscono direttamente sul risultato della query, ma possono compromettere l'efficienza della query e renderne più difficile la lettura

I punti di attenzione comuni includono:

- **Redundant Operations:** Operazioni ripetitive o superflue che potrebbero essere semplificate.
- **Unnecessary Columns:** Colonne incluse nel risultato che non sono necessarie per soddisfare la richiesta dell'utente.

3.2 Utilizzo di Large Language Models

3.2.1 Introduzione e Contesto

Lo studio precedentemente condotto da Cagliero et al. ha indagato l'efficacia di ChatGPT di OpenAI come strumento di supporto per l'insegnamento delle query SQL, con un focus particolare sull'automazione della correzione degli esercizi SQL somministrati agli studenti in corsi universitari di Database Management. Questa ricerca si distingue per un approccio rigoroso e quantitativo, orientato a valutare l'affidabilità e le potenzialità del modello nel contesto didattico, dove la capacità di fornire feedback rapido e accurato potrebbe agevolare sia gli studenti sia i docenti. L'attenzione è stata posta sulla possibilità di integrare ChatGPT nei corsi come assistente, in grado di correggere esercizi e fornire spiegazioni per aiutare gli studenti a comprendere meglio i loro errori.

Per svolgere l'analisi, Cagliero et al. si sono concentrati su tre domande di ricerca principali che esplorano la comparabilità, la precisione e l'efficacia dei modelli di linguaggio nei contesti educativi, affrontando aspetti critici per l'insegnamento e l'apprendimento delle query SQL:

1. **I voti assegnati da ChatGPT agli esercizi SQL sono comparabili a quelli dei docenti umani?**

Questo primo quesito rappresenta un elemento chiave per l'effettiva applicabilità di ChatGPT come strumento di supporto alla valutazione, poiché indica se il modello è in grado di emettere giudizi e assegnare punteggi che riflettano in modo accurato i criteri di valutazione dei docenti. La comparabilità dei voti tra ChatGPT e i docenti umani è fondamentale per assicurare che il modello possa, almeno in parte, replicare la valutazione umana, preservando la coerenza e l'equità nel contesto educativo. Lo studio ha quindi misurato la correlazione tra le valutazioni di ChatGPT e quelle dei docenti, cercando di individuare eventuali divergenze nei giudizi e le possibili motivazioni alla base di queste.

2. **ChatGPT è in grado di distinguere tra errori sintattici e semantici?**

Il secondo punto esplora la capacità di ChatGPT di riconoscere e classificare diverse tipologie di errori nelle query SQL. La distinzione tra errori sintattici, legati alla struttura formale delle query, ed errori semantici, che riguardano la logica e l'intento della query, è essenziale per un feedback educativo efficace. Se ChatGPT può identificare con precisione queste due categorie, potrebbe fornire feedback più dettagliato e informativo agli studenti, aiutandoli a capire non solo dove si è verificato l'errore, ma anche perché la logica della query non rispecchia l'intenzione richiesta dall'esercizio.

La capacità di ChatGPT di differenziare tra errori di sintassi e semantica è quindi cruciale per un approccio didattico che punti non solo alla correzione ma anche all'approfondimento della comprensione dello studente.

3. L'uso del few-shot learning migliora la qualità delle valutazioni di ChatGPT rispetto a un contesto di zero-shot learning?

La terza domanda esplora l'efficacia di due approcci di apprendimento per migliorare la precisione e la coerenza delle valutazioni di ChatGPT: il zero-shot learning, dove il modello riceve semplicemente una query senza esempi aggiuntivi, e il few-shot learning, che fornisce al modello alcuni esempi contestuali per guidarne il processo di valutazione. L'intento è verificare se il few-shot learning può ridurre gli errori di valutazione, permettendo al modello di beneficiare del contesto per rispondere in maniera più accurata alle esigenze didattiche. Questa analisi consente di comprendere in che misura l'inclusione di esempi migliora la capacità di ChatGPT di comprendere l'intento dietro una query, contribuendo così a fornire un feedback più contestualizzato e meno soggetto a interpretazioni errate.

Lo studio di Cagliero et al. rappresenta quindi un contributo importante per esplorare l'applicabilità di Large Language Models (LLM) nel contesto dell'educazione tecnica, dimostrando come tali modelli possano essere adattati a specifiche esigenze didattiche, quali la valutazione automatica di esercizi SQL. Il confronto tra zero-shot e few-shot learning, la valutazione della capacità del modello di distinguere tra vari tipi di errori e la verifica della comparabilità delle valutazioni con quelle dei docenti umani rappresentano punti chiave che gettano le basi per ulteriori ricerche e sviluppi nel campo della didattica assistita da intelligenza artificiale.

3.2.2 Dataset e Metodologia

Task di Semantic Parsing (Text2SQL)

Il compito di semantic parsing (SP) prevede la traduzione di una query in linguaggio naturale in codice SQL valido e eseguibile su un database relazionale. Questo studio si propone di valutare l'abilità degli studenti di un corso universitario di database nel risolvere problemi di SP, confrontando le loro soluzioni con quelle corrette e valutate da ChatGPT.

Il Benchmark MovieDB

L'analisi è stata condotta utilizzando il dataset MovieDB, che include oltre 12.000 dichiarazioni SQL generate dagli studenti di un corso universitario. Il dataset comprende anche il testo delle query in linguaggio naturale, il livello di difficoltà delle query, le soluzioni corrette, e le valutazioni date da tre docenti a un campione di risposte degli studenti. Questo dataset è stato utilizzato per addestrare e testare ChatGPT nel contesto dell'assegnazione dei voti e della fornitura di feedback.

3.2.3. Esperimento ChatGPT-as-a-SQL-Assistant

Impostazione dell'Esperimento

Gli esercizi degli studenti, già valutati dai docenti nel dataset MovieDB, sono stati sottoposti a ChatGPT per essere nuovamente valutati. Le valutazioni fornite da ChatGPT includevano un voto numerico e una spiegazione testuale, con suggerimenti su come migliorare o correggere la query. Questi risultati sono poi stati confrontati con le valutazioni assegnate da 3 diversi docenti.

Feature Analizzate

Per rispondere alle domande di ricerca, sono state raccolte e analizzate diverse feature per ciascuna submission degli studenti, tra cui:

- **Mean teachers' grade:** La media dei voti assegnati dai docenti.
- **ChatGPT grade:** Il voto assegnato da ChatGPT.
- **ChatGPT vs. humans grade difference:** La differenza tra il voto di ChatGPT e la media dei voti dei docenti.
- **ChatGPT explanation:** La spiegazione fornita da ChatGPT per la valutazione.
- **ChatGPT syntax error:** Indicazione se ChatGPT ha identificato errori sintattici.
- **ChatGPT semantic error:** Indicazione se ChatGPT ha identificato errori semantici.
- **ChatGPT evaluation:** Classificazione binaria della valutazione di ChatGPT come corretta o sbagliata.
- **ChatGPT suggestion:** La classificazione dei suggerimenti di ChatGPT come utili, non utili o mancanti.
- **True syntax errors:** Verifica se gli errori sintattici riportati da ChatGPT erano effettivamente presenti.
- **True semantic errors:** Verifica se gli errori semantici riportati da ChatGPT erano effettivamente presenti.

Definizione dei Prompt

Sono stati testati diversi prompt per ottimizzare la qualità delle valutazioni di ChatGPT, sia in contesti di zero-shot che di few-shot learning. Nella modalità zero-shot, ChatGPT veniva fornito solo con le informazioni strettamente necessarie per valutare la query, mentre nella modalità few-shot venivano forniti anche esempi di risposte corrette e i relativi voti.

Type	Prompt	Response
Zero-Shot Learning	<p>Given this database schema: (database scheme) Given the following question: (Assignment of Exercise 3) Considering that the following query would be the correct answer: (Reference statement of exercise 3) What grade from 0 to 100 would you give to the following query considering both semantics and syntax? (submitted answer 3880) Answer with a number followed by an explanation</p>	<p>I would rate the given query a 50 out of 100. Explanation: Syntax: The syntax of the query is correct. It selects the id from the person table where the year born is equal to the maximum year born in the person table. The SQL syntax is clear and follows the standard structure for such queries. Semantics: The query selects the person(s) with the maximum year born in the person table. This would retrieve the oldest person(s) in the database, which is contrary to the requirements of the problem. The query does not address the requirement to find the second youngest person(s) in the database. The query lacks the logic necessary to identify the second youngest person(s) in the database based on the birth year. Overall, while the syntax of the query is correct, it fails to fulfill the requirements of the problem and does not achieve the intended outcome. Hence, it receives a low score of 50.</p>
Few-Shot Learning	<p>Given this data base scheme: (database scheme) Given this problem: (Assignment of Exercise 3) This submission corresponds to a rating of 100/100 : (submitted answer 852) This submission corresponds to a rating of 5/100 : (submitted answer 5169) This submission corresponds to a rating of 50/100 : This submission corresponds to a rating of 30/100 : (submitted answer 6692) What rating from 0 to 100 would you give this query? (submitted answer 9391)</p>	<p>The query provided is not directly related to the problem of finding the person(s) who have worked on the largest number of distinct movies in the given database. Instead, it is selecting all columns from the "movie" table where the title is "psychi". This query doesn't address the problem statement. Rating: 0/100</p>

3.2.4 Risultati Sperimentali

Correlazione tra i Voti

La correlazione tra i voti assegnati dai docenti e quelli assegnati da ChatGPT è stata calcolata utilizzando il coefficiente di correlazione di Pearson. I risultati mostrano una forte correlazione positiva tra i due set di voti, specialmente per gli esercizi più semplici. Tuttavia, la correlazione diminuisce con l'aumentare della difficoltà degli esercizi, riflettendo la complessità crescente nel valutare correttamente esercizi più difficili.

Analisi delle Spiegazioni di ChatGPT

È stata condotta un'analisi qualitativa delle spiegazioni fornite da ChatGPT, confrontandole con le valutazioni dei docenti. I risultati indicano che ChatGPT è generalmente bravo a identificare errori esistenti, ma tende a sovrastimare la gravità degli errori, talvolta confondendo errori minori con problemi semantici più gravi. Inoltre, ChatGPT tende a fornire suggerimenti non sempre utili, focalizzandosi eccessivamente su aspetti sintattici o sulla leggibilità del codice.

Capacità di Distinzione tra Errori Sintattici e Semantici

ChatGPT è risultato molto efficace nell'identificare errori sintattici, ma spesso confonde gli errori sintattici con quelli semantici, dimostrando difficoltà nel distinguere correttamente tra i due tipi di errori.

Confronto tra Zero-Shot e Few-Shot Learning

Nel confronto tra i due approcci, la modalità zero-shot si è dimostrata altrettanto efficace, se non migliore, della modalità few-shot. Le spiegazioni fornite in modalità zero-shot sono risultate più strutturate e precise, mentre nella modalità few-shot ChatGPT tendeva a fornire spiegazioni meno informative e più sintetiche.

3.2.5 Conclusioni

Tramite l'esperimento è stato dimostrato da Cagliero et al. che ChatGPT può essere un valido strumento di supporto nell'insegnamento delle query SQL, sebbene con alcune limitazioni. I voti assegnati da ChatGPT mostrano una buona correlazione con quelli dei docenti, anche se tende a essere più severo e a sovrastimare gli errori degli studenti. Inoltre, è emerso che l'abilità di ChatGPT nel distinguere tra errori sintattici e semantici potrebbe essere migliorata. Infine, è stato evidenziato che la modalità zero-shot si è rivelata efficace e più efficiente rispetto alla modalità few-shot, suggerendo che l'adattamento dei modelli LLM per specifiche attività potrebbe non richiedere necessariamente un training estensivo.

4 Metodologia

4.1 Design della Ricerca

4.1.1 Origine del Dataset

Il dataset utilizzato per questa ricerca è stato raccolto durante un esperimento condotto nell'ambito di un corso universitario di Basi di Dati Relazionali presso l'Australian National University (ANU). Questo esperimento si è svolto nell'agosto 2018, quando 393 studenti iscritti al corso hanno iniziato a studiare il modello di basi di dati relazionale e il linguaggio SQL. L'esperimento si è svolto interamente online e ha avuto una durata di tre settimane.

Descrizione dell'Esperimento

Gli studenti sono stati invitati a partecipare attraverso una piattaforma di valutazione online, in cui dovevano completare un totale di 15 esercizi utilizzando query SQL. La piattaforma forniva un ambiente SQLite all'interno del browser degli studenti. Questo ambiente consentiva agli studenti di inserire ed eseguire query SQL, verificandone i risultati in tempo reale.

Le risposte degli studenti erano sotto forma di dichiarazioni SQL, che potevano essere eseguite direttamente all'interno della piattaforma. Quando il risultato della query eseguita coincideva con quello della query di riferimento fornita dai tutor, la piattaforma restituiva un messaggio di conferma che segnalava la correttezza della risposta.

Raccolta e Archiviazione dei Dati

Durante le interazioni degli studenti con la piattaforma di valutazione, tutte le query SQL inviate dagli studenti sono state registrate e archiviate. Complessivamente, l'esperimento ha raccolto 12.899 dichiarazioni SQL inviate dagli studenti. Queste dichiarazioni sono state successivamente utilizzate per costruire un dataset di benchmark volto a valutare diversi approcci di correzione automatica delle query SQL.

Per la creazione di questo benchmark, sono state selezionate casualmente 45 dichiarazioni SQL per ciascun esercizio. Tre assistenti didattici hanno valutato manualmente ogni dichiarazione, assegnando un punteggio basato sulla correttezza e sulla qualità della risposta. Il punteggio finale assegnato a ciascuna query è stato ottenuto come media dei punteggi forniti dai tre assistenti.

Struttura del Dataset

Il dataset raccolto è stato archiviato in un file SQLite denominato Submission.sqlite. Questo file contiene diverse tabelle che memorizzano le informazioni relative alle dichiarazioni SQL

inviare dagli studenti, i punteggi assegnati dagli assistenti e gli esercizi proposti dal tutor. Le tabelle principali del dataset sono:

1. `exercises_result`: Contiene tutte le dichiarazioni SQL inviate dagli studenti. Ogni dichiarazione è categorizzata in base al risultato dell'esecuzione della query:
 - non interpretabile: la query non è eseguibile.
 - parzialmente corretta: il risultato dell'esecuzione differisce dal risultato atteso.
 - corretta: il risultato dell'esecuzione della query coincide con il risultato atteso.

Dopo un'analisi approfondita delle query corrette, è emerso che alcuni studenti avevano formulato dichiarazioni SQL appositamente progettate per ingannare il sistema di valutazione. Ad esempio, in un esercizio, gli studenti dovevano identificare la seconda persona più giovane in un database, ma alcuni di loro, esplorando i dati, hanno scoperto direttamente l'ID della persona richiesta e hanno inviato una query SQL con quell'ID. Sebbene il risultato fosse tecnicamente corretto, non era quello che i tutor si aspettavano, e tali risposte sono state identificate come 'inganno'.

2. `exercises_benchmark`: Contiene i punteggi assegnati dagli assistenti didattici alle dichiarazioni selezionate casualmente. Ogni esercizio è stato valutato da tre assistenti, e il punteggio finale è stato calcolato come media dei punteggi assegnati.
3. `exercises_exercise`: Contiene i dettagli degli esercizi assegnati agli studenti, come il titolo, la descrizione e il livello di difficoltà.

4.1.2 Modelli di Machine Learning utilizzati

In questa sezione si esaminano i principali modelli di machine learning utilizzati per questa ricerca: ChatGPT 4.0, Gemma 2B, CodeStral, e LLama 3.1 Sonar. Ognuno di questi modelli offre caratteristiche uniche che li rendono adatti a diversi aspetti della correzione e valutazione delle query SQL.

ChatGPT 4.0

ChatGPT 4.0, sviluppato da OpenAI, rappresenta una delle versioni più avanzate di modelli di linguaggio naturale disponibili attualmente. Questo modello è stato scelto per la sua capacità di comprendere il contesto delle query SQL e di fornire feedback strutturati e coerenti.

Uno degli aspetti chiave di ChatGPT 4.0 è la sua versatilità. È in grado di risolvere query SQL complesse, identificando sia errori sintattici che semantici. Questa capacità deriva dalla sua formazione su una vasta gamma di dati, che includono non solo dati di linguaggio naturale, ma anche linguaggi di programmazione come SQL. La sua modalità zero-shot è particolarmente utile per la valutazione delle query, poiché consente di fornire risposte

accurate senza la necessità di un training specifico sulle query SQL, a differenza di molti altri modelli che richiedono una fase di addestramento più intensiva.

Nonostante la sua capacità di fornire un feedback generalmente accurato, ChatGPT 4.0 può rivelarsi eccessivamente rigoroso nel giudicare piccoli errori, come leggere imprecisioni semantiche. Questa tendenza si manifesta anche nella sua inclinazione a esagerare l'importanza degli errori sintattici, penalizzando query che, sebbene presentino minori difetti strutturali, si avvicinano comunque alla soluzione corretta.

Gemma 2B

Gemma 2B è un modello di machine learning più specializzato nel campo della correzione automatica di codice, inclusi i linguaggi di programmazione strutturati come SQL. Con una dimensione di 2 miliardi di parametri, Gemma 2B è stato sviluppato specificamente per gestire scenari di correzione automatizzata, offrendo una capacità di generalizzazione superiore rispetto ad altri modelli meno specializzati.

La sua peculiarità principale risiede nella capacità di gestire errori complessi nelle query SQL, come quelli che coinvolgono condizioni multiple o subquery annidate. Gemma 2B è anche altamente efficiente nel riconoscere errori logici all'interno delle query, come ad esempio il mancato utilizzo di join appropriati o l'errata applicazione di operatori logici.

Un punto debole di questo modello è però rappresentato dalla sua difficoltà nel gestire errori semantici. Sebbene eccella nell'individuare problemi logici e sintattici, fatica maggiormente a rilevare errori che compromettono la coerenza interna della query, senza influenzarne direttamente la struttura. Questo può portare a non identificare correttamente situazioni in cui una query, pur funzionando, non risponde all'intento originale del programmatore.

CodeStral

CodeStral è un altro modello significativo utilizzato per la valutazione delle query SQL. Progettato con un'architettura che privilegia l'efficienza computazionale, CodeStral eccelle nel rilevare errori di sintassi e ottimizzazioni delle query.

Questo modello è stato sviluppato per essere particolarmente adatto a scenari in cui la ridondanza del codice e l'efficienza delle query giocano un ruolo cruciale. Per esempio, è molto abile nel suggerire alternative più efficienti a subquery non necessarie o nell'eliminare l'uso improprio di operatori come DISTINCT o funzioni aggregate non necessarie. Ciò lo rende una scelta eccellente per l'ottimizzazione del codice SQL nei contesti accademici e professionali.

Tuttavia, CodeStral ha alcune limitazioni nel riconoscimento di errori semantici complessi. La sua valutazione si concentra principalmente sulla correttezza strutturale e sulla performance delle query, il che può portare a una sovrastima di query che sono formalmente corrette ma concettualmente errate. Ad esempio, se una query utilizza una subquery superflua che non influisce negativamente sulla prestazione, CodeStral potrebbe non segnalarla come problematica.

LLama 3.1 Sonar

LLama 3.1 Sonar è un modello di recente introduzione, appartenente alla famiglia di LLM basati sul meta-apprendimento. Si tratta di un modello altamente avanzato, con un forte focus sulla comprensione contestuale e sull'adattamento rapido a nuovi task. LLama 3.1 Sonar si distingue per la sua capacità di apprendere dai contesti forniti dagli esercizi SQL, offrendo risposte personalizzate e adattative in base agli input ricevuti.

Questo modello è particolarmente abile nel rilevamento degli errori semantici all'interno delle query SQL, riconoscendo quando una query, pur essendo corretta dal punto di vista esecutivo, non restituisce il risultato previsto dall'intenzione del programmatore. Inoltre, LLama 3.1 Sonar ha dimostrato una forte capacità di rilevare pattern di inganno, come le query costruite in modo da eludere la logica richiesta dall'esercizio.

Un limite di LLama 3.1 Sonar è la sua complessità computazionale, che lo rende meno efficiente rispetto a modelli più leggeri come CodeStral. Inoltre, il modello richiede un training e un adattamento maggiore per raggiungere prestazioni ottimali in nuovi contesti, risultando quindi meno efficiente nel breve periodo.

In sintesi, ogni modello di machine learning utilizzato per la valutazione delle query SQL in questa ricerca presenta punti di forza e debolezze distinti. **ChatGPT 4.0** si dimostra versatile e accurato, ma tende a essere troppo severo con errori minori. **Gemma 2B** eccelle nella gestione degli errori logici e complessi, ma è meno abile nel riconoscimento di errori semantici. **CodeStral** è ideale per ottimizzazioni ed efficienza, ma potrebbe trascurare alcuni aspetti concettuali delle query. **LLama 3.1 Sonar**, infine, si distingue per la sua capacità di adattamento e rilevamento degli inganni, ma richiede risorse computazionali elevate e una fase di training più intensa.

