

HODI (ET AMO) – *In medio stat virtus*

Erica Solinas, Fabiana Capozzi

1. Introduzione

Negli ultimi anni, i Large Language Models (LLM) hanno rivoluzionato il panorama dell'elaborazione del linguaggio naturale, soprattutto grazie al *prompting* come strategia di interazione e guida del modello. Si tratta di una tecnica che consiste nel formulare istruzioni in linguaggio naturale e indurre il modello a svolgere un compito specifico. Il prompting ha reso possibile, quindi, affrontare una vasta gamma di task senza la necessità di un addestramento supervisionato tradizionale.

All'interno di questo contesto innovativo, il presente progetto si propone di esplorare l'efficacia del prompting nell'ambito dell'*hate speech detection*, più in particolare dell'*homotransphobia detection*.

2. Obiettivi

Il focus del progetto è duplice: da un lato, si intende analizzare la capacità dei modelli di identificare contenuti discriminatori, offensivi o denigratori nei confronti della comunità LGBTQIA+; dall'altro, si vogliono mettere in luce le differenze tra modelli specializzati nella lingua italiana e modelli generalisti multilingua. La metodologia adottata si basa esclusivamente sul prompting, senza alcuna fase di fine-tuning o addestramento supervisionato: i modelli vengono interrogati direttamente attraverso istruzioni formulate ad hoc per ogni modello, progettate per massimizzare la chiarezza interpretativa e la coerenza dell'output. Ogni prompt è stato calibrato per mettere ciascun modello nelle condizioni ottimali di comprendere il compito richiesto e fornire una classificazione il più possibile accurata. Abbiamo scelto di valutare e confrontare le prestazioni di quattro modelli linguistici: due modelli italiani (Camoscio-7B e Phi-3.5 mini - ITA) e due modelli multilingua (GPT-4.1 mini e Gemini 2.0 Flash).

3. Dataset

Il riferimento della presente ricerca è stato il task HODI (Homotransphobia Detection in Italian) presentato ad Evalita 2023, dunque abbiamo utilizzato lo stesso dataset. HODI si articola in due subtask:

- **Subtask A – Homotransphobia detection:** un task di classificazione binaria in cui i modelli hanno l'obiettivo di classificare ogni testo come contenente o meno odio verso la comunità LGBTQIA+. Il compito è assegnare l'etichetta [0] se il testo non contiene odio, [1] se contiene odio.

- **Subtask B – Explainability:** una volta classificato un testo come contenente odio, l'obiettivo è identificare le ragioni di tale classificazione, ossia le porzioni di testo riconosciute come omotransfobiche.

Per realizzare il task è stato creato un dataset apposito, i cui dati sono stati raccolti da Twitter con un approccio basato su parole chiave, nel periodo compreso tra il 1° maggio 2022 e il 31 agosto 2022. La raccolta si è focalizzata sia su parole chiave comunemente usate in contesti omotransfobici, sia su parole chiave relative ad eventi specifici che coinvolgono o interessano direttamente la comunità LGBTQIA+ (ad esempio: Pride, DDL Zan). Il training set è composto da 5000 tweet, mentre il test set da 1000 tweet. Ogni dataset contiene un identificativo univoco (“*id*”) e una colonna con il testo del tweet (“*text*”).

Il nostro lavoro si concentra principalmente sul task di classificazione del Subtask A, e nel prossimo paragrafo ne verranno spiegate le ragioni.

4. Metodo: prompting e modelli

Esistono diverse strategie di prompting, ma non tutti i prompt sono efficaci allo stesso modo per ogni modello: per valutare correttamente la capacità di un LLM di svolgere un task, è fondamentale adattare il prompt alle caratteristiche specifiche del modello. Per questo, per ciascun modello impiegato abbiamo elaborato un prompt personalizzato, con l'obiettivo di metterlo nelle condizioni ottimali per fornire la miglior performance possibile.

