Relazione tecnica sul sistema di clustering gerarchico e visualizzazione di embedding testuali

1 Introduzione

Il codice presentato realizza una pipeline completa per l'analisi, la segmentazione e la visualizzazione di un insieme di embedding testuali, ottenuti da un modello di linguaggio. Lo scopo principale è identificare gruppi di risposte che presentano caratteristiche semantiche simili, attraverso un approccio di **clustering gerarchico a due livelli**, con successiva **visualizzazione bidimensionale** dei risultati.

La pipeline è suddivisa in più blocchi funzionali, ciascuno dei quali realizza una fase distinta della procedura. Di seguito viene descritta in dettaglio la logica seguita in ciascuna fase.

2 Fase 1: Estrazione degli embedding

Viene letto un file `JSON` contenente una lista di risposte testuali. Per ciascuna di esse viene calcolato l'**embedding vettoriale** utilizzando un modello di linguaggio neurale (ad esempio BERT o simili). L'embedding rappresenta la risposta come punto in uno spazio numerico ad alta dimensione (es. 768 dimensioni), riflettendo le sue caratteristiche semantiche.

Il risultato è una matrice $X \in \mathbb{R}^{n \times d}$, dove n è il numero di risposte e d la dimensione degli embedding.

3 Fase 2: Clustering al primo livello

Sul dataset di embedding X viene applicato un algoritmo di clustering **K-Means** con $k = 2$, con l'obiettivo di suddividere l'intero insieme in due gruppi principali, detti *macro-cluster*.

L'assunzione alla base è che esistano due classi semantiche predominanti, ad esempio:

- `jailbreak` (risposte che aggirano restrizioni)
- `no-jailbreak` (risposte conformi e sicure)

L'identificazione semantica di ciascun cluster viene effettuata manualmente ispezionando alcuni esempi per stabilire quale dei due gruppi rappresenti ciascuna classe.

4 Fase 3: Clustering gerarchico a livello locale

Ciascun macro-cluster identificato viene ulteriormente analizzato per rivelare eventuali **sottostrutture interne**. L'idea è che, anche all'interno delle classi principali, possano esistere comportamenti o strategie differenti.

4.1 Scelta del numero di sottocluster

Per ciascun macro-cluster:

1. Viene isolato il sottoinsieme degli embedding appartenenti a quel gruppo.
2. Viene eseguito K-Means per un intervallo di valori $k \in [2, 6]$.
3. Vengono calcolate due metriche:
   - **Inertia (somma delle distanze intra-cluster)** – utilizzata per l'*Elbow method*;
   - **Silhouette score** – misura quanto bene ciascun punto è assegnato al proprio cluster rispetto agli altri.
4. Viene scelto il numero ottimale di cluster come quello che massimizza il Silhouette score.

Il risultato finale è un insieme di sottocluster specifici per ciascun macro-cluster.

5 Fase 4: Visualizzazione bidimensionale dei risultati

Dato che gli embedding risiedono in uno spazio ad alta dimensionalità, non sono facilmente interpretabili visivamente. Si procede quindi con una **riduzione dimensionale a 2D** per consentire la visualizzazione:

- **PCA** (Principal Component Analysis) – metodo lineare.
- **t-SNE** (t-distributed Stochastic Neighbor Embedding) – metodo non lineare, specializzato per mantenere vicinanze locali.
- **UMAP** (Uniform Manifold Approximation and Projection) – simile a t-SNE, ma più efficiente e con migliore preservazione della struttura globale.

5.1 Visualizzazione dei macro-cluster

Viene effettuata la proiezione in 2D degli embedding completi. Ogni punto è colorato secondo il macro-cluster di appartenenza. I centroidi dei due gruppi vengono evidenziati per mostrare la separazione complessiva.

5.2 Visualizzazione globale dei sottocluster

In un unico grafico vengono rappresentati tutti i punti dell'insieme, ma colorati secondo il sottocluster gerarchico assegnato. Le etichette includono sia il macro-cluster di partenza sia l'indice del sottocluster (es. `jailbreak-1`).

5.3 Visualizzazione locale dei sottocluster

Per ciascun macro-cluster viene prodotto un grafico separato che mostra solo i sottocluster interni, con centroidi e distribuzione spaziale nel piano ridotto.

6 Finalità e vantaggi della pipeline

La strategia adottata consente di:

- Separare in modo grossolano risposte di natura potenzialmente dannosa da quelle sicure;

- Esplorare la diversità interna di ciascun gruppo, rivelando sottotipi semantici (es. diversi approcci di jailbreak);
- Analizzare visivamente la struttura latente dei dati in uno spazio interpretabile;
- Costruire una rappresentazione gerarchica interpretabile dell'universo delle risposte.

7 Considerazioni finali

L'approccio proposto, pur utilizzando metodi noti come K-Means, Silhouette analysis e riduzioni dimensionali, si distingue per l'applicazione strutturata in due livelli e per l'integrazione sinergica tra clustering e visualizzazione. Questo consente non solo di rilevare pattern quantitativi, ma anche di interpretare tali pattern in modo qualitativo e semantico, facilitando l'analisi di risposte generate da modelli di linguaggio.