L'utilizzo congiunto di questi modelli offre un approccio robusto e completo alla valutazione delle query SQL, permettendo di coprire un ampio spettro di errori, ottimizzazioni e miglioramenti del codice, contribuendo così a un sistema di valutazione automatizzato che può avvicinarsi alle capacità di correzione umana.

4.2 Descrizione degli errori utilizzati

La classificazione degli errori nelle query SQL è stata basata sulle classificazioni proposte da ricerche precedenti. In particolare, è stato considerato il framework degli errori definito dagli studi di Taipalus et al., arricchito successivamente con alcuni errori individuati da Ahadi et al. Di seguito è riportata la spiegazione degli errori utilizzati nella ricerca.

4.2.1 Errori sintattici

Gli errori sintattici si verificano quando la struttura della query non rispetta le regole grammaticali del linguaggio. Questi errori possono derivare da una varietà di cause, come l'uso errato delle parole chiave SQL o la cattiva formattazione della query. Quando si riscontrano errori sintattici, il sistema restituisce un messaggio di errore che indica la posizione e la natura del problema, aiutando il programmatore a correggerlo per assicurare l'esecuzione corretta della query

SYN-1 Definizione ambigua dei database

Errore causato da una errata o mancata assegnazione dei qualificatori ai database nella FROM. Ciò è in particolare dovuto a quando ho bisogno di effettuare una self JOIN, avendo quindi la necessità di definire due tabelle identiche. IL DBMS non riesce a collegare il database a ciò che è trattato nella query

SYN-2 Utilizzo di database non definiti

Questo tipo di errore si riscontra quando una query fa riferimento a database o tabelle che non esistono nel sistema. È spesso causato da errori grammaticali nella clausola FROM, come un'errata ortografia del nome del database o del nome della tabella. Quando questo errore si verifica, il sistema restituirà un messaggio di errore indicando l'incapacità di trovare il database specificato.

SYN-3 Operazione non supportata per tipo di dato

Si ha a causa dell'utilizzo di un operatore su colonne con un tipo di dato non adatto all'operazione, ad esempio cercando di eseguire una somma su una colonna di tipo String. Il DBMS non è in grado di eseguire l'operazione richiesta, generando un errore sintattico

SYN-4 Utilizzo errato di funzioni aggregate

Questo tipo di errore si verifica quando le funzioni di aggregazione vengono utilizzate in modo non corretto all'interno di una query. Ad esempio, se vengono posizionate al di fuori della clausola SELECT o HAVING, genereranno un errore sintattico. Per correggere questo errore, è necessario spostare correttamente le funzioni di aggregazione all'interno della clausola appropriata.

SYN-5 Raggruppamento errato o insufficiente

Questo errore è riscontrato quando una query che utilizza funzioni di aggregazione non è stata raggruppata correttamente utilizzando la clausola GROUP BY, causando un errore che non permette l'esecuzione del codice. Senza una corretta clausola GROUP BY, il sistema non può determinare come raggruppare i risultati, causando un errore sintattico. Questo errore è inoltre dovuto a quando nella SELECT sono inseriti degli attributi non presenti nella GROUP BY

SYN-6 Errori di sintassi comuni

Questo tipo di errore si verifica quando ci sono errori grammaticali nelle keyword o nella struttura della sintassi SQL. Gli errori comuni includono la mancanza di parentesi o la scrittura errata delle parole chiave.

4.2.2 Errori semantici

Gli errori semantici si verificano quando la logica della query non corrisponde all'intenzione del programmatore o produce risultati non desiderati. Questi errori possono derivare da una comprensione inaccurata dei dati o della struttura del database o da una selezione errata delle clausole SQL. Risolvere gli errori semantici richiede una valutazione approfondita della query per identificare e correggere le discrepanze tra l'intenzione del programmatore e il comportamento effettivo della query.

SEM-1 Espressioni inconsistenti

Quando una query produce una tabella risultato vuota o contenente tutte le righe del database, rendendo i risultati non pertinenti o non utili. Ciò può essere causato da condizioni nel WHERE errate, che causano l'eliminazione di tutte le righe dalla Result Table o da una logica di aggregazione inappropriata.

SEM-2 Join inconsistente

Questo errore si verifica quando il JOIN tra tabelle viene eseguito in modo errato, causando la creazione di una tabella risultato vuota o contenente tutte le righe delle tabelle coinvolte. Questo non corrisponde all'intenzione del programmatore e può portare a risultati errati.

SEM-3 Mancanza di JOIN

Errore causato dalla mancanza di un JOIN necessario tra le tabelle coinvolte. Questo può portare a risultati incompleti o errati poiché le relazioni tra i dati non vengono adeguatamente stabilite, compromettendo la completezza e la correttezza della query.

SEM-4 Righe duplicate nella Result Table

Questo tipo di errore si ha quando la tabella risultato contiene righe identiche tra loro, introducendo ridondanza nei risultati. Utilizzare la clausola DISTINCT nella SELECT può risolvere questo problema, eliminando le righe duplicate dalla tabella risultato. Questo errore comprende anche la mancanza di una DISTINCT nelle funzioni aggregate quando richiesto

SEM-5 Colonne ridondanti o insufficienti nella Result Table

Questo errore si verifica quando la tabella risultato contiene colonne non necessarie o ridondanti, che non contribuiscono alla comprensione o all'utilità dei risultati. È importante identificare e rimuovere le colonne ridondanti per ottimizzare le prestazioni e migliorare la chiarezza dei risultati della query. Questo errore comprende anche quando viene mostrata la colonna sbagliata

SEM-6 Errore nella selezione delle tabelle

Si ha quando le tabelle selezionate in una query SQL non sono appropriate per soddisfare la richiesta specificata. Questo può derivare dalla scelta di tabelle errate o dalla mancanza di una o più tabelle necessarie per ottenere i risultati desiderati. Correggere questo errore richiede una valutazione attenta delle tabelle coinvolte e una modifica appropriata della query per garantire che le tabelle siano pertinenti al contesto della richiesta.

4.2.3 Errori Logici

Gli errori logici in SQL si verificano quando la logica della query produce risultati errati a causa di un errore nel processo di ragionamento o di un'errata interpretazione dei dati. È fondamentale avere una solida comprensione dei requisiti del problema e della struttura del database per prevenire gli errori logici durante lo sviluppo delle query SQL.

LOG-1 Errore negli operatori, tra cui Nesting Errors

Questo tipo di errore riguarda l'uso improprio degli operatori booleani o la disposizione errata delle parentesi nelle espressioni logiche. Ad esempio, l'omissione o l'errato utilizzo di

operatori come NOT, AND e OR può portare a risultati errati nella query. Inoltre, errori di nidificazione delle parentesi possono influenzare l'ordine di valutazione delle condizioni e causare ambiguità nella logica della query.

LOG-2 Errore nel JOIN

Gli errori di JOIN si verificano quando vengono selezionate o combinate le tabelle in modo improprio, risultando in relazioni errate o risultati incoerenti. Ad esempio, un JOIN può essere applicato a tabelle non correlate o utilizzare condizioni di join errate, portando a risultati non desiderati o alla mancanza di dati rilevanti nella tabella risultato. Ciò comprende anche i JOIN tra tabelle effettuati erroneamente senza l'utilizzo di chiavi

LOG-3 Errore nell'utilizzo delle espressioni

Gli errori di espressione sono riscontrati quando le condizioni logiche o le operazioni aritmetiche sono formulate in modo improprio. Ad esempio, l'omissione di condizioni necessarie o l'utilizzo di colonne errate nelle espressioni possono compromettere l'integrità della query e produrre risultati inaccurati. Ciò comprende anche la mancanza di espressioni richieste dal testo

LOG-4 Errori nelle funzioni di aggregazione

Gli errori nelle funzioni di aggregazione in SQL si verificano quando le funzioni vengono utilizzate impropriamente o non necessariamente all'interno della query. Inoltre, l'utilizzo di DISTINCT come parametro per le funzioni di aggregazione in situazioni in cui non è applicabile può causare una valutazione errata dei dati aggregati.

LOG-5 Subquery errate o mancanti

Gli errori o l'assenza di subquery indicano un problema nella struttura logica della query SQL, dove una subquery è necessaria ma non è stata inclusa o è stata formulata in modo errato. Questo può portare a risultati inesatti o incompleti. È importante identificarle e correggerle correttamente per garantire la correttezza e la completezza dei risultati della query principale.

4.2.4 Punti di Attenzione

Di seguito sono riportate una serie di punti di attenzione del codice che, pur non essendo errori, rendono il codice meno efficiente e leggibile, introducendo ridondanza.

COM-1 DISTINCT o funzioni aggregate in SELECT non necessarie

Questo errore si verifica quando vengono utilizzate le clausole DISTINCT o funzioni di aggregazione come COUNT, SUM, AVG, ecc. nella SELECT senza una reale necessità. Questo può appesantire la query senza migliorare significativamente i risultati.

COM-2 JOIN non necessario

Un JOIN non necessario si verifica quando vengono unite tabelle che non sono pertinenti alla query o non contribuiscono alla logica della selezione dei dati. Questo aggiunge complessità alla query senza motivo. Questo tipo di complicazione comprende soltanto l'utilizzo di JOIN superflui e non la scelta di un tipo di JOIN più complicato, per lasciare allo studente la possibilità di scegliere il più adeguato alle sue esigenze

COM-3 Subqueries non necessarie

Le subquery non necessarie aggiungono complessità e possono rallentare le prestazioni della query senza aggiungere valore. Si presentano quando una query annidata può essere evitata utilizzando alternative più efficienti come JOINS o clausole WHERE.

COM-4 Utilizzo nel FROM di tabelle non necessarie

L'inclusione di tabelle non necessarie nel FROM introduce ridondanza e complessità nella query. Questo può causare confusione e rendere più difficile la comprensione della logica di selezione dei dati.

COM-5 Utilizzo di condizioni superflue nel WHERE

Le condizioni superflue nel WHERE possono rendere la query più difficile da interpretare. Questo si verifica quando vengono utilizzate condizioni che non influenzano i risultati della query.

COM-6 GROUP BY o HAVING non necessario

L'utilizzo di clausole GROUP BY o HAVING non necessarie aggiunge complessità alla query senza una reale utilità. Questo si verifica quando non ci sono aggregazioni o condizioni di filtro che richiedono il raggruppamento dei dati. Questo punto di attenzione non include i casi in cui l'utilizzo o non utilizzo di una GROUP BY causa la non esecuzione del codice

COM-7 ORDER BY non necessario

L'inclusione di clausole ORDER BY non necessarie aggiunge complessità e può influire negativamente sulle prestazioni della query. Questo si verifica quando non c'è una reale necessità di ordinare i risultati della query.

4.2.5 INV Query Invalida

Una soluzione è considerata invalida se, pur fornendo la soluzione corretta, è strutturata in modo tale da eludere la reale richiesta dell'esercizio, attraverso l'applicazione di conoscenze derivate da una valutazione specifica dei dati nel dataset (es. WHERE yearn_born = '1971' per trovare la persona più anziana nel database, dopo aver osservato che l'anno di nascita minimo è 1971).

4.3 Descrizione del Dataset

Il dataset utilizzato nella ricerca è suddiviso in due macrocategorie principali: una relativa alle domande dell'esercitazione e alle risposte fornite dagli studenti, ed un'altra contenente i dati utilizzati dagli studenti per rispondere agli esercizi assegnati.

4.3.1 Dati delle Esercitazioni

- Exercises_exercises: contiene gli esercizi assegnati agli studenti, con un identificativo univoco (id), il testo della richiesta, il livello di difficoltà e una possibile risposta di riferimento.
- Exercises_result: registra tutte le risposte inserite dagli studenti, incluso l'ID dello studente, l'ID dell'esercizio, il testo della query SQL inviata, il timestamp della risposta e un flag che indica se la risposta è corretta o meno.
- Exercises_benchmark: tabella che memorizza la relazione tra l'ID della risposta dello studente (submission_id) e il punteggio assegnato dal docente (tutor) per la valutazione.

Struttura delle Tabelle:

- exercises_exercise(id, title, preamble, difficulty, ref)
Chiave primaria: id
- exercises_result(submission_id, submitted_answer, submission_time, exercise_id, is_correct, student_id, category)
Chiave primaria: submission_id

- **exercises_benchmark**(submission_id, grade, tutor)
Chiave primaria: submission_id

4.3.2 Dati di Supporto agli Esercizi

Di seguito sono riportati i dataset utilizzati dagli studenti per risolvere gli esercizi assegnati:

Actor_award(title, production_year, description, award_name, year_of_award, category, result)

keys: title, production_year

Appearance(title, production_year, description, scene_no)

keys: title, production_year

Crew_Award(id, title, production_year, award_name, year_of_award, category, result)

primary_key: id

Director(id, title, production_year)

primary_key: id

Director_Award(title, production_year, award_name, year_of_award, category, result)

keys: title, production_year

Movie(title, production_year, country, run_time, major_genre)

keys: title, production_year

Movie_Award(title, production_year, award_name, year_of_award, category, result)

keys: title, production_year

Person(id, first_name, last_name, year_born)

primary_key: id

Restriction(title, production_year, description, country)

keys: title, production_year

Role(id, title, production_year, description, credits)

primary_key: id

Scene(title, production_year, scene_no, description)

keys: title, production_year

Writer(id, title, production_year, credits)

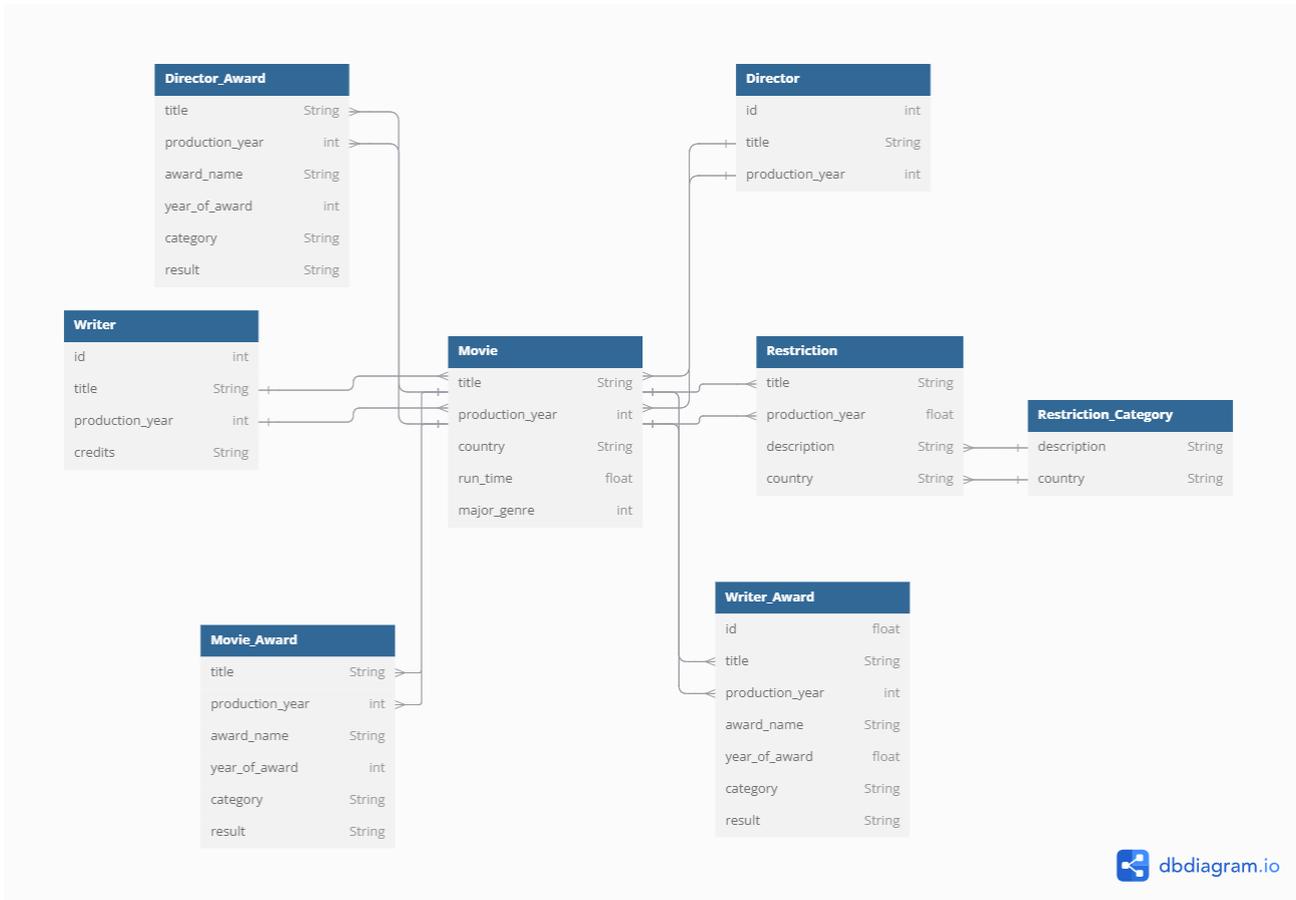
primary_key: id

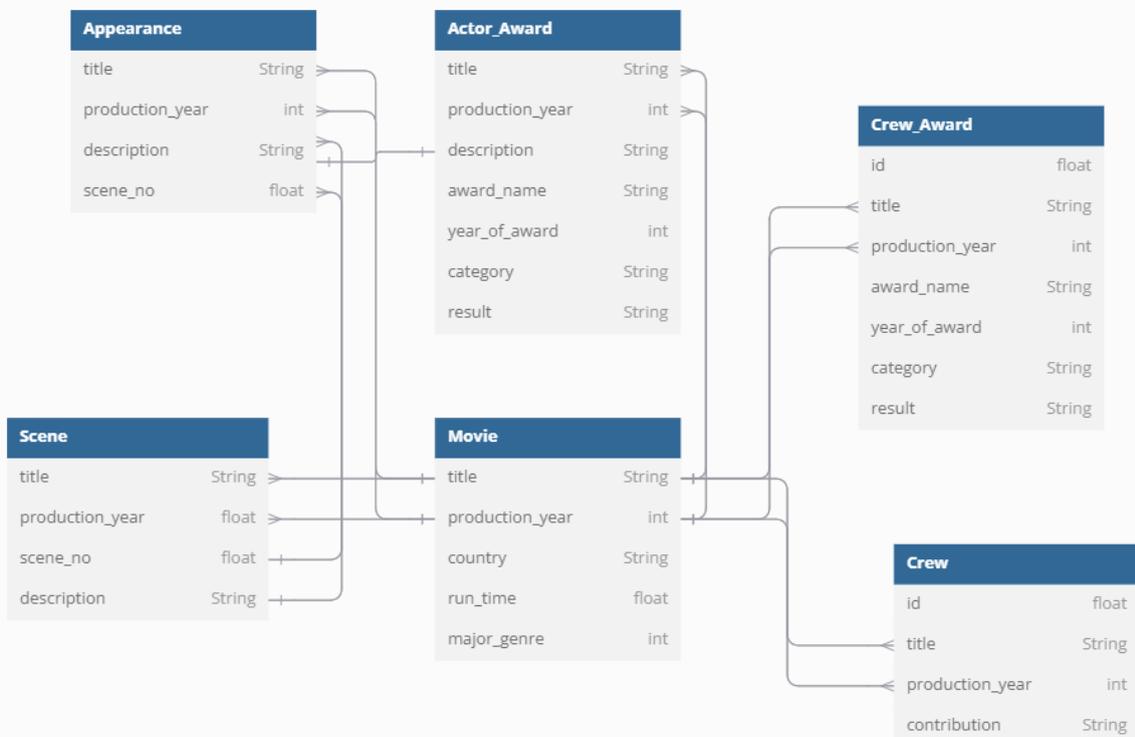
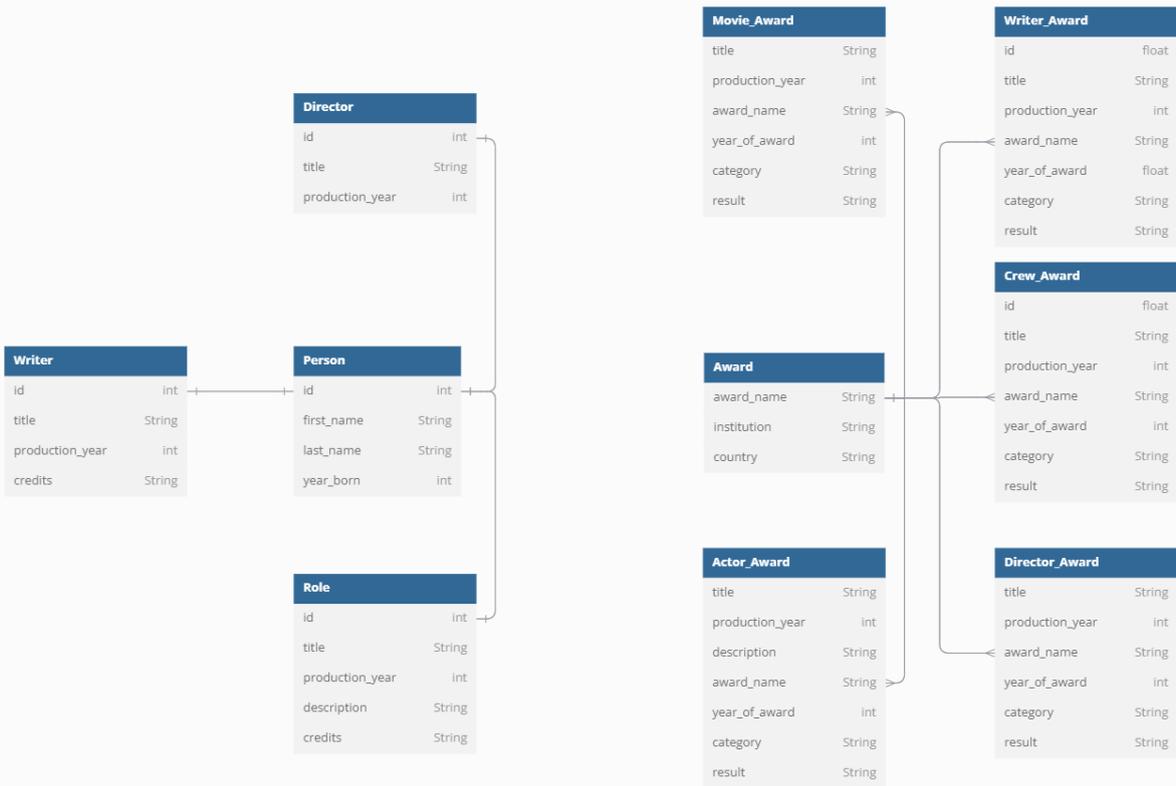
Writer_Award(id, title, production_year, award_name, year_of_award, category, result)

primary_key: id

Diagrammi ER del Database di Supporto agli Esercizi

Per facilitare la lettura, il diagramma Entità-Relazione è stato suddiviso in tre sotto-diagrammi.