La selezione dei modelli non è stata immediata, sia per vincoli computazionali e pratici, dal momento che non potevamo eseguire i modelli in locale ed eravamo limitate all'utilizzo tramite API o Google Colab; sia perché avevamo l'esigenza di trovare modelli in grado di svolgere compiti di classificazione esclusivamente tramite prompting.

Alla luce di queste necessità e dopo alcuni test, abbiamo scelto di confrontare due modelli di grandi dimensioni e allo stato dell'arte, GPT e Gemini, entrambi capaci di operare in italiano con buone prestazioni, sia nel subtask A di classificazione che nel subtask B di *explainability*; e due modelli specializzati nella lingua italiana, ma di dimensioni più contenute, Camoscio e Phi. Questi ultimi hanno mostrato maggiore sensibilità alla formulazione del prompt: a differenza dei modelli più avanzati, richiedevano istruzioni più brevi e strutturate in modo molto semplice e diretto, e non riuscivano a gestire entrambi i task all'interno dello stesso prompt. Vediamoli nel dettaglio.

4.1 GPT 4.1 Mini e Gemini 2.0 Flash

Il modello GPT-4.1 mini fa parte della famiglia GPT-4.1 sviluppata da OpenAI. Si tratta di una versione più leggera e veloce rispetto al modello completo, grazie a un numero ridotto di parametri e a una struttura ottimizzata, pur mantenendo gran parte delle capacità linguistiche del modello full-size. Come gli altri modelli della stessa famiglia, è basato su un'architettura transformer decoder-only. Nel nostro progetto, è stato utilizzato tramite API.

Gemini 2.0 Flash, invece, è uno dei modelli multimodali di intelligenza artificiale sviluppati da Google DeepMind. Tramite *Google AI Studio* è possibile interrogare il modello e testare l'efficacia del prompt, la correttezza delle risposte e modificare parametri come la temperatura, il top P e la lunghezza dell'output. Nella fase successiva, l'API Key richiesta a Google permette di avere 1.500 richieste giornaliere da inoltrare al modello direttamente in locale.

Tra le strategie di prompting suggerite, sono state di ispirazione le seguenti:

- istruzioni chiare e dirette
- aggiungere prefissi che scandiscano il prompt
- assegnazione di un ruolo
- aggiungere informazioni contestuali
- includere esempi che spingano all'identificazione di pattern
- indicare al modello cosa fare e cosa non fare
- fornire il formato della risposta
- sfruttare la capacità di completamento del modello.

Il prompt ottimale per GPT 4.1. Mini e Gemini 2.0 Flash è stato selezionato dopo vari test svolti su *OpenAI Platform* e *Google AI Studio*.

❖ Prompt iniziale:

###Istruzione:

Sei un linguista specializzato nell'identificazione di contenuti omotransfobici in tweet in lingua italiana. Il tuo compito è leggere ogni tweet e determinare con un'etichetta binaria se il contenuto contiene odio omotransfobico o meno. Considera come odio omotransfobico qualsiasi espressione di pregiudizio, stereotipo, discriminazione, disprezzo, derisione, incitamento all'odio o alla violenza verso persone LGBTQIA+ (esempio: appellativi quali 'finocchio', 'frocio', 'derivato', 'effeminato'). NON considerare omotransfobico l'uso di linguaggio volgare o offensivo che non sia a tema.

Procedura da seguire:

1. Assegna l'etichetta:

[0]: se il tweet NON contiene odio omotransfobico

[1]: se il tweet CONTIENE odio omotransfobico

2. SOLO se assegni l'etichetta [1], formula una breve spiegazione nel formato: "Il tweet contiene odio omotransfobico perché

[spiegazione]"; lascia vuoto in caso contrario.

Pensa ad ogni step!

Di seguito cinque esempi del formato che devi restituire [tweet, etichetta, spiegazione].

###Esempio 1:

Tweet: ve l'ho mai detto che i gau fanno schifo [URL]

Etichetta: 1

Spiegazione: Il tweet contiene odio omotransfobico perché l'espressione "i gay fanno schifo" è gravemente offensiva.