4.3.3 Selezione del Sub-Dataset

Dopo aver classificato tutte le query, si è proceduto con la definizione delle risposte da utilizzare per la ricerca, includendo sia le query da valutare con i modelli di ML che quelle di esempio per il metodo few-shot learning.

Per fare ciò, sono stati prima individuati i 3 errori più frequenti per ciascun esercizio, in modo da identificare le query di esempio per il metodo few-shot learning. Successivamente, è stato definito un dataset di test adeguato, che include, laddove possibile, almeno 4 query per ogni tipo di errore, mantenendo un numero bilanciato di query per esercizio.

A seguito di questa selezione, si è deciso di escludere l'esercizio numero 9 dal dataset scelto, poiché non conteneva errori non già presenti negli altri esercizi.

Di seguito sono riportati il numero di query nel sub-Dataset contenente ciascun tipo di errore.

Tipo Errore	SYN-1	SYN-2	SYN-3	SYN-4	SYN-5	SYN-6
Quantità nelle Query	2	5	2	4	5	5

Tipo Errore	SEM-1	SEM -2	SEM -3	SEM -4	SEM -5	SEM -6
Quantità nelle Query	5	1	5	4	5	5

Tipo Errore	LOG-1	LOG -2	LOG -3	LOG -4	LOG -5	LOG -6
Quantità nelle Query	2	5	5	5	5	5

Tipo Errore	COM-1	COM-2	COM-3	COM -4	COM -5	COM -6	COM-7
Quantità nelle Query	4	5	5	5	1	2	2

4.4 Addestramento dei Modelli

4.4.1 Implementazione dei Modelli

I modelli di linguaggio di grandi dimensioni (LLM) sono stati implementati per analizzare e classificare le risposte fornite dagli studenti. La loro implementazione è stata suddivisa in due fasi principali: la prima fase si concentra sull'inserimento dei prompt relativi agli errori, e la seconda sull'applicazione di tecniche di valutazione delle query attraverso specifici prompt, differenziati per i metodi di few-shot e zero-shot learning.

4.4.2 Prompt degli Errori

Il primo passo dell'implementazione ha coinvolto l'inserimento del "prompt degli errori". Questo prompt è stato progettato per fornire al modello una comprensione chiara delle possibili tipologie di errori che possono essere riscontrati nelle query SQL.

Di seguito il prompt testuale utilizzato per l'addestramento dei modelli:

4.4.3 Prompt degli Esercizi

La fase successiva ha riguardato l'applicazione del "prompt degli esercizi", ovvero i modelli sono stati chiamati a valutare le query sottoposte dagli studenti, utilizzando diversi approcci di prompt a seconda del tipo di addestramento adottato.

- Consigli aggiuntivi: in alcuni casi, ai modelli sono stati forniti prompt che includevano consigli aggiuntivi sugli errori riscontrati, suggerendo come correggerli o quali parti della query necessitavano di maggiore attenzione. Questo approccio si è rivelato utile soprattutto nelle prime fasi di few-shot learning, dove il modello ha avuto bisogno di ulteriori indicazioni per migliorare la precisione della valutazione.

Confronto tra Few-Shot e Zero-Shot Learning

Durante l'implementazione, sono stati utilizzati due principali metodi di apprendimento: few-shot e zero-shot learning.

Few-Shot Learning: In questo metodo, al modello sono stati forniti alcuni esempi di errori frequenti nell'esercizio.

Esempio prompt Few-Shot (Esercizio 1):

You are an expert SQL evaluator. Your task is to identify any Syntactic or Semantic errors, as well as Logic errors and Complications in the query

Given the following database schema:

```
Person(id, first_name, last_name, year_born)
```

```
primary_key: id
```

```
Writer(id, title, production_year, credits)
```

```
primary_key: id
```

Given the following question: How many writers were born in 1935?

Considering the SQL errors previously explained

Considering that the following SQL query is the correct answer: `SELECT count(*) FROM person p WHERE EXISTS (SELECT * FROM writer w WHERE w.id = p.id AND p.year_born = 1935);`

What grade from 0 to 100 would you give to the following SQL query?

{Answer}

These are the most popular errors found in this question, followed by a query containing them

SEM-4 Duplicate Rows in Result Table

Example Query: `select count(year_born) from person, writer where person.id = writer.id and person.year_born = 1935;`

LOG-2 Join Error

Example Query: `SELECT COUNT(*) FROM PERSON, WRITER WHERE PERSON.id=WRITER.id AND PERSON.year_born=1935 GROUP BY PERSON.id;`

When evaluating the SQL query, consider the detailed error types provided before.

Answer by listing the errors in the query, a grade followed by an explanation for it.

NOTE: The query can also be correct, not containing any errors

Zero-Shot Learning: In questo approccio, il modello ha affrontato direttamente le query senza alcun esempio preliminare.

Esempio prompt Zero-Shot (Esercizio 1):

You are an expert SQL evaluator. Your task is to identify any Syntactic or Semantic errors, as well as Logic errors and Complications in the query

Given the following database schema:

Person(id, first_name, last_name, year_born)

primary_key: id

Writer(id, title, production_year, credits)

primary_key: id

Given the following question: How many writers were born in 1935?

Considering the SQL errors previously explained

Considering that the following SQL query is the correct answer: `SELECT count(*) FROM person p WHERE EXISTS (SELECT * FROM writer w WHERE w.id = p.id AND p.year_born = 1935);`

What grade from 0 to 100 would you give to the following SQL query?

{Answer}

When evaluating the SQL query, consider the detailed error types provided before.

Answer by listing the errors in the query, a grade followed by an explanation for it.

NOTE: The query can also be correct, not containing any errors

4.4.4 Interventi Correttivi

In caso di necessità (ad esempio, in presenza di loop nelle risposte), sono stati utilizzati dei prompt correttivi per indicare al modello quali errori stava commettendo nella valutazione, come ripetere eccessivamente gli stessi errori o basarsi solo sugli errori più frequenti (nel caso del few-shot learning).

4.5 Tecniche di Classificazione degli Errori

Dopo aver eseguito l'istruzione dei modelli con la valutazione delle query degli studenti, si è proceduto alla registrazione dei risultati ottenuti per ciascun modello. Per ogni risposta generata, sono stati monitorati vari fattori:

- Errori identificati dal Large Language Model.
- Voto assegnato dal LLM (confrontato con il voto medio assegnato dai docenti).
- Errori correttamente identificati.
- Errori non identificati.
- Errori identificati erroneamente.

Successivamente, in base ai risultati ottenuti, sono stati calcolati diversi coefficienti:

- **Recall:** rapporto tra il numero di errori correttamente identificati ed il totale degli errori effettivamente presenti nelle query
- **Precision:** rapporto tra il numero di errori correttamente identificati ed il totale degli errori identificati dal modello
- **F-Score:** media armonica della Precision e del Recall dei modelli
- **Coefficiente di Pearson:** misura la correlazione lineare tra due variabili. Il suo valore varia tra -1 e 1, dove 1 indica una correlazione positiva perfetta, -1 una correlazione

negativa perfetta e 0 assenza di correlazione. Questo coefficiente viene utilizzato per verificare l'esistenza di una relazione lineare tra due serie di dati.

- $xy = xyxy$
- **Errore Assoluto Medio (MAE):** è una misura della precisione di un modello, calcolata come la media delle differenze assolute tra i valori predetti e quelli osservati. Espressa nella stessa unità dei dati, fornisce un'indicazione chiara dell'errore medio commesso dal modello. Il MAE è utile per valutare quanto le predizioni si discostino dai valori reali.
 - $MAE = \frac{1}{n} \sum_{i=1}^n |e_i|$

5. Analisi dei Risultati

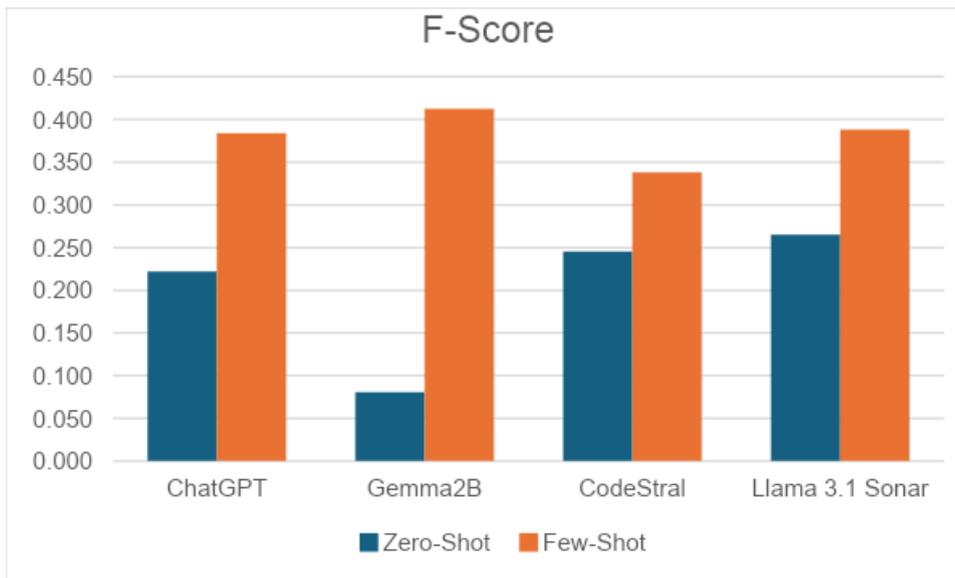
5.1 Prestazioni Generali dei Modelli

Di seguito sono riportati i risultati ottenuti dalla sperimentazione con i modelli di Machine Learning, che evidenziano le differenze nelle prestazioni tra le varie configurazioni di addestramento (zero-shot e few-shot learning) e i modelli stessi.

Modelli	Recall	Precision	F-Score	Pearson Coefficient	Mean Absolute Error(MAE)
ChatGPT Fine	0,500	0,312	0,384	0,819202688	16,00578947
ChatGPT Coarse	0,291	0,180	0,222	0,774210019	14,33881356
Gemma2B Fine	0,329	0,553	0,413	0,50885789	19,85912281
Gemma 2B Coarse	0,075	0,088	0,081	0,382759766	22,95055556
CodeStral Fine	0,336	0,340	0,338	0,577908496	20,42269231
CodeStral Coarse	0,241	0,250	0,245	0,519111476	26,49326923
Llama 3.1 Sonar Fine	0,435	0,351	0,388	0,764082896	18,87693878
Llama 3.1 Sonar Coarse	0,324	0,224	0,265	0,607774086	22,31772727

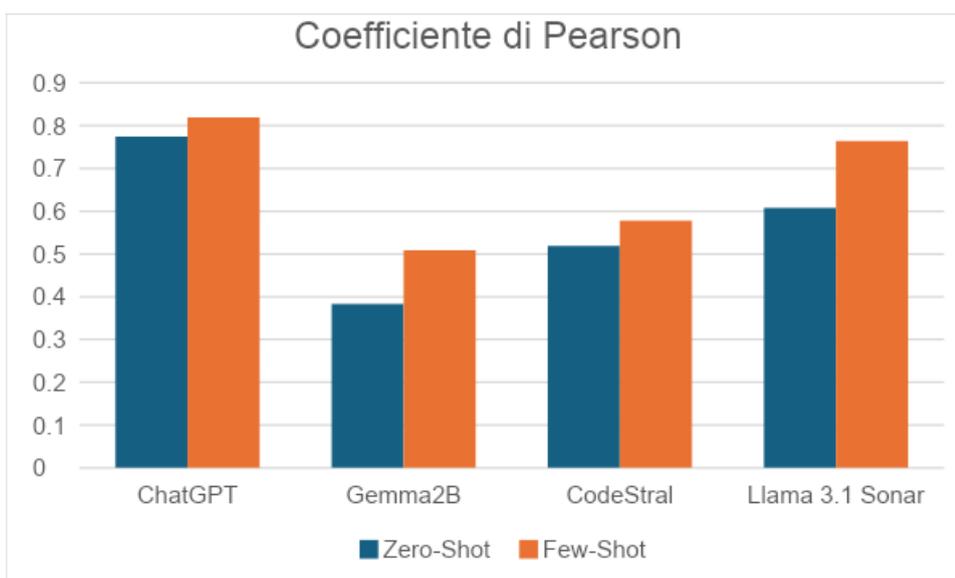
Dalla tabella si evince chiaramente che l'addestramento di tipo *few-shot* porta a un miglioramento delle prestazioni in quasi tutti i modelli. In particolare, i modelli che hanno ricevuto un addestramento più dettagliato riescono a valutare le query in modo più accurato sia in termini di voti assegnati che di errori analizzati, come dimostrato da una maggiore Precision e da valori più elevati del coefficiente di Pearson.

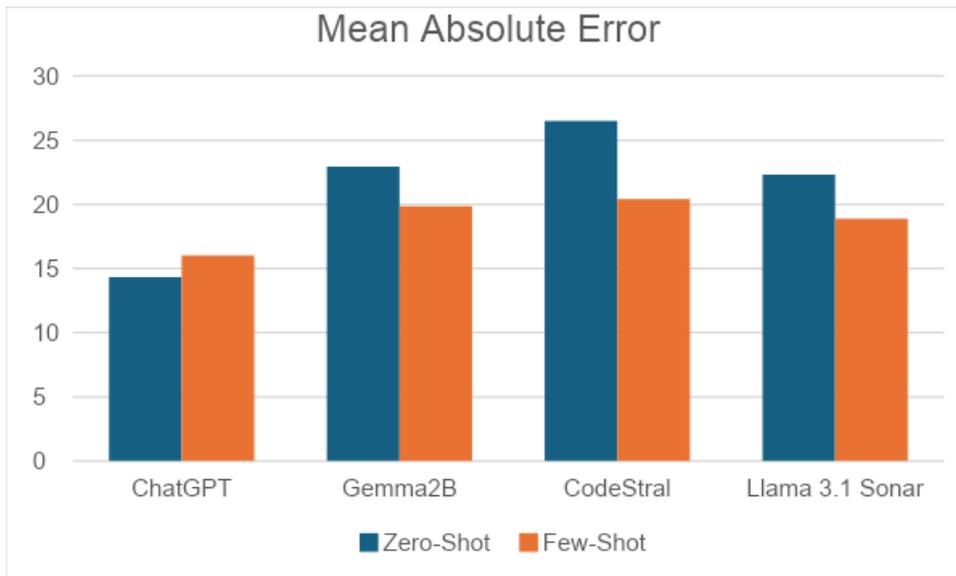
5.1.1 Confronto nel riconoscimento degli errori



Uno dei principali vantaggi dell'approccio *few-shot learning* è una maggiore accuratezza nel riconoscimento degli errori all'interno delle query SQL. Questo è evidenziato da un significativo aumento dell'F-Score, che rappresenta la media armonica di Precision e Recall. In particolare, i modelli Gemma2B e ChatGPT mostrano un miglioramento considerevole del loro F-Score con l'utilizzo del few-shot. L'aumento non è invece altrettanto evidente per i modelli CodeStral e Llama 3.1 Sonar, dove il progresso in termini di riconoscimento degli errori risulta più moderato.

5.1.2 Confronto nell'Accuratezza dei Voti





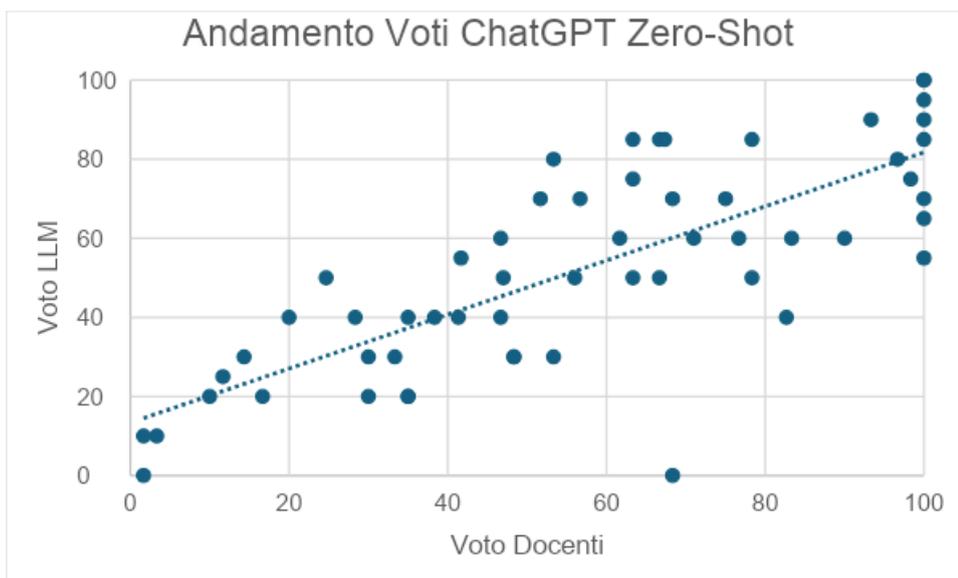
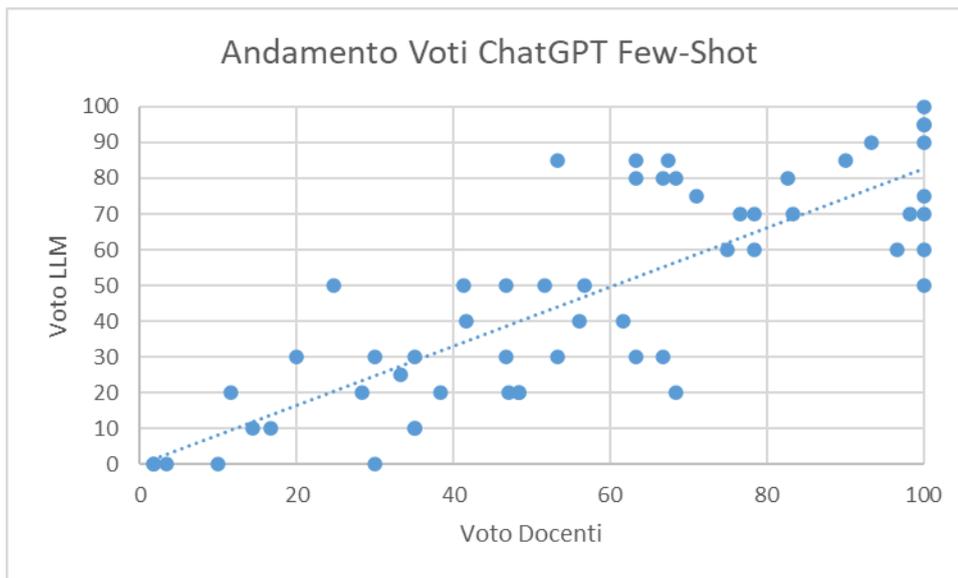
Esaminando i risultati in termini di coefficiente di Pearson e Mean Absolute Error (MAE), si nota che i modelli addestrati con l'approccio few-shot tendono a migliorare nella capacità di assegnare voti più precisi. L'aumento del coefficiente di Pearson suggerisce una correlazione più stretta tra i voti assegnati dai modelli e quelli forniti dai docenti, dimostrando un miglioramento nell'accuratezza generale. Tuttavia, l'incremento del MAE, in alcuni casi, può indicare che i modelli, pur essendo più accurati nella maggior parte delle query, tendono a commettere errori più significativi quando si verificano delle deviazioni.

5.2 Analisi Specifica dei Modelli

Per ciascuno dei modelli analizzati sono stati considerati diversi fattori per determinarne le prestazioni e i casi d'uso preferiti:

- andamento dei voti, confrontando la tendenza a fornire voti bassi o alti tra i modelli zero-shot e few-shot;
- facilità con cui il modello entra in loop di risposte identiche, richiedendo quindi interventi correttivi (come prompt di suggerimento) da parte dell'utilizzatore;
- differenze nel comportamento tra il tipo di addestramento zero-shot e quello few-shot, valutando di conseguenza la necessità di quest'ultimo.

5.2.1 ChatGPT



Durante la fase di sperimentazione, il modello ChatGPT ha mostrato sia punti di forza che debolezze nel processo di valutazione delle query SQL. Il comportamento del modello è stato analizzato in entrambe le modalità di addestramento, *few-shot* e *zero-shot*, e confrontato con le valutazioni dei docenti.

Difficoltà nell'Identificare le Risposte Completamente Corrette

Uno dei principali limiti riscontrati è stata la difficoltà del modello nell'identificare le risposte completamente corrette, ossia nel riconoscere quando una query fosse perfettamente esatta e meritevole di un punteggio pieno (100). Questo ha portato a una tendenza del modello a sottostimare le prestazioni degli studenti, soprattutto quando si trattava di attribuire il massimo punteggio. Questo comportamento potrebbe essere attribuito alla tendenza del modello a concentrarsi su minimi dettagli strutturali o semantici, che, pur non compromettendo il risultato della query, venivano percepiti come errori.

Resistenza alla Ripetizione delle Risposte

Un aspetto positivo di ChatGPT, in particolare nella modalità *zero-shot learning*, è stata la sua capacità di non ripetere le ultime risposte corrette o di imitare risposte precedenti. A differenza di altri modelli testati, ChatGPT ha mostrato una maggiore diversificazione nelle risposte e una minore tendenza a basarsi sugli output precedenti. Questo ha garantito una valutazione più dinamica e meno meccanica delle query SQL, un vantaggio significativo soprattutto in contesti in cui le risposte potrebbero essere notevolmente diverse tra loro.

Tendenza ad Assegnare Voti Inferiori

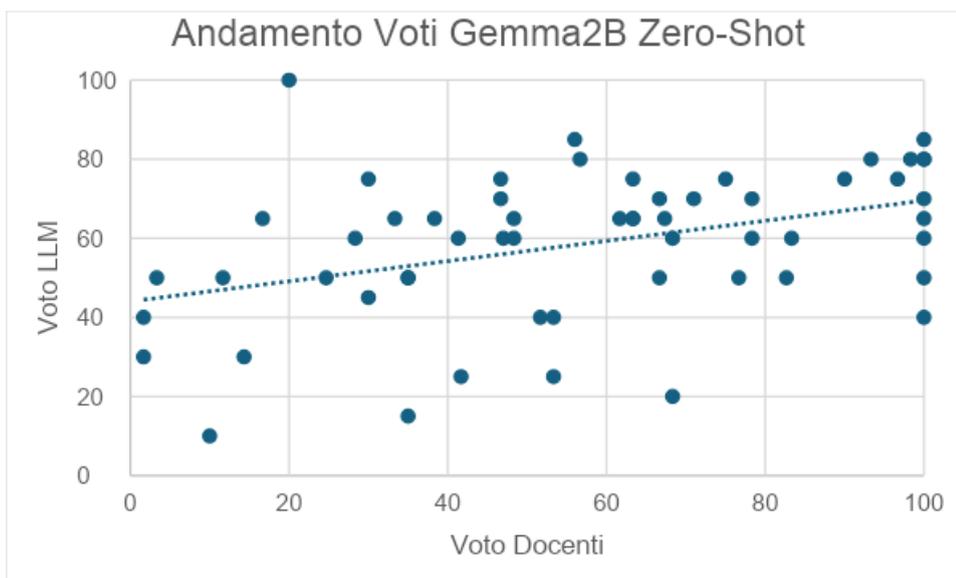
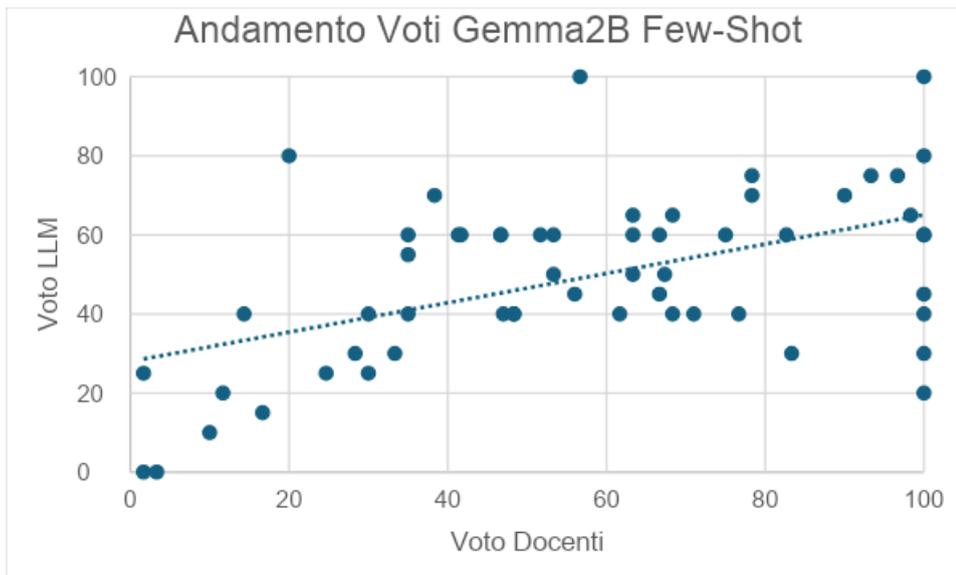
Durante la fase di sperimentazione, il modello si è rivelato generalmente più conservativo rispetto ai voti assegnati dai docenti, con una tendenza ad essere più severo nelle valutazioni, specialmente per voti alti. Ciò significa che, rispetto ai docenti, ChatGPT tendeva a penalizzare di più le query, anche in presenza di piccoli errori, generando un divario maggiore man mano che il punteggio assegnato dai docenti cresceva. Tuttavia, per le valutazioni più basse, la differenza rispetto ai voti dei docenti risultava meno marcata. Questo riflette la sua maggiore sensibilità agli errori che, sebbene minori, lo portavano a non riconoscere pienamente la qualità di alcune risposte.

Confronto tra Few-Shot e Zero-Shot training

Confrontando le modalità di addestramento *few-shot* e *zero-shot*, si è osservato un comportamento interessante. Il modello addestrato in modalità *zero-shot* tendeva a essere leggermente più generoso nelle valutazioni quando i voti erano bassi. Questo può essere spiegato dal fatto che il *zero-shot training* si basa su una comprensione più ampia e meno dettagliata delle query SQL, portando a valutazioni meno severe sugli errori commessi. Di conseguenza, mentre il modello in *few-shot* penalizzava maggiormente gli errori minori, in *zero-shot* mostrava un approccio più indulgente verso le risposte parzialmente corrette.

In conclusione, ChatGPT ha mostrato una buona capacità di valutazione delle query SQL, specialmente nella sua forma *fine-tuned*, pur con alcune limitazioni legate alla sovrastima degli errori minori e alla difficoltà di assegnare voti pieni.

5.2.2 Gemma 2B



Il modello Gemma2B ha dimostrato prestazioni peculiari rispetto agli altri modelli analizzati durante la fase di sperimentazione. Sebbene abbia presentato alcune caratteristiche vantaggiose, come una maggiore uniformità nei risultati tra *zero-shot* e *few-shot*, ha mostrato anche alcuni limiti specifici nel riconoscimento delle risposte corrette e nella valutazione complessiva delle query SQL.

Uniformità dei Risultati

Uno degli aspetti più significativi di Gemma2B è stata la sua capacità di mantenere un dislivello relativamente ridotto nei risultati tra le due modalità di addestramento, *zero-shot* e *few-shot*. Questo indica che il modello ha reagito in modo simile in entrambi i contesti, senza grossi miglioramenti nel passaggio da uno scenario senza esempi (*zero-shot*) a uno con esempi (*few-shot*). Tale uniformità potrebbe riflettere una maggiore coerenza interna nelle risposte generate, ma anche una scarsa capacità di apprendere dalle risposte di esempio, tipica del metodo *few-shot*.

Loop nelle Risposte

Uno dei difetti più evidenti di Gemma2B è stata la tendenza a entrare in un "loop" di risposte ripetitive, basate su risposte precedentemente corrette. Questo comportamento era particolarmente evidente durante le valutazioni delle query, dove il modello, anziché fornire una nuova analisi, ripeteva o riformulava risposte già date. Questo limite ha influenzato negativamente la sua capacità di fornire valutazioni uniche e ha ridotto la sua flessibilità e adattabilità alle diverse situazioni presentate dagli esercizi SQL.

Difficoltà nel Riconoscere Risposte Completamente Corrette

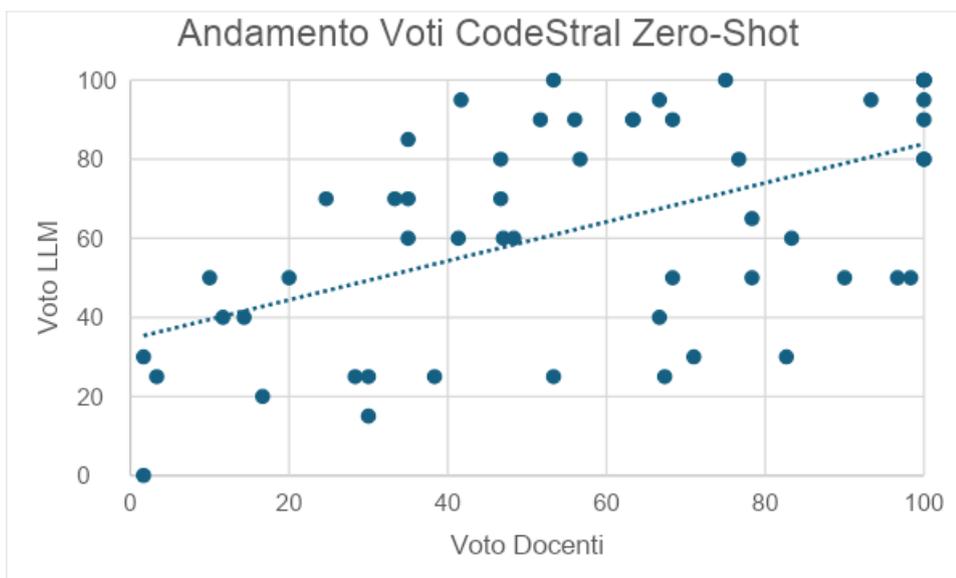
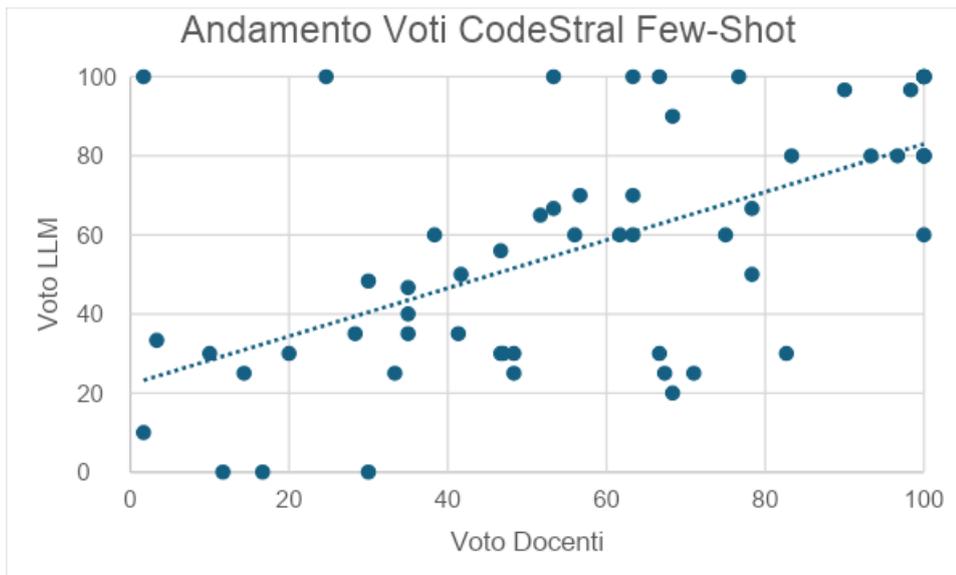
Come ChatGPT, anche Gemma2B ha mostrato difficoltà nel riconoscere risposte totalmente corrette e nel conferire un punteggio pieno (100). Questo limite si è tradotto in valutazioni più basse di quelle dei docenti per esercizi che avrebbero meritato un punteggio massimo. Il modello sembrava avere una maggiore sensibilità agli errori minori, che venivano ingigantiti nella valutazione complessiva, penalizzando così le risposte corrette.

Generosità per Voti Bassi

Gemma2B ha mostrato una maggiore generosità nel valutare risposte parzialmente corrette o contenenti errori significativi, attribuendo voti più alti rispetto a quanto avrebbero fatto i docenti. Questa tendenza era particolarmente evidente nei voti bassi, dove il modello tendeva a penalizzare meno gli errori rispetto ai docenti. Al contrario, per le risposte quasi perfette o completamente corrette, il modello si dimostrava meno generoso, riflettendo una maggiore severità nei voti alti.

Gemma2B ha offerto prestazioni uniformi e relativamente stabili, ma con alcuni difetti, come la tendenza a ripetere risposte e la difficoltà nel riconoscere risposte completamente corrette. Tuttavia, la sua generosità per voti bassi e la minore variabilità tra *zero-shot* e *few-shot* lo rendono un modello interessante, ma meno sofisticato rispetto ad altre soluzioni.

5.2.3 CodeStral



Il modello **CodeStral** ha mostrato alcune caratteristiche distintive durante la fase di sperimentazione, emergendo come uno dei modelli più generosi nelle valutazioni e mostrando una buona capacità nel riconoscere gli esercizi corretti. Tuttavia, il comportamento del modello variava significativamente a seconda che fosse applicato il metodo *zero-shot* o *few-shot*, evidenziando alcune limitazioni, soprattutto nella modalità *zero-shot*.

Generosità nelle Valutazioni

Uno degli aspetti più evidenti di CodeStral è stata la sua tendenza a essere molto generoso nelle valutazioni, in particolare per quanto riguarda l'assegnazione di punteggi perfetti. Il modello ha riconosciuto più frequentemente rispetto ad altri modelli gli esercizi completamente corretti, assegnando il punteggio massimo. Questo comportamento indica una buona capacità di identificare le risposte corrette e di non penalizzare ingiustamente esercizi che non contenevano errori.

Differenze tra *Zero-shot* e *Few-shot*

Durante l'utilizzo del metodo *zero-shot*, CodeStral ha mostrato una maggiore generosità nei voti bassi, rispetto a quanto avveniva con il metodo *few-shot*. Ciò significa che, quando il modello non disponeva di esempi di riferimento (*zero-shot*), tendeva a penalizzare meno gli errori, risultando quindi più indulgente nelle sue valutazioni complessive. Al contrario, con il metodo *few-shot*, CodeStral si dimostrava più rigoroso nei voti bassi, grazie alla possibilità di apprendere dalle risposte di esempio, migliorando la sua accuratezza nell'identificazione degli errori.

Alta Precisione nel Riconoscimento degli Errori con *Few-shot*

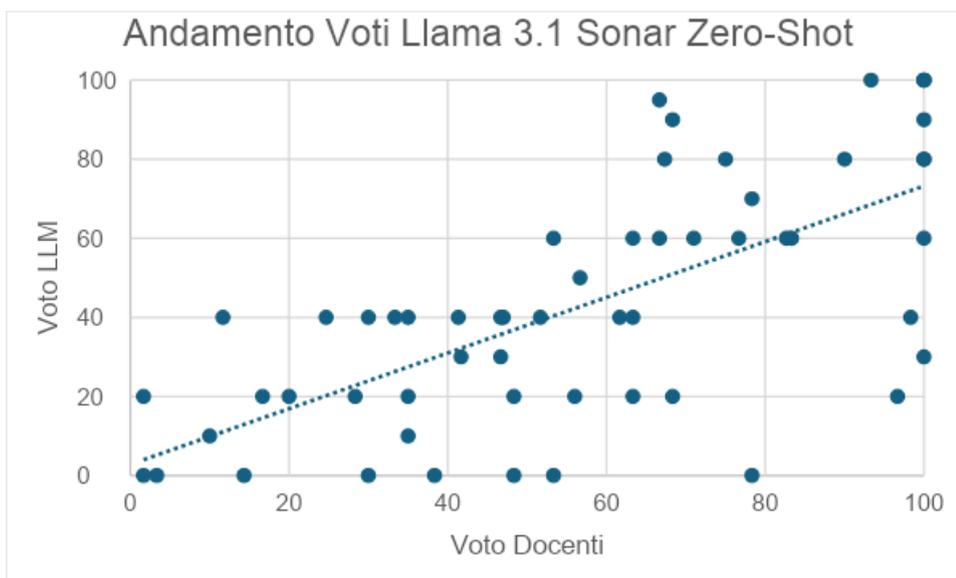
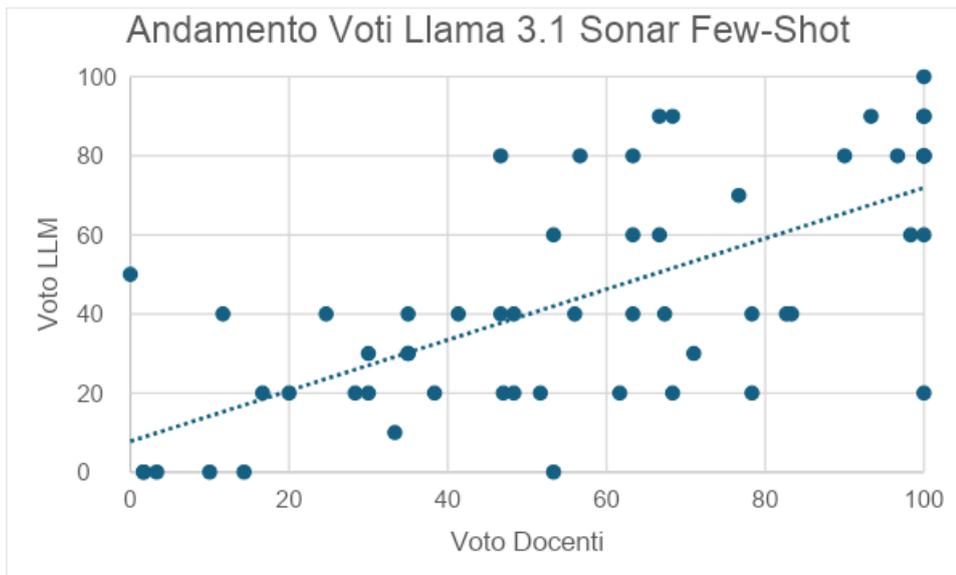
CodeStral ha mostrato una precisione particolarmente elevata nel riconoscere gli errori durante l'applicazione del metodo *few-shot*. La presenza di esempi di errore ha consentito al modello di effettuare una valutazione più accurata delle query SQL, identificando con maggiore precisione i difetti e assegnando voti coerenti con le performance degli studenti. Tuttavia, questa precisione si è ridotta notevolmente in modalità *zero-shot*, dove il modello era più propenso a ripetere le stesse risposte o a entrare in "loop", replicando soluzioni precedentemente corrette senza adattarsi alle specifiche delle nuove query.

Comportamento in Loop in Modalità *Zero-shot*

Uno dei principali difetti di CodeStral è stata la tendenza a entrare in loop durante la valutazione in modalità *zero-shot*, ovvero senza esempi di riferimento. Il modello spesso replicava le risposte corrette delle query precedenti, anziché adattarsi alle nuove query. Questo comportamento ha limitato la capacità di CodeStral di fornire soluzioni uniche e pertinenti per ogni esercizio, penalizzando la sua performance nella modalità *zero-shot*.

In sintesi, CodeStral è stato uno dei modelli più generosi nelle valutazioni, con ottime capacità nel riconoscere le risposte corrette e nell'assegnare punteggi elevati. Tuttavia, il suo comportamento in modalità *zero-shot* era limitato da una tendenza a ripetere risposte precedenti, mentre la sua performance in modalità *few-shot* ha dimostrato un'elevata precisione nel riconoscimento degli errori, grazie alla presenza di esempi. Confrontato con gli altri modelli, CodeStral ha evidenziato pregi e difetti specifici, risultando particolarmente efficace nel riconoscimento degli esercizi corretti, ma con alcune limitazioni nella gestione di query nuove o non supportate da esempi.

5.2.4 Llama 3.1 Sonar



Durante la fase di sperimentazione, il modello **LLaMA 3.1 Sonar** ha mostrato un comportamento rigoroso e affidabile, caratterizzato da una notevole precisione nella valutazione degli esercizi con voti bassi e da una generale tendenza a essere meno generoso nell'assegnazione di voti alti. Questo modello ha richiesto poche correzioni durante l'addestramento, dimostrando stabilità e coerenza nelle risposte, specialmente rispetto ad altri modelli.