###Esempio 2:

Tweet: minuto 20 del primo episodio e mi è già partita a milla la ship tra Leah e Fatih. a volte odio essere così frocia

Etichetta: 0

Spiegazione:

###Esempio 3:

Tweet: Non hai abortito ed é nata una bambina rompicoglioni con una malattia non curabile + frocia non sei contenta mamma [URL]

Etichetta: 1

Spiegazione: Il tweet contiene odio omotransfobico perché l'appellativo 'frocia' viene preceduto da un '+', che indica un'aggiunta peggiorativa: essere 'frocia' viene messo sullo stesso piano o addirittura in coda peggiorativa a condizioni già etichettate come "negative".

###Esempio 4:

Tweet: I VERI BASTARDI VOTANO PER IL GAY PRIDE RUMENO!

Etichetta: 1

Spiegazione: Il tweet contiene odio omotransfobico perché la frase è omofoba perché associa in modo diretto il sostegno al Gay Pride a un insulto morale ("bastardi"), denigrando chi sostiene i diritti LGBTQIA+

###Esempio 5:

Tweet: i gay che non la votano sono omofobi [URL]

Etichetta: 0

Spiegazione:

Adesso, sulla base del formato sopra illustrato:

Tweet:

Etichetta:

Spiegazione:

Punti chiave di questo prompt: assegnazione del ruolo, spiegazione del contesto e soprattutto di ciò che s'intenda per "odio omotransfobico", esempi scelti per essere indicativi e descrizione del formato da restituire.

Il prompt risulta in generale efficace, ma ha bisogno di miglioramenti per casi come il seguente, in cui non c'è comprensione completa del tweet e l'etichetta restituita è scorretta:

Tweet: @user_abcdef Sia chiaro, "ricchione" è meno omofobo di "froccio".

Etichetta: 1

Spiegazione: il tweet contiene odio omotransfobico perché, pur cercando di fare una distinzione, normalizza e utilizza due appellativi omofobi come "ricchione" e "froccio".

❖ Prompt intermedio:

Fra il punto "1" e il punto "2" viene aggiunto il testo seguente:

NON basarti solo su singole parole, ma valuta:

- il significato globale del tweet
- il tono usato (offensivo, ironico, denigratorio, neutro o positivo).

Fai attenzione a fasi ironiche, sarcastiche o ambigue: analizza il messaggio nel suo insieme.

Punti chiave: viene aggiunta la richiesta esplicita di guardare a tutto il contesto del tweet e di far attenzione al tono (fondamentale nel caso dei tweet e in generale per il task in questione).

Il prompt risulta ancora non abbastanza adatto, per questo viene migliorato ancora una volta.

❖ Prompt finale:

Nella fase di descrizione del compito è stato aggiunto il testo che segue:

Il tuo compito è leggere ogni tweet e per ognuno PRIMA valuta se contiene odio omotransfobico o meno, POI classifica con:

[0]: se il tweet NON contiene odio omotransfobico

[1]: se il tweet CONTIENE odio omotransfobico.

Punti chiave: viene invertito l'ordine etichetta – spiegazione per costringere il modello a ragionare prima di assegnare un'etichetta e rendere quest'ultima auspicabilmente più corretta.

Visto il buon livello di risposta al prompt, abbiamo deciso di affidare loro entrambi i subtask del dataset HODI: il subtask A (classificazione) e il subtask B (spiegazione). Per il secondo, invece di chiedere al modello di identificare i token esatti che motivavano la classificazione, abbiamo scelto un approccio più flessibile, invitandolo a fornire una spiegazione discorsiva “con parole sue”. Di seguito un esempio di risposta da GPT:

Tweet: “@user_abcdefg @user_abc Quasi quasi è meglio femminiello!”

Task A: [1]

Task B: Il termine "femminiello" in contesti offensivi o denigratori può essere usato come insulto omotransfobico nei confronti di persone LGBTQIA+, in particolare persone transgender o effeminate. L'espressione "Quasi quasi è meglio femminiello!" ha un tono denigratorio e suggerisce una preferenza sprezzante verso questa identità, configurando odio omotransfobico.