Precisione nelle Valutazioni con Voti Bassi

Uno dei punti di forza principali di LLaMA 3.1 Sonar è la sua elevata precisione nel valutare gli esercizi che meritano voti bassi. Similmente a **CodeStral**, il modello ha dimostrato un'ottima capacità di identificare con esattezza gli errori più gravi nelle query, riflettendo in modo accurato i difetti presenti nelle risposte degli studenti. Questa precisione ha reso il modello

particolarmente affidabile nel segnalare quando un esercizio necessita di una valutazione bassa, riuscendo a distinguere in modo netto tra errori significativi e minori.

Poca Generosità nei Voti Alti

LLaMA 3.1 Sonar è risultato tra i modelli meno generosi durante la fase di sperimentazione, assegnando pochi voti pari a 100. Questo indica che il modello applicava criteri stringenti anche quando gli esercizi erano quasi corretti, un comportamento simile a quello di **CodeStral**. La sua inclinazione a non assegnare voti massimi suggerisce un approccio conservativo e una certa severità nell'analisi delle query SQL, probabilmente dovuta a una valutazione rigorosa della corrispondenza tra la risposta e le aspettative dell'esercizio.

Stabilità durante l'Addestramento

Un aspetto distintivo di LLaMA 3.1 Sonar è la sua stabilità durante l'addestramento. Insieme a **ChatGPT**, il modello ha richiesto meno interventi correttivi e ha dimostrato una maggiore capacità di adattarsi alle nuove situazioni senza ripetere continuamente risposte precedenti. Questo significa che LLaMA 3.1 Sonar è riuscito a evitare "loop" di risposte, una problematica che ha invece afflitto modelli come **Gemma2B**, dove le risposte corrette venivano riproposte in modo reiterato senza adattamenti. La ridotta necessità di consigli o interventi correttivi ha confermato la solidità del modello.

Differenze tra *Few-shot* e *Zero-shot*

Nel confronto tra le modalità *few-shot* e *zero-shot*, LLaMA 3.1 Sonar ha mostrato una leggera tendenza a essere più generoso con i voti in modalità *few-shot*, in particolare per gli esercizi con valutazioni basse. Questo comportamento può essere spiegato dall'accesso a esempi di errore, che ha permesso al modello di comprendere meglio i contesti e di diventare più indulgente nelle valutazioni. Tuttavia, questa generosità non ha compromesso la precisione generale del modello, che ha mantenuto una buona coerenza nella valutazione degli errori.

Riassumendo, **LLaMA 3.1 Sonar** si è dimostrato un modello altamente affidabile, capace di identificare con precisione gli errori e di mantenere stabilità e coerenza durante l'addestramento. Sebbene meno generoso nell'assegnare voti alti, il modello ha mantenuto una solida performance, soprattutto nelle valutazioni di esercizi con errori significativi, posizionandosi tra i migliori in termini di stabilità e precisione, insieme a **ChatGPT**.

6. Discussione

6.1 Confronto con la Letteratura

Il confronto tra i risultati ottenuti in questo studio e lo studio condotto da Cagliero et al. rivela una serie di progressi significativi e conferme rispetto a quanto documentato negli studi precedenti. Durante la fase di sperimentazione, molti dei comportamenti osservati nei modelli di apprendimento automatico utilizzati per la valutazione delle query SQL hanno riflettuto le tendenze già evidenziate dalla ricerca, ma con alcune variazioni rilevanti. In particolare, la correlazione tra i voti assegnati dai modelli e quelli dei docenti, l'efficacia del Few-Shot Learning e la capacità dei modelli di affrontare diversi tipi di errori sono stati gli aspetti centrali del confronto con la letteratura preesistente.

Correlazione di Pearson e Accuratezza dei Voti

Uno dei principali risultati di questo studio è stato l'alto grado di correlazione tra i voti assegnati dai modelli e quelli dei docenti, misurato attraverso il coefficiente di Pearson. Questo indice ha mostrato valori generalmente elevati, indicando una forte corrispondenza tra le valutazioni generate dai modelli di apprendimento automatico e quelle assegnate manualmente dai docenti. Nella letteratura precedente, in particolare nello studio sull'uso di Large Language Models (LLM) per la correzione automatica degli esercizi di Cagliero et al., si era già notato che i modelli come ChatGPT tendono a seguire abbastanza fedelmente i criteri di valutazione umani, pur mostrando una leggera tendenza alla severità. Anche negli esperimenti effettuati, ChatGPT ha mantenuto un atteggiamento piuttosto severo nelle sue valutazioni, assegnando voti più bassi rispetto ai docenti in molti casi. Tuttavia, la correlazione elevata indica che, nonostante la severità, il modello è in grado di mantenere una coerenza generale con il giudizio umano.

Questo risultato è coerente con quanto riportato in letteratura, dove si evidenzia che, pur essendo rigorosi, i modelli di apprendimento automatico mostrano una buona capacità di replicare i criteri di valutazione umana. La correlazione di Pearson elevata conferma che i modelli utilizzati, sebbene più severi, seguono un criterio di valutazione che riflette fedelmente quello degli insegnanti, riducendo così il rischio di discrepanze eccessive tra le valutazioni automatiche e quelle manuali. Tuttavia, è stato notato che non solo ChatGPT, ma anche altri modelli analizzati, come Gemma2B e CodeStral, tendevano a essere severi, confermando una tendenza generale dei modelli ad assegnare voti più bassi rispetto a quelli degli esseri umani.

ChatGPT e la Severità del Modello

Uno degli aspetti che ha maggiormente attirato l'attenzione nella fase di sperimentazione è stato il comportamento di ChatGPT rispetto alla severità delle valutazioni. Sebbene i risultati ottenuti abbiano confermato i risultati della letteratura in cui ChatGPT viene definito un

modello più severo nel confronto tra modelli e docenti, anche gli altri modelli analizzati, come Gemma2B e CodeStral, hanno mostrato una tendenza simile. Tuttavia, ChatGPT si è dimostrato particolarmente punitivo quando si trattava di valutare query che contenevano lievi errori o imperfezioni, assegnando spesso voti inferiori rispetto ai docenti. Questo comportamento era atteso, poiché lo studio di Cagliero et al. aveva già evidenziato come i modelli basati su LLM tendano a dare maggior peso agli errori formali e sintattici, rispetto a quelli semantici o logici, che spesso richiedono una comprensione più profonda del contesto.

In termini di severità, ChatGPT ha mostrato un atteggiamento più rigoroso nei confronti delle query con errori, specialmente quando si trattava di errori semantici e logici, risultando meno indulgente rispetto ai docenti umani. Tuttavia, questo è stato un comportamento condiviso in vari gradi anche dagli altri modelli. La letteratura precedente aveva già osservato che ChatGPT, a differenza di altri modelli, tendeva a penalizzare più pesantemente i piccoli errori sintattici, attribuendo un peso maggiore a questi aspetti rispetto ai docenti, che talvolta considerano anche altri fattori più legati alla logica dell'esercizio.

Few-Shot Learning e il Problema della Ripetizione delle Risposte

Una delle principali innovazioni dello studio rispetto alla ricerca precedente è stata l'approfondimento dell'efficacia del Few-Shot Learning, che consiste nel fornire al modello pochi esempi prima di sottoporgli un compito. Durante la sperimentazione, si è osservato che il Few-Shot Learning migliorava le performance generali del modello, consentendo una valutazione più accurata delle query rispetto al Zero-Shot Learning, che non prevede alcun esempio preliminare. Tuttavia, è emerso anche un limite significativo: in alcuni casi, il modello tendeva a dare troppa enfasi agli esempi forniti, entrando in un "loop di risposte" dove copiava soluzioni da query precedenti o replicava la risposta agli esercizi d'esempio forniti inizialmente.

Questo comportamento è stato documentato anche in studi precedenti, che avevano evidenziato come il Few-Shot Learning, pur migliorando la performance, può portare a una sovra-dipendenza dagli esempi. Nella sperimentazione, questo fenomeno si è manifestato in modo evidente, soprattutto con ChatGPT e Gemma2B, che tendevano a ripetere risposte già utilizzate nelle query di esempio. La letteratura precedente aveva evidenziato lo stesso problema, suggerendo che, in alcuni casi specifici, l'uso dello Zero-Shot Learning può risultare più vantaggioso, poiché elimina il rischio di ripetizione meccanica delle risposte. Tuttavia, anche in questo caso, il Few-Shot Learning ha complessivamente portato a un miglioramento delle performance dei modelli, confermando quanto già noto in letteratura: la fornitura di alcuni esempi migliora la comprensione del modello e la sua capacità di generalizzare a nuove query.

Tipologie di Errori: Sintattici, Semantici e Logici

Un'altra importante conferma emersa dalla ricerca riguarda la capacità dei modelli di identificare e correggere i diversi tipi di errori presenti nelle query SQL. Come evidenziato

dallo studio di Cagliero et al. e specificato anche nell'analisi degli errori di Taipalus et al., è emerso che i modelli di apprendimento automatico hanno una maggiore facilità nell'identificare e correggere errori sintattici rispetto a quelli semantici e logici. Gli errori sintattici, essendo legati alla struttura del codice e quindi più facilmente rilevabili tramite regole formali, sono stati individuati con maggiore precisione. Questa tendenza era già stata documentata, e i risultati la confermano: i modelli, in particolare ChatGPT e CodeStral, hanno mostrato una notevole efficacia nel rilevare errori di sintassi.

Per contro, gli errori semantici, che richiedono una comprensione più profonda delle intenzioni del programmatore, e gli errori logici, che riguardano la coerenza della logica all'interno della query, sono risultati più difficili da individuare. La letteratura precedente aveva già segnalato queste difficoltà, evidenziando come i modelli tendano a privilegiare la forma rispetto al contenuto. Anche in questo studio, è emerso che i modelli, pur mostrando un miglioramento nell'identificazione di errori sintattici, hanno continuato a fare fatica nel riconoscere e correggere errori semantici e logici. Questo riflette un limite intrinseco degli attuali modelli di apprendimento automatico, che potrebbero beneficiare di tecniche di addestramento più avanzate o di dataset più ampi e rappresentativi.

Punti di Attenzione e Riconoscimento degli Errori Complessi

Infine, uno dei limiti osservati durante la sperimentazione riguarda la difficoltà dei modelli nel riconoscere i punti di attenzione nelle query SQL. Queste complicazioni, che possono derivare dall'uso non ottimale di funzioni o dall'inclusione di condizioni non necessarie, rappresentano una sfida per i modelli di apprendimento automatico. In questo esperimento, i modelli hanno mostrato difficoltà nel distinguere tra errori gravi e punti di attenzione minori, penalizzando a volte query che, pur essendo corrette, contenevano elementi ridondanti o non necessari. Questo comportamento era stato documentato anche nello studio di Cagliero et al., che aveva segnalato la necessità di migliorare la capacità dei modelli di riconoscere non solo la correttezza formale, ma anche l'efficienza e la qualità complessiva della query.

I risultati ottenuti in questo studio confermano molte delle tendenze già evidenziate nella ricerca di Cagliero et al., ma apportano anche nuove informazioni riguardo all'uso del Few-Shot Learning e alla capacità dei modelli di affrontare query con errori complessi. Sebbene siano stati fatti progressi significativi, restano ancora margini di miglioramento, in particolare nell'ambito del riconoscimento degli errori semantici e logici, così come nella gestione dei punti di attenzione all'interno delle query SQL.

6.2 Implicazioni della Ricerca

Durante la fase di sperimentazione sono emersi diversi punti di forza e debolezza nei vari modelli e nei metodi di addestramento utilizzati, che potrebbero rendere alcuni di essi più

adatti o meno idonei a determinati ambiti di applicazione. Di seguito, una relazione sulle caratteristiche di ciascun modello analizzato.

6.2.1 ChatGPT

Pro e Contro

- **Pro:** ChatGPT si è distinto per una maggiore capacità di diversificare le risposte e non ripetere pattern già utilizzati, soprattutto nel contesto *zero-shot*. Inoltre, la correlazione tra i voti assegnati da ChatGPT e quelli dei docenti, misurata dal coefficiente di Pearson, è risultata migliore rispetto a modelli come Gemma2B e CodeStral. Questo riflette una maggiore capacità di ChatGPT di allinearsi, seppur con una tendenza al ribasso, al criterio di valutazione umano.
- **Contro:** La difficoltà di assegnare il punteggio massimo e la tendenza ad essere più severo rispetto ai docenti rappresentano le principali limitazioni di ChatGPT. In particolare, se confrontato con Llama 3.1 Sonar, il modello ha mostrato una minore accuratezza nel riconoscere i casi di completa correttezza. Anche il suo approccio più punitivo per voti più alti può rappresentare uno svantaggio in contesti educativi, dove una valutazione eccessivamente rigorosa potrebbe non riflettere il reale livello di competenza dello studente.

Casi d'Uso consigliati

ChatGPT è ideale per valutazioni complesse in cui è necessaria una maggiore attenzione ai dettagli e una diversificazione delle risposte, come esami finali o verifiche avanzate dove la precisione è più importante della generosità nella valutazione.

Valutazione di esami complessi e test di fine corso: ChatGPT è particolarmente indicato per valutare esami complessi che richiedono un'attenzione specifica ai dettagli semantici e sintattici. In un contesto di fine corso, dove gli studenti devono dimostrare un livello elevato di competenza, ChatGPT fornisce valutazioni precise, soprattutto in compiti complessi in cui le risposte possono variare notevolmente. Il modello è eccellente nel diversificare le risposte ed evitare la ripetizione di pattern, rendendolo adatto per valutazioni che richiedono un'analisi approfondita delle query SQL.

Ambiti accademici con focus su alta precisione: Data la sua capacità di correlarsi bene ai voti dei docenti, ChatGPT può essere utilizzato in contesti accademici avanzati, come corsi di livello universitario o master. In questi scenari, dove è importante che la valutazione sia allineata agli standard degli insegnanti, il modello eccelle, specialmente quando si richiede un'accurata valutazione delle competenze logiche e di problem solving degli studenti.

Valutazioni in ambienti regolamentati: ChatGPT è adatto in contesti dove le regole di valutazione sono ben definite e devono essere rispettate rigorosamente. Il suo approccio più severo, seppur penalizzante per alcuni studenti, può essere utile in ambienti dove la

precisione e la coerenza dei voti sono fondamentali, come certificazioni professionali in cui la verifica delle competenze è rigorosa.

6.2.2 Gemma2B

Pro e Contro

- **Pro:** L'uniformità dei risultati tra *zero-shot* e *few-shot* rappresenta un pregio in termini di stabilità, ma anche un possibile difetto nella capacità di apprendere dagli esempi. A differenza degli altri modelli, che mostravano una maggiore variabilità tra i due metodi, Gemma2B ha mantenuto un comportamento più costante, limitando però le opportunità di miglioramento con l'addestramento *few-shot*.
- **Contro:** La tendenza a entrare in loop e a ripetere risposte corrette rappresenta un notevole svantaggio, che ha dimostrato una maggiore diversificazione delle risposte. Inoltre, la maggiore generosità nei voti bassi, accompagnata da una minore generosità nei voti alti, ha reso Gemma2B meno equilibrato nella valutazione complessiva. Questo comportamento era meno evidente in modelli come Llama 3.1 Sonar, che ha mostrato un miglior bilanciamento tra le diverse valutazioni.

Casi d'Uso consigliati

Gemma2B è adatto per contesti in cui la coerenza delle valutazioni è più importante dell'apprendimento dinamico, come test standardizzati o valutazioni di base in cui gli errori semantici non sono il focus principale.

Valutazioni di massa in ambienti di apprendimento standardizzato: Gemma2B, con la sua costanza nei risultati tra le diverse modalità di addestramento, è particolarmente utile in contesti in cui si deve valutare un gran numero di studenti su compiti standardizzati. La sua stabilità lo rende adatto per valutazioni su larga scala, dove la coerenza nei risultati è più importante della precisione assoluta. Questo lo rende ideale per esami nazionali o internazionali, dove l'obiettivo è garantire una valutazione equa e uniforme.

Esercitazioni di routine e feedback continui: Dato che Gemma2B si dimostra più generoso sui voti bassi, è utile in contesti educativi in cui lo scopo principale è fornire feedback formativi agli studenti, come esercitazioni settimanali o quiz di autovalutazione. In questi casi, lo scopo è incoraggiare lo studente, e un modello che tende a premiare anche risposte parzialmente corrette può favorire la motivazione all'apprendimento.

Valutazione di studenti principianti: Grazie alla sua capacità di essere più indulgente nei confronti degli errori minori e di dare più spazio a risposte parzialmente corrette, Gemma2B è adatto per corsi introduttivi, dove gli studenti stanno ancora acquisendo le basi delle query SQL. La sua natura generosa lo rende particolarmente utile in corsi di livello base, dove l'obiettivo principale è sviluppare la comprensione generale delle tecniche piuttosto che una rigorosa verifica delle competenze.

6.2.3 CodeStral

Pro e Contro

- **Pro:** CodeStral si distingue per la sua generosità nelle valutazioni, specialmente nei punteggi alti. A differenza di Gemma2B, che tendeva a essere meno generoso con i voti alti e a faticare nel riconoscere risposte totalmente corrette, CodeStral ha dimostrato di essere molto più indulgente e di saper riconoscere in modo efficace le risposte senza errori, assegnando più frequentemente il punteggio massimo.
- **Contro:** Rispetto agli altri modelli, che hanno mostrato una maggiore diversità nelle risposte e una minore tendenza a entrare in loop, CodeStral ha sofferto del problema del loop in modalità *zero-shot*, riducendo la qualità delle valutazioni quando non disponeva di esempi. Inoltre, la precisione nel riconoscimento degli errori di CodeStral era elevata solo con *few-shot*, a differenza di altri modelli che si sono dimostrati invece più equilibrati.

Casi d'Uso consigliati

CodeStral è ideale per valutazioni formative o esercitazioni in cui si vuole incentivare lo studente e premiare il progresso, come durante la fase di apprendimento iniziale o in ambienti di feedback continuo.

Verifiche formative e supporto all'apprendimento personalizzato: CodeStral è ideale per ambienti in cui si vuole supportare l'apprendimento degli studenti fornendo un feedback generoso e incoraggiante. La sua capacità di assegnare più frequentemente voti alti e di riconoscere correttamente le risposte giuste lo rende un buon candidato per strumenti di apprendimento personalizzati, come tutor virtuali o piattaforme di apprendimento adattivo.

Esami intermedi e test di metà corso: Con la sua precisione nel riconoscere errori e la sua generosità nelle valutazioni, CodeStral può essere utilizzato per esami intermedi, dove l'obiettivo è valutare il progresso dello studente senza penalizzare eccessivamente gli errori minori. È particolarmente utile in contesti in cui si vuole monitorare il livello di apprendimento degli studenti a metà del percorso formativo, premiando i progressi senza essere troppo rigorosi sui dettagli.

Feedback rapido in corsi online o piattaforme MOOC: CodeStral può essere implementato efficacemente in piattaforme di apprendimento online (come MOOC) che richiedono valutazioni rapide e automatiche. La sua capacità di assegnare voti alti per risposte corrette e di fornire una valutazione rapida è ideale in corsi che hanno un numero elevato di partecipanti, dove è importante che gli studenti ricevano feedback immediato senza compromettere la qualità della valutazione.

6.2.4 Llama 3.1 Sonar

Pro e Contro

- **Pregi:** LLaMA 3.1 Sonar si distingue per la sua precisione nella valutazione degli esercizi con errori significativi, un aspetto che condivide con CodeStral. Inoltre, la sua stabilità durante l'addestramento e la capacità di evitare risposte ripetitive lo pongono tra i modelli più efficienti, al pari di ChatGPT. Questo lo rende affidabile e solido nel contesto della sperimentazione.
- **Difetti:** Come CodeStral, LLaMA 3.1 Sonar tende a essere meno generoso nei voti alti, una caratteristica che potrebbe penalizzare risposte quasi corrette che potrebbero invece essere valutate in modo più favorevole da altri modelli. Anche se si è dimostrato leggermente più generoso in modalità *few-shot*, questa tendenza non ha avuto un impatto significativo sul miglioramento delle sue valutazioni complessive.

Casi d'Uso consigliati

Llama 3.1 Sonar è adatto per contesti in cui l'identificazione degli errori è critica e si richiede una certa stabilità nel processo di valutazione, come test di valutazione sommativa o verifiche finali in cui si preferisce una penalizzazione minima degli errori formali.

Valutazioni dettagliate di errori logici e sintattici complessi: Llama 3.1 Sonar è particolarmente efficace nel riconoscere e valutare errori complessi in query SQL, rendendolo ideale per scenari dove l'analisi logica dettagliata è fondamentale. Questo lo rende adatto per esami di alto livello, dove la comprensione della logica sottostante alle query SQL è cruciale, come corsi avanzati di database o valutazioni professionali per sviluppatori.

Esami finali con focus su precisione ed errori significativi: Llama 3.1 Sonar è utile in contesti in cui l'obiettivo è penalizzare gli errori significativi e premiare la precisione. Poiché tende ad essere meno generoso con i voti alti, è perfetto per esami finali in cui si richiede una valutazione rigorosa delle competenze. La sua capacità di valutare con precisione esercizi complessi è un vantaggio in contesti dove è richiesta una verifica rigorosa della padronanza del linguaggio SQL.

Valutazioni di certificazione professionale: Il modello può essere utilizzato per esami di certificazione, in cui gli errori minori non possono essere penalizzati pesantemente, ma dove si deve garantire che gli errori logici e concettuali vengano correttamente identificati. La sua accuratezza nel valutare esercizi con errori significativi lo rende perfetto per ambienti dove l'affidabilità della valutazione è critica, come nel rilascio di certificazioni professionali in ambito IT.

6.3 Costi di Inferenza

Uno degli aspetti rilevanti nell'implementazione dei modelli di Large Language Models (LLM) per la valutazione automatizzata delle query SQL è rappresentato dai costi di inferenza, ossia

i costi associati all'elaborazione e alla generazione delle risposte in base ai token di input e output. Di seguito vengono analizzati i costi di inferenza dal punto di vista dell'uso dei token, con un confronto tra modelli open-source e modelli proprietari, ponendo particolare attenzione a come le differenze tra questi due tipi di modelli possano influire sui costi complessivi e sulle applicazioni pratiche in contesti educativi e di ricerca.

6.3.1 Comprensione dei Costi di Inferenza e Struttura dei Token

I costi di inferenza per i modelli LLM sono direttamente correlati al numero di token che devono essere elaborati per ogni richiesta, ossia i token di input, e quelli che il modello genera in risposta, i token di output. I token rappresentano frammenti di testo e corrispondono a parole, prefissi di parole o simboli di punteggiatura. Ogni token è associato a un costo di elaborazione, che dipende dal modello e dal contesto di utilizzo, ed è spesso proporzionale alla potenza computazionale richiesta per elaborarlo.

In contesti pratici, l'uso di un LLM per la valutazione delle query SQL richiede di:

- Elaborare il contesto della query, la struttura della domanda, e la base di conoscenza del database in termini di token di input.
- Generare la valutazione della query, indicando se è corretta e quali errori sono presenti, in termini di token di output.

In termini di applicazione pratica, maggiore è la complessità della query e più alto è il numero di errori o casi particolari da analizzare, maggiore sarà il costo di inferenza complessivo, poiché più token saranno necessari per costruire il prompt di input e generare un output dettagliato e informativo.

6.3.2 Confronto tra Modelli Open-Source e Modelli Proprietari

I modelli open-source e i modelli proprietari differiscono notevolmente sia in termini di struttura dei costi di inferenza sia nell'approccio alla gestione dei token, con ripercussioni sulle scelte pratiche e sui costi per gli utenti finali.

Modelli Open-Source

I modelli open-source come LLaMA, CodeStral e Gemma2B offrono un vantaggio economico rilevante, in quanto non richiedono necessariamente un pagamento per le richieste se implementati su infrastrutture proprie o locali. Tuttavia, l'uso di modelli open-source comporta il costo associato all'infrastruttura hardware, che può essere elevato in contesti che richiedono inferenze su larga scala. Inoltre, l'ottimizzazione dell'uso dei token è meno controllabile rispetto ai modelli proprietari: sebbene gli utenti possano adattare i modelli alle loro esigenze, ogni modifica richiede una gestione accurata per bilanciare l'accuratezza delle risposte e l'efficienza dell'inferenza.

Vantaggi dei modelli open-source:

- **Riduzione dei costi variabili:** Non essendo soggetti a costi per token, se gestiti su hardware proprio, possono rappresentare una soluzione più economica a lungo termine.
- **Flessibilità di adattamento:** Gli utenti possono modificare e addestrare i modelli secondo le necessità specifiche, pur con un costo iniziale di setup.
- **Efficacia per applicazioni statiche:** I modelli open-source sono ideali quando il contesto e i casi d'uso non richiedono aggiornamenti frequenti o complessi, mantenendo i costi di inferenza stabili.

Limiti dei modelli open-source:

- **Costi di infrastruttura:** La gestione di server ad alte prestazioni richiede costi fissi di hardware e manutenzione che possono essere elevati.
- **Ottimizzazione dei token:** È meno immediato ottimizzare il modello per gestire in modo efficace i token di input e output rispetto ai modelli proprietari, con il rischio di inferenze meno efficienti.

Modelli Proprietari

I modelli proprietari, come ChatGPT di OpenAI, offrono il vantaggio di un'infrastruttura di inferenza completamente gestita e ottimizzata, in grado di fornire risultati accurati e veloci senza la necessità di costosi server locali. Tuttavia, ogni richiesta in un modello proprietario comporta costi di inferenza per il numero di token di input e output utilizzati. Questi costi variano in base al modello e possono aumentare rapidamente se le risposte richiedono numerosi token.

Vantaggi dei modelli proprietari:

- **Ottimizzazione dei costi in base all'uso:** Le piattaforme di modelli proprietari offrono infrastrutture ottimizzate che rendono l'inferenza immediatamente disponibile senza costi fissi di server e manutenzione.
- **Efficienza nella gestione dei token:** I modelli proprietari sono progettati per ottimizzare la risposta anche in contesti di alta complessità, bilanciando il numero di token generati con la qualità della risposta.
- **Aggiornamenti e miglioramenti automatici:** I modelli proprietari vengono continuamente aggiornati dai provider, garantendo che le risposte siano basate sulle versioni più avanzate disponibili.

Limiti dei modelli proprietari:

- **Costi elevati per applicazioni frequenti:** In contesti che richiedono inferenze frequenti e di lunga durata, i costi per token possono diventare significativi, aumentando proporzionalmente alla complessità delle richieste.
- **Minor flessibilità:** Rispetto ai modelli open-source, i modelli proprietari sono meno flessibili e meno modificabili dall'utente.

6.3.3 Considerazioni Sui Costi di Inferenza e Scelte di Implementazione

Alla luce dei vantaggi e svantaggi tra i modelli open-source e quelli proprietari, la scelta del modello ideale dipende dalle necessità specifiche dell'applicazione e dal budget disponibile per la valutazione delle query SQL. Per esempio:

- In un contesto educativo con un ampio numero di studenti, dove è necessaria una valutazione frequente e dettagliata delle query, l'uso di un modello open-source implementato su un'infrastruttura locale può rappresentare una soluzione economicamente vantaggiosa.
- In contesti più specializzati o di ricerca, dove la precisione e l'aggiornamento costante delle risposte è cruciale, un modello proprietario può offrire vantaggi in termini di accuratezza e tempo di implementazione, a fronte di un costo variabile per inferenza.

Questa analisi evidenzia come i costi di inferenza siano un fattore da considerare nella scelta tra modelli open-source e proprietari. Ogni scelta comporta compromessi tra l'accuratezza e la personalizzazione delle risposte, la flessibilità dei costi e la possibilità di gestire direttamente l'infrastruttura o di affidarsi a soluzioni esterne.

6.4 Limitazioni dello Studio

In questo studio, sono emerse diverse limitazioni che riguardano sia la metodologia utilizzata sia la generalizzazione dei risultati. Tali limitazioni sono intrinsecamente legate al dataset impiegato, alla capacità di addestramento dei modelli, e alla complessità degli stessi. Esplorare questi limiti è fondamentale per comprendere in che misura i risultati ottenuti possano essere migliorati e generalizzati, nonché per identificare le potenziali aree di miglioramento future.

Numero limitato di tipi di errori analizzato

Uno dei limiti principali dello studio è stato l'impossibilità di coprire in modo approfondito tutte le casistiche di errore esistenti nelle query SQL. Durante la fase di sperimentazione, sono stati presi in considerazione diversi tipi di errori, sia di tipo sintattico che semantico, ma la varietà di errori inclusa non era sufficiente a coprire l'intero spettro delle potenziali problematiche che potrebbero sorgere nell'analisi delle query. Ciò significa che, sebbene i modelli abbiano mostrato una buona capacità di identificare errori comuni, la loro performance potrebbe essere inferiore in presenza di errori più rari o complessi che non erano inclusi nel dataset di training.

Questo limite evidenzia la difficoltà intrinseca nell'addestrare modelli di linguaggio su un set limitato di errori rappresentativi, poiché in un contesto reale, le query SQL possono

presentare una vasta gamma di errori che vanno oltre quelli considerati nello studio. La mancanza di una copertura esaustiva sugli errori riduce quindi la generalizzabilità dei modelli, limitando la loro efficacia in scenari più ampi e complessi.

Limite dei token per l'addestramento del modello

Un'altra significativa limitazione riscontrata è stata il limite di token per l'addestramento del modello. I Large Language Models utilizzati hanno una capacità limitata in termini di numero di token gestibili per ogni prompt. Questo ha richiesto una continua ottimizzazione dei prompt relativi agli esercizi e, in particolar modo, degli errori utilizzati per addestrare il modello.

Il limite dei token ha reso difficile fornire prompt dettagliati ed esaustivi, che avrebbero potuto migliorare le capacità del modello di individuare e correggere errori più complessi. Per ovviare a questo problema, è stato necessario sintetizzare le informazioni e fornire descrizioni meno dettagliate sugli errori e sulle casistiche presenti nelle query. Tuttavia, questa semplificazione ha comportato una riduzione dell'accuratezza del modello, specialmente quando si trattava di riconoscere errori semantici più sottili o di fornire suggerimenti approfonditi.

Prompt degli errori non estremamente dettagliato

Il limite dei token ha avuto un impatto diretto sulla qualità del prompt degli errori. I prompt utilizzati non sono stati sufficientemente dettagliati a causa della necessità di rimanere entro i limiti di token gestibili dal modello. Questo ha limitato la precisione con cui il modello è stato in grado di identificare gli errori e ha ridotto la sua capacità di fornire correzioni appropriate o suggerimenti utili.

La mancanza di dettaglio nei prompt può aver influito negativamente sulla capacità del modello di individuare con precisione gli errori nelle query SQL, in particolare per quanto riguarda errori più complessi che richiedono una descrizione accurata del contesto e delle aspettative del codice SQL. Di conseguenza, la performance del modello può essere stata inferiore rispetto alle aspettative, soprattutto in situazioni in cui un'analisi più dettagliata degli errori sarebbe stata cruciale per fornire un feedback accurato agli utenti.

Dimensioni del dataset di training

Le dimensioni limitate del dataset di training, in particolare per il metodo few-shot learning, hanno rappresentato un ulteriore ostacolo. Sebbene il few-shot learning sia un approccio efficace per addestrare modelli di machine learning con una quantità ridotta di esempi, la quantità limitata di query passate come esempi ha influito sull'accuratezza del modello. Con un maggior numero di query di esempio, sarebbe stato possibile addestrare il modello in modo più efficace, migliorando la sua capacità di identificare correttamente gli errori e assegnare valutazioni più precise.

Tuttavia, l'aumento del numero di query di esempio avrebbe comportato l'uso di un maggior numero di token. Ciò significa che, per migliorare le performance del modello, sarebbe stato

necessario trovare un compromesso tra l'aumento della dimensione del dataset di training e la capacità del modello.

Query disponibili e mancanza di errori rari

Infine, un altro limite rilevante dello studio riguarda le query disponibili nel dataset. Sebbene il dataset fosse sufficientemente ampio da permettere un addestramento efficace dei modelli, non includeva casistiche di errore rare, che sono spesso quelle più complesse da individuare e correggere. Di conseguenza, i modelli analizzati potrebbero non essere altrettanto efficaci nel riconoscere errori rari o situazioni meno comuni che possono emergere durante l'analisi di query SQL in contesti reali.

Questo limite influisce sulla capacità del modello di generalizzare i risultati ottenuti durante la sperimentazione. In un contesto pratico, gli utenti potrebbero incontrare situazioni in cui il modello fallisce nell'identificare correttamente gli errori, poiché non è stato addestrato su query che includono tali casistiche. Per migliorare questo aspetto, sarebbe necessario espandere il dataset includendo una gamma più ampia di query con errori rari, garantendo così una maggiore copertura delle possibili situazioni di errore che gli utenti potrebbero incontrare.

7. Conclusione

7.1 Sintesi dei principali risultati ottenuti

La ricerca condotta ha portato alla luce una serie di risultati significativi, che non solo confermano l'utilità dei Large Language Models (LLM) nella valutazione delle query SQL, ma contribuiscono anche ad arricchire la comprensione delle loro potenzialità e delle loro limitazioni. I risultati ottenuti rappresentano un contributo importante sia dal punto di vista pratico che teorico, dimostrando come l'uso dei LLM possa supportare in modo efficace l'insegnamento e l'apprendimento di linguaggi complessi come SQL, nonché automatizzare il processo di correzione delle esercitazioni.

7.1.1 Accuratezza nella Valutazione delle Query SQL

Uno dei risultati chiave della ricerca riguarda l'accuratezza dei modelli LLM nell'identificazione e valutazione degli errori presenti nelle query SQL. La sperimentazione ha mostrato che i LLM, tra cui ChatGPT, Gemma2B, CodeStral e Llama 3.1 Sonar, sono in grado di valutare in maniera efficace le query, individuando errori di vario tipo. In particolare, i modelli hanno dimostrato una grande capacità nell'identificazione degli errori sintattici, che sono stati correttamente riconosciuti e classificati in una larga percentuale di casi.

I risultati ottenuti indicano una forte correlazione tra i voti assegnati dai LLM e quelli attribuiti dai docenti umani. Questo risultato, misurato attraverso il coefficiente di correlazione di Pearson, ha mostrato che i modelli possono assegnare voti che riflettono, con una certa accuratezza, le valutazioni umane. Il modello ChatGPT ed in misura minore anche Llama 3.1 Sonar, hanno ottenuto una correlazione elevata, dimostrando che l'LLM è in grado di allinearsi al criterio di valutazione umano, pur mostrando una tendenza a penalizzare più severamente gli errori nelle query SQL rispetto ai docenti.

7.1.2 Valutazione degli Errori Sintattici, Semantici e Logici

Un altro risultato importante riguarda la capacità dei modelli di riconoscere diversi tipi di errori nelle query SQL. La ricerca ha identificato tre principali categorie di errori: errori sintattici, semantici e logici.

- **Errori Sintattici:** I modelli LLM si sono dimostrati particolarmente efficaci nell'identificare errori sintattici, come l'uso scorretto di parole chiave SQL, errori di formattazione e problemi nella struttura delle query. Questo conferma quanto emerso

in studi precedenti e dimostra che i LLM possono fornire un valido supporto nell'individuare problemi di sintassi.

- **Errori Semantici:** Riguardo agli errori semantici, i risultati sono stati meno consistenti. Gli errori semantici, che riguardano la logica interna delle query e la loro capacità di restituire il risultato desiderato, sono stati più difficili da individuare per i modelli. Questo rappresenta una delle principali aree di miglioramento, poiché, sebbene i modelli abbiano mostrato una buona capacità di rilevare errori logici e strutturali, la comprensione della logica più complessa alla base delle query è risultata meno efficace.
- **Errori Logici:** La capacità dei modelli di riconoscere errori logici, cioè errori che emergono dal malinteso delle richieste dell'esercizio o dalla logica errata delle query, è stata più variabile. I modelli hanno avuto prestazioni miste in questo ambito, e mentre alcuni modelli come ChatGPT e CodeStral hanno mostrato una buona precisione, altri come Gemma2B e Llama 3.1 Sonar hanno avuto difficoltà a riconoscere correttamente errori complessi di logica.

Un altro tipo di errore considerato nella ricerca sono state le Query Invalide, in cui lo studente eludeva la reale richiesta dell'esercizio utilizzando conoscenze pregresse del database. I modelli hanno generalmente incontrato difficoltà nell'identificare questo tipo di errore, poiché spesso tendevano ad assegnare molteplici errori di diverso tipo alla query.

Questo risultato evidenzia la necessità di ulteriori miglioramenti nei modelli LLM, in particolare per quanto riguarda la loro capacità di comprendere la logica alla base delle query SQL. Il riconoscimento degli errori semantici e logici rimane un'area di miglioramento critica per l'implementazione futura.

7.1.3 Prestazioni dei Modelli in con addestramento Few-Shot e Zero-Shot

Un altro risultato ottenuto riguarda l'efficacia delle diverse modalità di addestramento, in particolare il confronto tra few-shot learning e zero-shot learning. La sperimentazione ha dimostrato che la modalità di addestramento few-shot, in cui vengono forniti esempi specifici per migliorare le prestazioni del modello, ha portato a risultati complessivamente migliori rispetto alla modalità zero-shot.

- **Few-Shot Learning:** Questa modalità ha dimostrato di essere particolarmente efficace per migliorare la capacità dei modelli di riconoscere errori complessi e fornire risposte più accurate. I modelli hanno beneficiato dell'uso di esempi concreti e hanno mostrato un miglioramento nella valutazione delle query più difficili. Tuttavia, è stato anche riscontrato che un numero eccessivo di esempi forniti può portare i modelli a ripetere le risposte eccessivamente, riducendo così la capacità di diversificare le valutazioni.

- **Zero-Shot Learning:** Al contrario, la modalità si è rivelata più adatta per situazioni in cui non sono disponibili esempi concreti o quando è necessario un approccio più generico. Sebbene i risultati siano stati meno accurati rispetto al metodo few-shot, lo zero-shot learning ha comunque fornito buone prestazioni, soprattutto in situazioni in cui i modelli dovevano valutare un'ampia gamma di query senza riferimenti precisi.

Il confronto tra queste due modalità di addestramento evidenzia l'importanza di selezionare il metodo più adatto in base al contesto di utilizzo. Il few-shot è particolarmente utile in contesti educativi dove è richiesta una valutazione più precisa, mentre il zero-shot può essere applicato in situazioni meno strutturate.

7.1.4 Generosità e Severità dei Modelli

Un altro risultato importante della ricerca riguarda il comportamento dei modelli in termini di generosità o severità nella valutazione delle query SQL. È stato osservato che alcuni modelli tendono ad essere più generosi nell'assegnare voti, mentre altri modelli risultano più severi, penalizzando anche errori minori.

- **ChatGPT:** Questo modello si è dimostrato tra i più severi nella valutazione, con una tendenza a penalizzare in modo più marcato gli errori, specialmente nelle query che presentavano solo lievi inesattezze. Tuttavia, ha anche dimostrato un'elevata correlazione con i voti assegnati dai docenti, il che suggerisce che, nonostante la severità, le sue valutazioni sono in linea con i criteri di valutazione umana.
- **Gemma2B:** Gemma2B si è rivelato più generoso nel valutare le query con errori minori, ma ha mostrato una minore accuratezza nel riconoscere risposte completamente corrette. Questo comportamento può essere utile in contesti educativi dove è importante incoraggiare gli studenti a migliorare, ma rappresenta anche una limitazione in situazioni in cui è richiesta una valutazione rigorosa.
- **CodeStral e Llama 3.1 Sonar:** Questi modelli hanno mostrato un comportamento più equilibrato, con una buona capacità di riconoscere errori complessi e un approccio più generoso nelle valutazioni. Tuttavia, la loro generosità può rappresentare uno svantaggio in contesti in cui è necessario un feedback più rigoroso.

Il comportamento differente dei modelli in termini di severità e generosità rappresenta un contributo importante, poiché permette di selezionare il modello più adatto in base al tipo di valutazione richiesto.

7.1.5 Identificazione dei Punti di Attenzione nelle Query

La ricerca ha anche evidenziato che i modelli LLM, pur essendo efficaci nell'identificare errori sintattici e logici, faticano nel riconoscere le complicazioni all'interno delle query SQL. I punti di attenzione ricercati includono l'uso superfluo di subquery o l'introduzione di condizioni non necessarie, che non sono veri e propri errori, ma che rendono la query meno efficiente e più complessa.

I risultati ottenuti indicano che i modelli tendono a ignorare o a non penalizzare adeguatamente questi punti, che invece potrebbero compromettere la leggibilità o l'efficienza delle query. Questo rappresenta un limite importante per l'utilizzo dei modelli LLM in contesti in cui l'ottimizzazione del codice è cruciale, suggerendo la necessità di ulteriori ricerche per migliorare la capacità dei modelli di identificare tali situazioni.

7.1.6 Contributi al Campo dell'Educazione

Uno dei contributi più significativi di questa ricerca riguarda l'applicazione pratica dei modelli LLM nel campo dell'educazione. La sperimentazione ha dimostrato che l'uso dei LLM per la valutazione automatica delle query SQL può fornire un supporto valido ai docenti, riducendo il carico di lavoro e migliorando l'efficienza del processo di valutazione.

L'introduzione dei LLM nei contesti educativi può anche migliorare l'esperienza degli studenti, offrendo feedback immediato e suggerimenti su come migliorare le proprie query. I modelli possono essere utilizzati come strumenti di auto-apprendimento, consentendo agli studenti di sperimentare con le query e di ricevere una valutazione automatica e dettagliata degli errori commessi. Questo approccio rappresenta un'innovazione significativa nel campo dell'educazione, contribuendo a rendere il processo di apprendimento più interattivo e personalizzato.