Nonostante le buone prestazioni generali, GPT ha mostrato alcune fragilità, in particolare nel gestire tweet caratterizzati da ironia o sarcasmo, oppure nel distinguere tra odio generico e odio specificamente omotransfobico. Per cercare di compensare queste lacune, abbiamo aggiunto al prompt ulteriori istruzioni, mirate a orientare meglio l'attenzione del modello sul tipo di contenuto da individuare, ma il modello tende comunque a focalizzarsi solo sulla parola offensiva, senza considerare il contesto, come in questo esempio:

Tweet: “io: è bono eh papà: ma non eri frocia te 🤪🤪🤪🤪”

Task A: [1]

Task B: Il tweet utilizza il termine "frocia" in modo dispregiativo e derisorio, accompagnato da emoji che accentuano il tono offensivo e denigratorio verso una persona presumibilmente LGBTQIA+. Questo costituisce odio omotransfobico.

Per quanto riguarda gli iperparametri, abbiamo scelto per GPT temperatura 0.0 per limitare la variabilità nell'assegnazione dell'etichetta e un valore di temperatura di 0.7 per Gemini.

4.2. Camoscio -7B e Phi 3.5 Mini - ITA

Camoscio è un modello linguistico basato sull'architettura LLaMA e specificamente ottimizzato per l'italiano attraverso tecniche di instruction tuning. Più in particolare è la versione italiana del modello Stanford Alpaca, addestrata utilizzando la tecnica di Low-Rank Adaptation (LoRA). Questo modello è stato anche utilizzato dal team *extremITA* per eseguire il task HODI 2023 tramite prompting.

Phi, invece, è una versione fine - tuned, ottimizzata per performance migliori in italiano, del modello Phi 3.5 mini - instruct di Microsoft. Il modello conta 3.82 B di parametri ed è interrogabile tramite interfaccia web grazie ad una demo messa a disposizione dagli autori¹. Per eseguire il codice, invece, è stato utilizzato l'ambiente di Google Colab. Il modello sembra adatto a ricevere istruzioni tramite prompting per task in lingua italiana grazie al fine - tuning che è stato eseguito, ed è stato per questo selezionato come adatto a questo lavoro.

Sia per le dimensioni notevolmente minori rispetto a modelli come Gemini e GPT, sia per la difficoltà intrinseca del task di cui qui ci occupiamo, il prompt più funzionale per Camoscio e Phi ha delle caratteristiche in parte opposte:

- prompt estremamente conciso e diretto
- chiarezza maggiore nella descrizione delle etichette
- tecniche di zero - shot prompting.

I modelli si sono mostrati piuttosto confusi di fronte a prompt più lunghi e complessi e anche di fronte a degli esempi, che non aiutavano ad apprendere pattern, ma piuttosto complicavano il ragionamento. Inoltre, i modelli sono stati completamente incapaci di eseguire il subtask B quando gli è stato chiesto, generando spiegazioni anche nel caso dell'etichetta [0] oppure entrando in un loop creativo in cui lo stesso tweet veniva analizzato più volte di seguito, perciò, abbiamo scelto di assegnargli esclusivamente il compito di classificazione binaria (subtask A).

❖ Prompt finale:

###Istruzione:

Sei un linguista specializzato in analisi di testi per l'identificazione di odio omotransfobico, ovvero odio verso la comunità LGBTQ+.

Ti verrà fornito un tweet. Il tuo compito è:

- Restituire solo *1* se contiene odio omotransfobico.
- Restituire solo *0* se non lo contiene.

¹ <https://huggingface.co/spaces/anakin87/Phi-3.5-mini-ITA>.

Scrivi solo il numero, senza spiegazioni.

###Tweet:

{text}

###Etichetta:

Punti chiave del prompt: uso di prefissi, assegnazione del ruolo, spiegazione molto concisa del contesto e del compito, richiesta di non dare spiegazioni, utilizzo della capacità di completamento.