Riassumendo, i principali risultati di questa ricerca dimostrano che i LLM hanno un grande potenziale nell'ambito della valutazione automatica delle query SQL. Sebbene ci siano ancora alcune aree da migliorare, come il riconoscimento degli errori semantici e dei punti di attenzione, i modelli utilizzati hanno dimostrato una notevole capacità di fornire valutazioni accurate e di supportare efficacemente il processo educativo. La ricerca offre quindi contributi significativi sia dal punto di vista teorico che pratico, ponendo le basi per futuri sviluppi nell'uso dei LLM nell'insegnamento e nella correzione automatica di esercizi complessi come quelli relativi al linguaggio SQL.

7.2 Contributi forniti alla Ricerca

La ricerca presentata offre un contributo significativo all'area della valutazione delle query SQL, sia dal punto di vista teorico che pratico. In un contesto in cui l'automazione dei processi di apprendimento e la correzione automatica degli esercizi stanno diventando strumenti sempre più utilizzati nell'educazione, specialmente per discipline tecniche come l'informatica, questa ricerca esplora le potenzialità dei modelli di linguaggio di grandi dimensioni (LLM) nell'automazione della valutazione delle query SQL. Di seguito si delineano i principali contributi teorici e pratici che questa tesi fornisce.

7.2.1 Contributo Teorico

1. Approfondimento sull'uso dei LLM nella valutazione SQL: Uno dei contributi più rilevanti è l'analisi dettagliata dell'efficacia dei LLM nella valutazione automatica delle query SQL. In particolare, il lavoro esplora il potenziale di vari modelli, tra cui ChatGPT, Gemma2B, CodeStral e Llama 3.1 Sonar, nell'identificare errori sintattici, semantici e logici nelle query SQL prodotte dagli studenti. La ricerca conferma quanto emerso dallo studio di Cagliero et al., ossia che i LLM sono particolarmente adatti a individuare errori sintattici, ma aggiunge nuove osservazioni riguardo alle difficoltà nel riconoscimento di errori semantici e logici, nonché nella gestione dei punti di attenzione all'interno delle query.

Questo studio amplia la letteratura esistente sull'utilizzo dei LLM per il supporto educativo, concentrandosi in particolare sulla complessità delle query SQL. L'analisi approfondita delle prestazioni dei modelli in contesti educativi conferma che i LLM possono essere strumenti efficaci per supportare i docenti, ma mette anche in luce i limiti che devono essere affrontati per ottimizzarne l'efficacia.

2. Validazione della modalità Few-Shot e Zero-Shot Learning: Un altro contributo significativo è la validazione delle modalità di apprendimento few-shot e zero-shot nella valutazione delle query SQL. Questa ricerca ha esaminato l'impatto dell'inclusione di esempi concreti (few-shot) e dell'approccio senza esempi (zero-shot) sulle prestazioni dei modelli. I risultati confermano che il few-shot learning, pur migliorando le prestazioni globali dei modelli, può portare a una sovra-emulazione degli esempi forniti, causando talvolta una ripetizione delle risposte e una perdita di creatività nel riconoscere soluzioni alternative.

Tuttavia, la ricerca dimostra che il few-shot learning è generalmente più efficace per migliorare la capacità del modello di riconoscere errori complessi, suggerendo che l'integrazione di ulteriori esempi ben calibrati potrebbe essere una strategia chiave per migliorare ulteriormente l'accuratezza della valutazione.

7.2.2 Contributo Pratico

1. Automatizzazione della Valutazione SQL: Un contributo particolarmente significativo di questa ricerca è l'implementazione di un sistema che utilizza i LLM per automatizzare la valutazione delle query SQL. Questo sistema può ridurre notevolmente il carico di lavoro dei docenti, specialmente in corsi con un alto numero di iscritti, dove la correzione manuale delle query SQL potrebbe essere un compito arduo e dispendioso in termini di tempo. Grazie all'utilizzo dei LLM, è possibile fornire agli studenti un feedback immediato e personalizzato, che li aiuta a comprendere meglio i propri errori e a migliorare le proprie competenze.

L'implementazione di un sistema automatizzato di valutazione delle query SQL basato sui LLM offre un notevole vantaggio in termini di efficienza. Sebbene l'accuratezza dei modelli non sia ancora perfetta, specialmente per quanto riguarda la comprensione degli errori semantici, la possibilità di fornire una correzione tempestiva e accurata in molti casi rappresenta un contributo essenziale all'educazione tecnologica.

2. Adattabilità a diversi contesti educativi: Il sistema sviluppato in questa ricerca ha dimostrato di essere adattabile a contesti diversi, non limitandosi all'uso in un corso universitario di basi di dati. L'analisi delle prestazioni dei modelli in contesti di addestramento differenti (fine-tuning e coarse-tuning) suggerisce che questo approccio può essere facilmente adattato ad altri settori o aree di insegnamento, come la programmazione, le scienze e l'ingegneria. Questo rappresenta un progresso significativo rispetto ai tradizionali strumenti di valutazione automatizzata, che spesso richiedono un processo di sviluppo lungo e complesso per ogni nuovo dominio.

L'adattabilità del sistema offre quindi un contributo pratico importante per gli educatori, poiché consente una maggiore flessibilità nella sua implementazione e una riduzione dei tempi e dei costi necessari per l'integrazione in nuovi contesti.

3. Identificazione automatizzata di errori comuni e rari: La ricerca ha dimostrato che i modelli LLM utilizzati sono in grado di identificare in modo efficace una gamma di errori comuni nelle query SQL. Questo è un contributo pratico significativo, poiché consente di individuare rapidamente i problemi più frequenti che si presentano durante la scrittura di query SQL, come errori di sintassi, uso scorretto delle funzioni di aggregazione, o problemi nel raggruppamento di dati.

Tuttavia, lo studio ha anche evidenziato le difficoltà incontrate dai modelli nell'identificare errori più rari o specifici. Questo aspetto può essere migliorato con future ricerche, ma già nel suo stato attuale, il sistema sviluppato fornisce un valido supporto agli studenti nell'identificazione di problemi standard, facilitando l'apprendimento e la correzione delle proprie soluzioni in tempo reale.

4. Supporto alla personalizzazione dell'apprendimento: Un altro contributo pratico della ricerca riguarda la capacità dei LLM di fornire feedback personalizzati. Grazie all'automatizzazione della valutazione, gli studenti possono ricevere suggerimenti su come migliorare le proprie query SQL in base agli errori commessi. Questo feedback personalizzato

non solo accelera il processo di apprendimento, ma permette anche di identificare aree di miglioramento specifiche per ciascuno studente.

Inoltre, il sistema di valutazione automatizzato può essere utilizzato come strumento di auto-apprendimento per gli studenti. Gli studenti possono sperimentare con le query SQL, ricevendo feedback immediato sui propri errori, senza dover attendere la correzione manuale da parte del docente. Questo aspetto della ricerca contribuisce quindi a un approccio più interattivo e coinvolgente all'apprendimento delle query SQL, offrendo agli studenti la possibilità di perfezionare le proprie competenze in modo autonomo.

7.2.3 Contributi alla Valutazione degli Errori

1. Riconoscimento degli Errori Sintattici e Semantici: I risultati della sperimentazione hanno confermato che i modelli LLM sono particolarmente efficaci nel riconoscimento degli errori sintattici, che rappresentano il tipo di errore più facilmente individuabile dai modelli. Questo contributo è rilevante in contesti educativi in cui la precisione nella sintassi delle query SQL è fondamentale per l'esecuzione corretta del codice. Tuttavia, la ricerca ha anche messo in luce le limitazioni dei modelli nel riconoscere errori più complessi, come quelli semantici e logici. Questo punto rappresenta una sfida futura, ma il riconoscimento già accurato degli errori sintattici è un contributo importante che migliora la capacità dei modelli di offrire valutazioni valide e supporto agli studenti.

2. Gestione dei Punti di Attenzione: La ricerca ha mostrato che i modelli LLM tendono a faticare nel riconoscimento delle complicazioni all'interno delle query SQL, come l'uso non ottimale di subquery o l'introduzione di elementi ridondanti. Questo rappresenta un importante spunto per future ricerche, che potrebbero approfondire la capacità dei modelli di identificare e suggerire soluzioni per query più complesse o meno dirette. Tuttavia, la capacità già dimostrata di identificare errori di base e fornire suggerimenti di miglioramento rappresenta un contributo rilevante per il supporto didattico.

L'uso dei LLM per automatizzare il processo di valutazione rappresenta un progresso significativo sia per l'efficienza dell'insegnamento che per l'apprendimento autonomo degli studenti. I risultati ottenuti confermano l'efficacia dei modelli nell'identificazione di errori sintattici e offrono nuove prospettive per migliorare la capacità di riconoscere errori semantici e logici. Questi risultati, insieme alla possibilità di personalizzare il feedback e alla flessibilità del sistema, dimostrano l'importanza dell'utilizzo dei LLM nel contesto educativo e aprono la strada a nuove ricerche in questo campo.

7.3 Principali Takeaways della Ricerca

La ricerca condotta offre importanti indicazioni e guidelines, delineando i potenziali vantaggi e considerazioni pratiche per l'applicazione dei modelli di Large Language Models alla valutazione delle query SQL. I principali *takeaways* illustrati di seguito includono aspetti che possono guidare l'implementazione pratica nei contesti educativi e formativi.

1. **Automatizzazione della Valutazione delle Query SQL**

La ricerca dimostra che i LLM possono svolgere un ruolo fondamentale nell'automazione della correzione delle query SQL, facilitando sia il lavoro dei docenti che l'esperienza di apprendimento degli studenti. Grazie alla loro capacità di riconoscere e classificare errori comuni, i modelli come ChatGPT, CodeStral e Llama 3.1 Sonar risultano utili per automatizzare il processo di valutazione, riducendo il carico di lavoro manuale per i docenti. Questo approccio permette di offrire feedback più tempestivo agli studenti, accelerando il loro percorso di apprendimento. Per gli utenti finali, questo significa avere un sistema di correzione sempre disponibile e in grado di identificare in modo rapido errori e incongruenze, supportando così un apprendimento più efficace.

2. **Precisione nell'Identificazione degli Errori Sintattici e Limitazioni nei Compiti Complessi**

Un altro risultato evidenziato è che i LLM sono molto efficaci nell'identificazione degli errori sintattici, come quelli relativi all'uso scorretto delle parole chiave SQL o alla formattazione. Questo tipo di errore risulta essere facilmente riconoscibile dai modelli, confermando che l'implementazione di LLM nelle piattaforme didattiche è particolarmente utile per supportare gli studenti nel perfezionamento delle loro abilità tecniche di base. Tuttavia, l'utente finale deve essere consapevole delle limitazioni del sistema nel riconoscere errori logici e semantici, che possono richiedere una comprensione più profonda delle intenzioni dell'autore della query e della logica sottostante all'esercizio.

3. **Severità nella Valutazione: Maggiore Rigore rispetto ai Docenti Umani**

Dalla ricerca emerge che i modelli di linguaggio, in particolare ChatGPT, tendono a essere più severi rispetto ai docenti umani nella valutazione delle query SQL. Questo rigore può risultare utile in contesti dove è necessario mantenere un alto standard di precisione, ma al tempo stesso può rappresentare una sfida per gli studenti, specialmente per quelli che hanno appena iniziato a familiarizzare con SQL. Per l'utente finale, questa severità può quindi tradursi in un'opportunità di crescita, ma è importante considerare la necessità di calibrare la valutazione per evitare di scoraggiare lo studente. In contesti educativi, un bilanciamento tra feedback rigoroso e costruttivo può aiutare a fornire agli studenti un quadro realistico delle loro competenze.

4. **Miglioramento delle Prestazioni con Few-Shot Learning**

L'utilizzo del few-shot learning, ovvero l'inclusione di esempi per guidare la valutazione, ha dimostrato di migliorare sensibilmente la capacità dei LLM di identificare errori complessi e di adattarsi a un contesto specifico. Per gli utenti finali, il few-shot learning offre quindi un vantaggio pratico, permettendo una valutazione più accurata e contestualizzata. Tuttavia, un eccessivo numero di esempi può portare a ripetizioni o a risposte non sempre diversificate, oltre all'utilizzo di un maggior numero di token, che può portare ad una limitazione del numero di errori inseribili nel prompt. Nelle applicazioni educative, è dunque consigliabile calibrare attentamente il numero di esempi forniti, per assicurare che il modello mantenga un buon equilibrio tra accuratezza e creatività nella correzione delle query.

5. **Versatilità dell'Applicazione in Altri Contesti Didattici**

Un altro aspetto rilevante emerso dallo studio è la versatilità dei LLM, che possono essere adattati per la valutazione di altri linguaggi e competenze tecniche, oltre che per SQL. La possibilità di utilizzare i LLM in diversi contesti didattici rappresenta un importante vantaggio, poiché consente di estendere il sistema di valutazione anche a corsi di programmazione, matematica, o altre discipline STEM. Per l'utente finale, questa adattabilità significa poter utilizzare un unico strumento per valutazioni che spaziano su più aree disciplinari, ottimizzando così l'uso delle risorse educative e creando un sistema flessibile e multiuso.

6. **Feedback Personalizzato e Miglioramento delle Competenze Autonome**

I LLM offrono un supporto prezioso per personalizzare il feedback agli studenti, fornendo suggerimenti su come migliorare le proprie query in base agli errori commessi. Questo tipo di feedback non solo rende l'apprendimento più immediato e interattivo, ma permette agli studenti di comprendere le aree in cui possono migliorare, incentivando un apprendimento proattivo. Per gli utenti finali, l'implementazione di questi modelli di valutazione automatica può quindi rappresentare uno strumento di autoapprendimento, offrendo agli studenti la possibilità di perfezionare le proprie abilità senza necessità di un intervento costante da parte dei docenti.

7. **Aumento dell'Efficienza Didattica**

Per docenti e istituzioni educative, i risultati di questa ricerca dimostrano che l'uso di LLM nella correzione delle query SQL può migliorare l'efficienza del processo didattico. Riducendo il tempo necessario per la correzione manuale degli esercizi, i LLM consentono ai docenti di dedicarsi maggiormente ad attività più strategiche, come il supporto personalizzato agli studenti o l'elaborazione di nuovi materiali didattici. Questa maggiore efficienza rappresenta un valore aggiunto per l'utente finale, poiché offre un processo di insegnamento più snello, accessibile e capace di rispondere prontamente alle esigenze degli studenti.

Questa ricerca offre importanti considerazioni pratiche per l'utente finale, evidenziando come l'uso dei LLM possa contribuire a rendere più agevole ed efficace l'apprendimento di SQL. Tuttavia, è essenziale essere consapevoli delle limitazioni del sistema, specialmente riguardo alla severità della valutazione e alla capacità di interpretare errori complessi o specifici. Questi *takeaways* forniscono una guida utile per l'implementazione futura dei LLM, indicando le migliori modalità d'uso dei modelli per supportare la valutazione automatica e personalizzata delle competenze tecniche degli studenti.

7.4 Proposte di Ricerca Future

Il presente studio ha posto delle basi per l'utilizzo dei Large Language Models (LLM) nella valutazione delle query SQL, ma ci sono numerose aree che potrebbero essere ulteriormente esplorate e sviluppate. Le proposte per le ricerche future sono incentrate sull'espansione del modello, sull'integrazione di nuove tecnologie e sull'adattamento a contesti educativi più ampi e diversificati. Di seguito, alcune delle principali linee di sviluppo che potrebbero essere perseguite.

7.4.1 Applicazione di Altri Modelli LLM

Una delle prime direzioni per le prossime ricerche riguarda l'integrazione e la sperimentazione con altri modelli LLM oltre a quelli già utilizzati. Mentre ChatGPT, Gemma2B, CodeStral e Llama 3.1 Sonar sono stati fondamentali in questa fase, altri modelli, come PaLM di Google o Gopher di DeepMind, potrebbero offrire prestazioni differenti. La diversità dei modelli LLM fornisce una varietà di approcci all'elaborazione del linguaggio naturale, e l'inclusione di modelli basati su architetture più recenti o avanzate potrebbe incrementare le capacità di valutazione delle query SQL, migliorando al contempo l'accuratezza e la comprensione degli errori semantici e logici.

Questa linea di ricerca potrebbe anche includere modelli multimodali, che integrano dati visivi o grafici insieme al testo, permettendo una valutazione più ricca e complessa, particolarmente utile per query che includono rappresentazioni complesse di dati relazionali.

7.4.2 Estensione del Set di Errori Considerati

Il numero di errori utilizzati per la valutazione delle query SQL è stato limitato a quelli più comuni. Tuttavia, uno sviluppo cruciale per ricerche future sarebbe l'espansione di questo set di errori. Un numero maggiore di tipologie di errori, tra cui quelli più rari o specifici, permetterebbe di testare le capacità dei modelli LLM in modo più completo. Ad esempio,

errori di ottimizzazione delle query, che non sono stati trattati in modo approfondito in questa fase, potrebbero essere inclusi per verificare come i modelli gestiscono query inefficienti dal punto di vista computazionale.

Inoltre, l'integrazione di errori legati a particolari domini di dati o settori specifici, come l'analisi finanziaria o le scienze della vita, potrebbe rappresentare un importante passo avanti per comprendere la versatilità e l'adattabilità dei modelli LLM in contesti diversi. Ciò richiederebbe la creazione di nuovi dataset che rappresentino scenari più variegati, aumentando la complessità delle query da analizzare e potenzialmente migliorando la capacità dei modelli di gestire casi particolari.

7.4.3 Inclusione di Dati da Contesti Educativi Diversificati

Un'altra importante proposta è l'applicazione di modelli LLM in contesti educativi diversi, non limitati al settore delle scienze informatiche. Ad esempio, l'uso dei LLM per la correzione di esercizi in ambiti come la matematica, le scienze sociali o le lingue potrebbe offrire nuove opportunità di sviluppo. Il contesto educativo di provenienza dei dati è fondamentale per valutare la generalizzabilità del modello. Utilizzare dati provenienti da corsi universitari in diverse discipline o in scuole di vario livello permetterebbe di testare se i risultati ottenuti nel contesto SQL sono trasferibili ad altri ambiti di apprendimento.

Inoltre, la sperimentazione con studenti di diverse nazionalità o provenienti da sistemi educativi differenti potrebbe rivelare differenze culturali nell'approccio alla risoluzione dei problemi e, di conseguenza, nella generazione di errori. L'adattabilità dei modelli LLM potrebbe essere ulteriormente testata per comprendere quanto questi strumenti siano efficaci nel fornire feedback personalizzato in ambienti di apprendimento con caratteristiche uniche.

7.4.4 Aumento del Dataset di Addestramento

Il dataset utilizzato in questa ricerca, pur completo e valido, potrebbe essere ulteriormente espanso per migliorare l'efficacia e l'accuratezza dei modelli LLM. Un dataset più ampio, con un numero maggiore di query SQL e relativi errori, permetterebbe di addestrare i modelli su un campione più rappresentativo, migliorando così la capacità di generalizzazione. Inoltre, l'inclusione di dati provenienti da fonti diverse, come esercitazioni pratiche di aziende o database open source, potrebbe arricchire la varietà di casi gestibili dal modello.

Un numero maggiore di query e di tipologie di errori non solo aumenterebbe la complessità del problema, ma fornirebbe anche ai modelli LLM maggiori opportunità di apprendere da casi differenti. Questo approccio potrebbe essere particolarmente utile per modelli come Gemma2B, che hanno mostrato difficoltà nel diversificare le risposte e nell'apprendere dal metodo few-shot.

7.4.5 Miglioramento dei Prompt e della Capacità di Addestramento

Un'altra area di miglioramento significativo riguarda l'ottimizzazione dei prompt utilizzati per istruire i modelli. Nel corso della ricerca, il limite di token rappresentava un ostacolo nel fornire ai modelli tutte le informazioni necessarie in modo efficiente. Ricerche future potrebbero concentrarsi sullo sviluppo di tecniche di compressione o riassunto dei prompt che massimizzino la quantità di informazioni trasmesse senza superare i limiti imposti dalle attuali architetture LLM.

Inoltre, l'utilizzo di tecniche avanzate di fine-tuning e di apprendimento attivo potrebbe essere esplorato per migliorare le capacità dei modelli di apprendere in modo più efficiente dagli esempi forniti. Ad esempio, l'implementazione di tecniche di reinforcement learning potrebbe permettere ai modelli di ottimizzare le loro risposte basandosi sui feedback ricevuti durante l'interazione con docenti o studenti.

7.4.6 Adattamento alle Lingue Diverse

Un'altra area di espansione interessante riguarda l'uso dei LLM per la valutazione delle query SQL in lingue diverse dall'inglese. Attualmente, la maggior parte delle ricerche e dei dataset sono focalizzati sull'inglese come lingua principale, ma l'adattamento dei modelli a contesti multilingue potrebbe rappresentare una sfida interessante. L'espansione a lingue diverse richiederebbe dataset specifici e l'addestramento su query SQL in lingue differenti, offrendo così la possibilità di testare la capacità dei LLM di adattarsi a sintassi linguistiche differenti mantenendo l'accuratezza nella correzione delle query.

7.4.7 Integrazione di Modelli Ibridi

Infine, una proposta per future ricerche potrebbe riguardare l'integrazione di modelli LLM con altre tecnologie di intelligenza artificiale. Ad esempio, l'utilizzo di tecniche di Supervised Learning abbinata ai modelli LLM potrebbe migliorare la capacità di riconoscere errori più complessi e meno comuni. Modelli ibridi, che combinano l'analisi statica e dinamica delle query SQL, potrebbero migliorare l'accuratezza delle valutazioni e fornire feedback più dettagliati e comprensibili.

Inoltre, l'integrazione di Convolutional Neural Network per il riconoscimento di pattern visivi nei dati o con modelli di apprendimento per rinforzo potrebbe migliorare la capacità di adattamento ai diversi contesti di apprendimento. Questa integrazione rappresenterebbe una sfida complessa ma potenzialmente rivoluzionaria per migliorare l'accuratezza e la flessibilità dei sistemi di valutazione automatica delle query SQL.

8. Appendici

8.1 Risultati del modello ChatGPT (primi 6 esercizi)

Few-Shot Learning

Esercizio	ID Risposta	Errori	Errori identificati da LLM	Errori Correttamente Identificati	Errori non Identificati	Errori identificati in eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SEM-4	1	0	2	78,33	70
			SYN-3					
			COM-2					
1	6791	LOG-3	SEM-5	1	1	3	46,67	50
		SEM-5	SEM-4					
			COM-2					
			COM-1					
1	10804	LOG-2	LOG-2	1	0	2	75	60
			SEM-4					
			COM-2					
2	156	COM-7	COM-7	1	0	0	100	96
2	8894	COM-3	COM-3	1	0	2	100	60
			SYN-3					
2	9196		LOG-2	2	1	1	56,67	50
		COM-6	COM-5					
		LOG-3	LOG-3					
		SEM-5	SEM-5					
3	6186	SEM-1	SEM-1	2	0	1	10	0
		LOG-3	LOG-3					
			LOG-5					
3	9767	SEM-6	LOG-5	1	1	3	1,67	0
		SYN-6	SYN-6					
		INV	SYN-1					
			SYN-2					
3	11277	SEM-1	LOG-3	2	1	0	20	30
		LOG-3	LOG-3					
		LOG-5	LOG-5					
3	12525		INV	1	0	2	66,33	20
			LOG-3					
			LOG-5					
3	4131			0	0	0	100	90
4	268	COM-1	SYN-5	0	1	4	83,33	70
			LOG-3					
			COM-4					
			SEM-5					
4	2366	SYN-1	SYN-2	0	1	5	61,67	40
			SEM-6					
			LOG-3					
			SEM-5					
			COM-3					
4	4163	COM-5	COM-5	1	0	4	100	70
			COM-2					
			LOG-3					
			SEM-5					
4	5027	SYN-4	SYN-6	0	2	2	63,33	30
		LOG-4	LOG-3					
			LOG-3					
4	6303	COM-3	COM-3	1	0	0	93,33	90
4	12466	SEM-1	SEM-1	2	1	0	63,33	85
		COM-3	COM-3					
			COM-3					
			SYN-5					
4	3809		COM-3				100	75
			SEM-5					
5	1986	COM-2	COM-2	2	0	1	96,67	60
		COM-4	COM-4					
			SYN-2					
5	2564	COM-2	COM-2	2	0	1	98,33	70
		COM-4	COM-4					
			SYN-2					
5	8876	SYN-2	SYN-1	0	1	2	90	85
			COM-2					
6	3537	SYN-2	SYN-4	0	2	2	68,33	80
		LOG-3	LOG-2					
6	12321	COM-2	COM-2	1	1	2	78,33	60
		COM-4	SYN-2					
		LOG-2						

Zero-Shot Learning

Esercizio	ID Risposta	Errori	Errori Identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SYN-3	0	1	3	78,33	85
			COM-2					
			SEM-5					
1	6791	LOG-3	SEM-1	0	2	4	46,67	60
		SEM-5	SEM-4					
			COM-5					
			LOG-1					
1	10304	LOG-2	LOG-2	1	0	1	75	70
			COM-2					
2	156	COM-7	COM-7	1	0	0	100	95
			COM-3					
2	8894		SYN-3	1	0	4	100	65
			COM-2					
			COM-5					
			SEM-5					
			SEM-5					
2	9196	COM-6	SYN-6	0	3	3	56,67	70
		LOG-3	LOG-4					
		SEM-5	COM-5					
3	8185	SEM-1	SEM-1	2	0	2	10	20
		LOG-3	LOG-3					
			COM-5					
3	9767		LOG-5	2	1	3	1,67	0
		SEM-6	SEM-6					
		SYN-6	SYN-6					
		INV	SYN-2					
			LOG-3					
3	11277		COM-4	2	1	2	20	40
		SEM-1	SEM-1					
		LOG-3	LOG-3					
		LOG-5	COM-5					
3	12525		COM-7	1	0	0	68,33	0
		INV	INV					
3	4131			0	0	0	100	100
4	268	COM-1	SYN-3	0	1	2	83,33	60
			COM-3					
4	2368	SYN-1	SYN-1	1	0	1	61,67	60
			SEM-4					
4	4163	COM-5	COM-6	0	1	1	100	85
4	5027	SYN-4	COM-3	0	2	2	63,33	50
		LOG-4	LOG-1					
		COM-3	SYN-1					
4	6303		LOG-3	0	1	4	93,33	90
			LOG-2					
			SEM-1					
			SEM-1					
4	12498	SEM-1	LOG-1	0	3	2	63,33	75
		COM-3	LOG-2					
		SYN-5						
4	3809		SYN-4	0	0	2	100	90
			LOG-1					
5	1965	COM-2	SEM-4	0	2	3	96,67	80
		COM-4	SYN-5					
			COM-3					
5	2564	COM-2	SYN-5	0	2	3	98,33	75
		COM-4	SYN-2					
			SEM-4					
5	8876	SYN-2	SYN-2	1	0	2	90	60
			SYN-5					
			SEM-1					
6	3537	SYN-2	SYN-1	0	2	2	68,33	70
		LOG-3	SEM-2					
6	12321	COM-2	SEM-1	1	2	2	78,33	50
		COM-4	SYN-3					
		LOG-2	LOG-2					

8.2 Risultati del Modello Gemma2B (primi 6 esercizi)

Few-Shot Learning

Esercizio	ID Risposta	Errori	Errori Identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SEM-4	1	0	1	78,33	70
			LOG-2					
1	6791	LOG-3	SEM-4	0	2	1	46,67	60
		SEM-5						
1	10304	LOG-2	SEM-4	0	1	1	75	60
2	156	COM-7	LOG-3	0	1	1	100	100
2	8894	COM-3	SEM-4	0	1	2	100	30
			LOG-3					
2	9196	COM-6		1	2	0	56,67	100
		LOG-3	LOG-3					
		SEM-5						
3	6165	SEM-1	LOG-5	0	2	1	10	10
		LOG-3						
3	9767	SEM-6		1	2	0	1,67	0
		SYN-6	SYN-6					
		INV						
3	11277	SEM-1		1	2	0	20	80
		LOG-3	LOG-3					
		LOG-5						
3	12525	INV	SYN-6				68,33	40
			LOG-5					
3	4131		SYN-6	0	0	2	100	20
			LOG-5					
4	268	COM-1	SYN-5	0	2	1	83,33	30
			SEM-5					
4	2368	SYN-1	SEM-5	0	2	1	61,67	40
			COM-3					
4	4163	COM-5	SEM-5	0	1	2	100	60
			COM-3					
4	5027	SYN-4	SYN-5	0	2	2	63,33	50
		LOG-4	SEM-5					
4	6303	COM-3	COM-3				93,33	75
4	12498	SEM-1	SEM-1	1	2	0	63,33	65
		COM-3						
		SYN-5						
4	3809			0	0	0	100	80
5	1965	COM-2		0	2	0	96,67	75
		COM-4						
5	2564	COM-2		0	2	0	98,33	65
		COM-4						
5	8876	SYN-2	SEM-5	0	1	1	90	70
6	3537	SYN-2		1	1	0	68,33	65
		LOG-3	LOG-3					
6	12321	COM-2	LOG-3	0	3	1	78,33	75
		COM-4						
		LOG-2						

Zero-Shot Learning

Esercizio	ID Risposta	Errori	Errori Identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SEM-6	0	1	1	78,33	60
1	6791	LOG-3	COM-5	0	2	1	46,67	70
		SEM-5						
1	10304	LOG-2	COM-6	0	1	1	75	75
2	156	COM-7	SEM-5	0	1	1	100	80
2	8894	COM-3	COM-7	0	1	1	100	65
2	9196	COM-6		1	2	0	56,67	80
		LOG-3						
		SEM-5	SEM-5					
3	6165	SEM-1	SEM-5	0	2	1	10	10
		LOG-3						
3	9767	SEM-6	SEM-5	0	3	2	1,67	30
		SYN-6	LOG-2					
		INV						
3	11277	SEM-1		0	3	0	20	100
		LOG-3						
		LOG-5						
3	12525	INV	LOG-5				68,33	20
3	4131	INV					100	60
4	268	COM-1	COM-1	1	0	1	83,33	60
			LOG-3					
4	2368	SYN-1	COM-1	0	1	2	61,67	65
			LOG-3					
4	4163	COM-5	LOG-3	0	1	1	100	80
4	5027	SYN-4	SYN-1	0	2	2	63,33	75
		LOG-4	LOG-3					
4	6303	COM-3	SYN-1	0	1	1	93,33	80
4	12498	SEM-1	SYN-1	0	3	1	63,33	65
		COM-3						
		SYN-5						
4	3809	SYN-1					100	85
5	1965	COM-2	SYN-1	0	2	1	96,67	75
		COM-4						
5	2564	COM-2	SYN-1	0	2	1	98,33	80
		COM-4						
5	8876	SYN-2	SYN-2	1	0	1	90	75
			SYN-1					
6	3537	SYN-2		0	2	1	68,33	60
		LOG-3	COM-1					
6	12321	COM-2	COM-2	1	2	0	78,33	70
		COM-4						
		LOG-2						

8.3 Risultati del Modello CodeStral (primi 6 esercizi)

Few-Shot Learning

Esercizio	ID Risposta	Errori	Errori Identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SEM-4	1	0	1	78,33	50
			LOG-2					
			SYN-3					
1	6791	LOG-3	LOG-3	1	1	2	46,67	30
		SEM-5	SEM-4					
			LOG-2					
1	10304	LOG-2	LOG-3	0	1	2	75	60
			SEM-5					
2	156	COM-7		0	1	0	100	100
2	8894	COM-3	SYN-3	0	1	2	100	80
			COM-7					
2	9196	COM-6		2	1	0	56,67	70
		LOG-3	LOG-3					
		SEM-5	SEM-5					
3	6165	SEM-1	LOG-5	1	1	2	10	30
		LOG-3	LOG-3					
			SEM-5					
3	9767	SEM-6	LOG-3	1	2	2	1,67	10
		SYN-6	SYN-6					
		INV	LOG-5					
3	11277	SEM-1	SEM-5	2	1	1	20	30
		LOG-3	LOG-3					
		LOG-5	LOG-5					
3	12525	INV	LOG-3	0	1	3	68,33	20
			LOG-5					
			SEM-1					
3	4131			0	0	0	100	100
4	288	COM-1	SYN-5	0	1	2	83,33	80
			SEM-5					
4	2388	SYN-1	SYN-2	0	1	4	61,67	60
			LOG-2					
			LOG-5					
4	4163	COM-5	LOG-2	0	1	2	100	80
			SEM-5					
4	5027	SYN-4	COM-3	0	2	4	63,33	60
		LOG-4	LOG-3					
			SYN-3					
4	6303	COM-3	SEM-5	0	1	1	93,33	80
		SEM-1	SEM-5					
		COM-3	COM-3					
4	12498	COM-3	COM-3	1	2	1	63,33	70
		SYN-5						
4	3809		SEM-5	0	0	1	100	80
5	1965	COM-2	COM-2	2	0	0	96,67	80
		COM-4	COM-4					
5	2564	COM-2		1	1	0	98,33	96,67
		COM-4	COM-4					
5	8876	SYN-2	SYN-2	1	0	0	90	96,67
6	3537	SYN-2		1	1	0	68,33	90
		LOG-3	LOG-3					
6	12321	COM-2	SYN-2	0	2	0	78,33	66,67
		COM-4						
		LOG-2						

Zero-Shot Learning

Esercizio	ID Risposta	Errori	Errori Identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SYN-3	0	1	2	78,33	50
			LOG-5					
1	6791	LOG-3	LOG-3	1	1	1	46,67	70
		SEM-5	COM-1					
1	10304	LOG-2		0	1	0	75	100
2	156	COM-7		0	1	0	100	100
2	8894	COM-3	SYN-3	0	1	1	100	80
2	9196	COM-6		1	2	0	56,67	80
		LOG-3	LOG-3					
		SEM-5						
3	6165	SEM-1		1	1	0	10	50
3	9767	LOG-3	LOG-3	0	3	2	1,67	30
		SEM-6	SYN-2					
		SYN-6	LOG-2					
3	11277	INV		1	2	0	20	50
		SEM-1						
		LOG-3	LOG-3					
3	12525	LOG-5		0	1	1	68,33	50
3	4131	INV	LOG-3	0	1	1	68,33	50
4	268	COM-1	COM-1	1	0	2	83,33	60
			SYN-6					
			SEM-1					
4	2368	SYN-1	SYN-6	0	1	3	61,67	
			LOG-2					
			LOG-5					
4	4163	COM-5	COM-5	1	0	1	100	90
			COM-4					
4	5027	SYN-4	SYN-4	2	0	1	63,33	
		LOG-4	LOG-4					
			LOG-5					
4	6303	COM-3	COM-7	0	1	1	93,33	95
4	12498	SEM-1	COM-2	1	2	1	63,33	90
		COM-3	COM-3					
		SYN-5						
4	3809		SYN-6	0	0	2	100	80
			LOG-5					
5	1965	COM-2	LOG-3	0	2	1	96,67	50
		COM-4						
5	2564	COM-2	LOG-3	0	2	1	98,33	50
		COM-4						
5	8876	SYN-2	LOG-3	0	1	1	90	50
6	3537	SYN-2	SYN-2	1	1	0	68,33	90
		LOG-3						
6	12321	COM-2	LOG-3	0	3	1	78,33	65
		COM-4						
		LOG-2						

8.4 Risultati del Modello Llama 3.1 Sonar (primi 6 esercizi)

Few-Shot Learning

Esercizio	ID Risposta	Errori	Errori Identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SEM-4	1	0	1	78,33	20
			SYN-2					
			LOG-2					
1	6791	LOG-3	SEM-4	0	2	1	46,67	80
		SEM-5						
1	10304	LOG-2	LOG-3	0	1	2	0	50
			SEM-5					
2	156	COM-7	COM-7	1	0	1	100	90
			LOG-3					
2	8894	COM-3	COM-3	1	0	3	100	20
			COM-1					
			SYN-3					
			LOG-3					
2	9196	COM-6	COM-6	2	1	1	56,67	80
		LOG-3	LOG-3					
		SEM-5	LOG-4					
3	6185	SEM-1	LOG-5	1	1	1	10	0
		LOG-3	LOG-3					
3	9767	SEM-6	SEM-6	2	1	1	1,67	0
		SYN-6	SYN-6					
		INV	LOG-3					
3	11277	SEM-1	SEM-6	2	1	1	20	20
		LOG-3	LOG-3					
		LOG-5	LOG-5					
3	12525	INV	LOG-5	0	1	3	68,33	20
			LOG-6					
			SEM-6					
3	4131		SYN-6	0	0	2	100	90
			LOG-3					
4	268	COM-1	SYN-5	0	1	3	83,33	40
			SEM-5					
			LOG-3					
4	2388	SYN-1	SYN-6	0	1	5	61,67	20
			LOG-3					
			LOG-5					
			SEM-5					
4	4163	COM-5	LOG-3	0	1	2	100	80
			COM-3					
4	5027	SYN-4	SEM-5	0	2	2	63,33	40
		LOG-4	SYN-6					
4	6303	COM-3	COM-5	0	1	2	93,33	90
			SEM-5					
4	12488	SEM-1	LOG-5	0	3	2	63,33	80
		COM-3	COM-3					
		SYN-5						
4	3809		LOG-3	0	0	2	100	80
			COM-3					
5	1965	COM-2	SYN-2	1	1	1	96,67	80
		COM-4	COM-4					
5	2564	COM-2	SYN-2	1	1	1	98,33	80
		COM-4	COM-4					
5	8876	SYN-2	SYN-2	1	0	1	90	80
			SYN-6					
6	3537	SYN-2		1	1	0	68,33	90
		LOG-3	LOG-3					
6	12321	COM-2	SYN-2	1	2	1	78,33	40
		COM-4	COM-4					
		LOG-2						

Zero-Shot Learning

Esercizio	ID Risposta	Errori	Errori identificati da LLM	Errori Correttamente Identificati	Errori Non Identificati	Errori Identificati in Eccesso	Voto Medio Docenti	Voto LLM
1	1104	SEM-4	SYN-6	0	1	3	78,33	70
			SYN-2					
			LOG-2					
1	6791	LOG-3	LOG-3	2	0	1	46,67	40
		SEM-5	LOG-5					
			SEM-6					
1	10304	LOG-2	SEM-3	0	1	2	75	80
			LOG-3					
2	156	COM-7		0	1	0	100	100
2	8894	COM-3	COM-3	1	0	2	100	60
			SYN-6					
2	9196		COM-5	2	1	1	56,67	50
		COM-6	COM-6					
		LOG-3	LOG-3					
		SEM-5	SEM-1					
3	6165	SEM-1	SEM-1	2	0	1	10	10
		LOG-3	LOG-3					
			LOG-5					
3	9767	SEM-6	SEM-6	2	1	2	1,67	0
		SYN-6	SYN-6					
		INV	COM-4					
			LOG-4					
3	11277	SEM-1	SEM-1	2	1	1	20	20
		LOG-3	LOG-3					
		LOG-5	COM-5					
3	12525	INV	SEM-1	0	1	4	68,33	20
			LOG-3					
			SEM-6					
			COM-5					
3	4131			0	0	0	100	100
4	268	COM-1	SYN-6	0	1	3	83,33	60
			LOG-3					
			SEM-6					
4	2388	SYN-1	SYN-6	0	1	3	61,67	40
			LOG-3					
			SEM-6					
4	4163	COM-5	SYN-6	0	1	2	100	80
			LOG-3					
4	5027	SYN-4	SYN-6	0	2	3	63,33	20
		LOG-4	SEM-6					
			LOG-3					
4	6303	COM-3		0	1	0	93,33	100
4	12498	SEM-1	LOG-3	0	3	1	63,33	60
		COM-3						
		SYN-5						
4	3809		LOG-3	0	0	1	100	80
5	1965	COM-2	SEM-6	0	2	2	96,67	20
		COM-4	LOG-3					
5	2564	COM-2	SEM-6	0	2	2	98,33	40
		COM-4	LOG-3					
5	8876	SYN-2	SYN-6	0	1	2	90	80
			SEM-1					
6	3537	SYN-2	SYN-6	0	2	2	68,33	90
		LOG-3	SEM-1					
6	12321	COM-2	SYN-6	1	2	2	78,33	0
		COM-4	SEM-6					
		LOG-2	LOG-2					

9. Bibliografia

- Cagliero, L., Farinetti, L., Fior, J., & Manenti, A. I. (2024). *ChatGPT, Be my teaching assistant! Automatic correction of SQL exercises* (pp. 81–87).

<https://doi.org/10.1109/compsac61105.2024.00021>

- Ahadi, A., Prior, J., Behbood, V., & Lister, R. (2016). *Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries*. <https://doi.org/10.1145/2899415.2899464>

- Taipalus, T., Siponen, M., & Vartiainen, T. (2018). Errors and complications in SQL query formulation. *ACM Transactions on Computing Education*, 18(3), 1–29.

<https://doi.org/10.1145/3231712>

10. Strumenti e Tecnologie Utilizzate

Modelli di Machine Learning

GPT-4o: <https://chatgpt.com/>

Gemma2B: <https://huggingface.co/google/gemma-2b>

Codestral (utilizzato sulla piattaforma di Mistral AI): <https://mistral.ai/>

Llama 3.1 Sonar (utilizzato sulla piattaforma di Perplexity Labs): <https://labs.perplexity.ai/>

Altri Strumenti

LMStudio: <https://lmstudio.ai/>

Dbdiagram.io: <https://dbdiagram.io/>

Microsoft Excel

11. Ringraziamenti

Un ringraziamento speciale alla mia famiglia, per il loro supporto incondizionato e la fiducia costante che mi hanno trasmesso, rendendo possibile ogni passo di questo percorso.

Ringrazio il Prof. Cagliero e alla Prof.ssa Farinetti per avermi guidato con dedizione e competenza nella fase più importante del mio percorso accademico.

Un pensiero speciale va ai miei amici e coinquilini, che hanno saputo portare leggerezza e allegria anche nei momenti più complessi e impegnativi.

Infine, ringrazio tutti i docenti che ho avuto il privilegio di conoscere e ascoltare in questi anni, per aver contribuito alla mia crescita personale e accademica.