Già durante una preliminare analisi dei risultati abbiamo osservato delle criticità. Per esempio, Camoscio tendeva a classificare come omotransfobici anche messaggi offensivi, ma privi di riferimenti specifici alla comunità LGBTQIA+. Allo stesso tempo, non era in grado di riconoscere il sarcasmo o i casi più impliciti di discriminazione. In generale, il modello tendeva ad assegnare l'etichetta [1] a qualsiasi frase contenente parolacce o insulti espliciti, come ad esempio:

Tweet: @user_a brava Mara come sempre puntatona.. Sei la zia di tutti noi gay che tanto ti amiamo. Grazie. Commovente. Teocoli e ritorno di Roberta Capua che dovrebbe rimanere in Italy e non. Tornare in Florida. ...cervelli in fuga purtroppo accade troppo. Spesso ora Mara riposati

Task A: [1]

Tweet: Però loro non danno fastidio, ragazzi che si divertono a disturbare e sporcare un condominio. Spero che si caghino tutti addosso.

Task A: [1]

Per quanto riguarda gli iperparametri, per Phi abbiamo scelto temperatura 0.7.

5. Metriche ed evaluation

Considerando soltanto il subtask di classificazione A, i modelli utilizzati hanno raggiunto i seguenti livelli di accuratezza:

- GPT: 0.748
- Gemini: 0.77
- Camoscio: 0.488
- Phi: 0.49.

Nello specifico, GPT ha ottenuto una *precision* del 72,8 %, un *recall* del 77,3 % e un *f1-score* del 75,0% per la classe ‘0’; *precision* di 76,9%, *recall* di 72,4% e *f1-score* di 74,6% per la classe ‘1’. Inoltre, ha ottenuto un F1 Macro del 74,8 %.

Gemini ha ottenuto una *precision* del 80,2%, un *recall* del 70,9% e un *f1-score* del 75,3% per la classe ‘0’; *precision* di 75,0%, *recall* di 83,2% e *f1-score* di 78,9% per la classe ‘1’. F1 Macro: 77,1%.

I risultati di Camoscio e Phi vengono riportati ai fini di un confronto utile con gli altri due modelli, ma sono molto al di sotto delle performance desiderate.

Camoscio ha ottenuto una *precision* del 45,6 %, un *recall* del 22,3 % e un *f1-score* del 29,9 % per la classe ‘0’; *precision* di 49,9 %, *recall* di 74,4 % e *f1-score* di 59,7 % per la classe ‘1’. F1 Macro: 44,8%.

Phi ha ottenuto una *precision* del 49,0 %, un *recall* del 95,9 % e un *f1-score* del 64,8 % per la classe ‘0’; *precision* di 52,4 %, *recall* di 0,43 % e *f1-score* di 0,80 % per la classe ‘1’. F1 Macro: 36,4%.

Model	precision '0'	recall '0'	f1-score '0'	precision '1'	recall '1'	f1-score '1'	F1 Macro
GPT 4.1 MINI	72,8 %	77,3%	75,0%	76,9%	72,4%	74,6%	74,8%
GEMINI 2.0 FLASH	80,2%	70,9%	75,3%	75,0%	83,2%	78,9%	77,1%
CAMOSCIO - 7B	45,6%	22,3%	29,9%	49,9%	74,4%	59,7%	44,8%
PHI 3.5 MINI - ITA	49,0%	95,9%	64,8%	52,4%	0,4%	0,8%	36,4%

Una premessa è necessaria: il task risulta essere estremamente soggettivo e dai confini sfumati soprattutto per alcuni tweet; inoltre, il dataset Gold standard è il risultato dell’annotazione di tre umani, a cui, non sempre, è facile aderire completamente.

Nonostante i numerosi test eseguiti con Camoscio e Phi per permettere loro di identificare correttamente l’etichetta, un task di classificazione come questo è forse una sfida troppo ambiziosa.

5.1. Confronto tra modelli

GPT mostra performance abbastanza equilibrate tra le due classi ed una performance complessiva affidabile. Gemini mostra un *recall* molto elevato per la classe ‘1’, perciò

intercetta chiaramente gli esempi di questa classe, e un *recall* più basso per la classe '0', mostrando di essere più prudente e di voler evitare falsi positivi. In generale, è stato anche più performante di GPT.

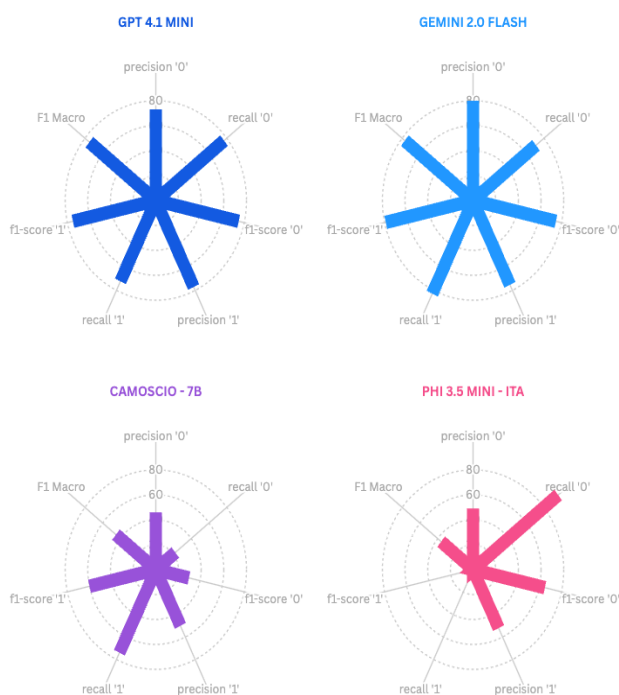
Camoscio e Phi mostrano prestazioni deboli e sbilanciate. Camoscio manca la maggior parte degli esempi di classe '0' (*recall* al 22,3%) e ha una *precision* bassa per la classe '1', perciò, pecca di molti falsi positivi. In conclusione, assolutamente inefficace al compito. Phi ha prestazioni alquanto problematiche, classificando praticamente quasi tutti i tweet con l'etichetta '0' (da qui deriva anche una *precision* discreta e un buon f1-score) e non riesce a riconoscere quasi mai la classe '1'. Il modello risulta altamente insufficiente.

Queste differenze molto marcate dipendono da un insieme di fattori:

- architettura del modello
- dati di addestramento
- fine-tuning.

GPT e Gemini sono modelli strutturalmente capaci di gestire task complessi, addestrati su vasti dataset multilingua e multi - dominio, motivo per cui funzionano bene anche per task in lingua italiana. Camoscio, e ancora di più Phi, sono modelli con molti meno parametri, strutturalmente più piccoli e con capacità limitate di generalizzazione, anche a causa dei dati di addestramento utilizzati, solitamente più ristretti e specifici.

Per una visione d'insieme:



5.2. Evaluation rispetto ad HODI 2023

Dalla documentazione ufficiale di HODI 2023² proviene la seguente tabella, con la rappresentazione dei team che hanno partecipato al task, i modelli utilizzati e la descrizione delle tecniche:

Team	Country	Task	dbmdz-BERT-Italian	AIBERTO	Open AI Davinci	IT5	Camoscio	UmBERTO - Oscar Corpus	Twitter XML-R Sentiment	Fine-tuning	Knowledge injection	Data augmentation	Multi-task Learning	Few-shot Learning	Feature Extraction	Prompting
DH-FBK [19]	IT	A, B	✓							✓		✓	✓			
CHILab [20]	IT	A		✓											✓	
extremITA [21]	IT	A, B				✓	✓									✓
O-Dang [22]	IT,UK	A		✓	✓					✓	✓					
LCTs [23]	ES,NL	A						✓		✓						
Team_Tamil [24]	IE,IN	A							✓					✓		

I risultati ottenuti sono i seguenti³:

Team	Macro F1	Rank
LCTs ₃	0.8108	1
LCTs ₂	0.8000	2
O-Dang ₁	0.7959	3
DH-FBK ₁	0.7950	4
extremITA ₂	0.7942	5
O-Dang ₂	0.7920	6
DH-FBK ₂	0.7837	7
O-Dang ₃	0.7804	8
LCTs ₁	0.7709	9
CHILab ₂ **	0.7525	-
CHILab ₃ **	0.7454	-
extremITA ₁	0.7431	10
CHILab ₁ **	0.7248	-
INGEOTEC ₁	0.7153	11
Team_Tamil ₁	0.6735	12
Baseline	0.6691	13
SOVRAG ₃	0.6634	14
SOVRAG ₂	0.6334	15
SOVRAG ₁	0.6108	16
CHILab ₃	0.5528	17
CHILab ₁	0.5205	18
CHILab ₂	0.5199	19

I modelli Camoscio e Phi sono assolutamente non comparabili con i risultati della ricerca, mentre GPT e Gemini si collocano con le loro percentuali (rispettivamente 74,8% e 77,1%) abbastanza in alto nel ranking.

² Tabella 4 di Nozza D., Cignarella A., Damo G., Caselli T., Patti V., *HODI at EVALITA 2023: Overview of the first Shared Task on Homotransphobia Detection in Italian*, 2023.

³ Tabella 5 di Nozza D., Cignarella A., Damo G., Caselli T., Patti V., *HODI at EVALITA 2023: Overview of the first Shared Task on Homotransphobia Detection in Italian*, 2023.

6. Conclusioni

Il lavoro qui presentato aveva l'obiettivo di condurre il task di *Homotransphobia detection* su tweet in lingua italiana, presentato per la prima volta durante Evalita 2023, utilizzando tecniche di prompting e diversi modelli linguistici. I modelli scelti sono stati Camoscio, Phi, GPT e Gemini e le tecniche più funzionali il few - shot prompting per GPT e Gemini e lo zero - shot prompting per Camoscio e Phi.

Gli esempi utilizzati per il few - shot prompting sono stati selezionati dal dataset di training in modo conforme e strategico rispetto al task e sono risultati funzionali alla classificazione. In particolare, sono stati scelti per la classe '1': un esempio esplicito di odio omotransfobico (ve l'ho mai detto che i gay fanno schifo [URL]), un esempio riferito in generale ad un evento a tema (I VERI BASTARDI VOTANO PER IL GAY PRIDE RUMENO!) e un esempio implicito/velato (Non hai abortito ed é nata una bambina rompicoglioni con una malattia non curabile + frocia non sei contenta mamma [URL]). Per la classe '0': un esempio con tono ironico e per questo non discriminatorio (minuto 20 del primo episodio e mi è già partita a milla la ship tra leah e fatin. a volte odio essere così frocia) e un esempio in cui è presente una parola chiave, ma non in un contesto discriminatorio (i gay che non la votano sono omofobi [URL]).

Come mostrano chiaramente le performance, nonostante tutti i modelli siano stati messi nelle condizioni di eseguire correttamente la classificazione, Gemini e GPT sono stati capaci di una comprensione più sottile, capace di cogliere la maggior parte degli esempi impliciti e ironici, oltre che gli esempi espliciti, e di collocarsi allo stato dell'arte. Camoscio e Phi sono risultati assolutamente inadeguati e necessitano di una specializzazione sul task. Questa impressione è confermata dal paper che fa da riferimento allo studio, in cui i modelli istruiti tramite prompting (Camoscio e iT5) hanno ricevuto un fine-tuning sui dataset del task ed infatti raggiungono performance molto più alte.

Bibliografia

- Nozza D., Cignarella A., Damo G., Caselli T., Patti V., *HODI at EVALITA 2023: Overview of the first Shared Task on Homotransphobia Detection in Italian*, 2023.
- GPT 4.1 mini documentation: <https://platform.openai.com/docs/models/gpt-4.1-mini>
- Camoscio 7.b Llama documentation: <https://huggingface.co/teelinsan/camoscio-7b-llama>
- Santilli, Rodolà, *Camoscio: an Italian Instruction-tuned LLaMA*, 2023
- Gemini 2.0 Flash documentation: <https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash>
- Phi 3.5 Mini ITA documentation: <https://huggingface.co/anakin87/Phi-3.5-mini-ITA>.