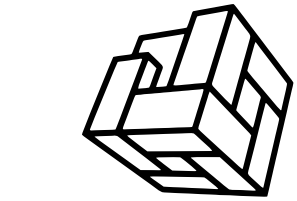
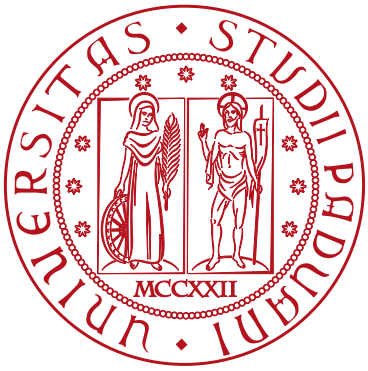


PEBKAC
Gruppo: 11
Email: pebkacswe@gmail.com
Docs: https://pebkac-swe-group-11.github.io
GitHub: https://github.com/PEBKAC-SWE-Group-11



Università degli Studi di Padova
Corso di Laurea: Informatica
Corso: Ingegneria del Software
Anno Accademico: 2024/2025

Verbale Esterno
7 Gennaio 2025

Informazioni sul documento:

Responsabile	Davide Martinelli
Verificatore	Matteo Gerardin
Redattore	Davide Martinelli
Uso	Esterno
Destinatari	Vimar S.p.A.
	Tullio Vardanega
	Riccardo Cardin

Abstract:

Quarta riunione SAL del gruppo con Vimar S.p.A in cui abbiamo illustrato i nostri progressi e chiarito alcuni dubbi riguardo a tecnologie e problematiche riscontrate durante il periodo natalizio.

Registro delle versioni

Versione	Data	Autore	Ruolo	Descrizione
1.0.0	2025-01-11	Davide Martinelli	Responsabile	Approvazione e rilascio
0.1.0	2025-01-10	Matteo Gerardin	Verificatore	Verifica
0.0.1	2025-01-10	Davide Martinelli	Responsabile	Stesura

Indice

1 Informazioni generali

- **Tipo riunione**: esterna;
- **Luogo**: telematica, Teams;
- **Data**: 2025-01-07;
- **Ora inizio**: 15:00;
- **Ora fine**: 16:00;
- **Presenti**:
  - Alessandro Benin
  - Ion Bourosu
  - Matteo Gerardin
  - Derek Gusatto
  - Davide Martinelli
  - Matteo Piron
  - Tommaso Zocche
  - ⋆ Mariano Sciacco (Vimar S.p.A.)
  - ⋆ Francesca Stival (Vimar S.p.A.)
- **Assenti**:

2 Riassunto della riunione

Come di consueto, la riunione si è svolta con il seguente ordine:

- **Presentazione dei ruoli:** vengono elencati i ruoli assunti dai membri del gruppo in questo periodo:
  - Davide Martinelli - Responsabile;
  - Alessandro Benin - Analista;
  - Tommaso Zocche - Amministratore/Programmatore;
  - Matteo Piron - Progettista;
  - Matteo Gerardin - Verificatore;
  - Derek Gusatto - Programmatore;
  - Ion Bourosu - Programmatore;
- **Panoramica ad ampio spettro**: abbiamo pianificato e fissato l'obiettivo per il 15 gennaio 2025, ovvero una revisione in azienda del lavoro svolto in questo periodo. Il prodotto proposto comprende un applicativo utile a dimostrare la capacitá di utilizzare le tecnologie e la documentazione accessoria. In seguito abbiamo esplicato alcuni dubbi nati da riflessioni fatte durante il periodo natalizio.
- **Attività completate ed in corso**: In questo periodo ci siamo impegnati a:
  - Sviluppare la parte di chunking semantico;
  - Iniziare lo sviluppo di un'interfaccia grafica;
  - Completare l'implementazione di Llama 3.2 1B.
- **Prossime attività da svolgere**: nel prossimo periodo si svolgeranno le seguenti attività:
  - Completare l'implementazione del chunking semantico;
  - Completare la scrittura dell'interfaccia grafica;
  - Predisporre delle API utili al funzionamento dell'applicativo e consecutiva unione di frontend e backend (in vista della revisione);
  - Verificare e rilasciare la documentazione scritta finora.
- **Discussione di dubbi e domande**:

  L'applicativo prevede un'area riservata al solo Admin, ma dall'immagine di esempio per la potenziale interfaccia dell'applicativo (Figura 6 del capitolato) sembra che il pulsante per accedere all'area riservata sia visibile a tutti gli installatori. É corretto?
  Nonostante nel disegno di esempio dell'applicativo fornito dall'azienda sia presente un pulsante adibito all'area di login, dopo la consultazione del proponente è emerso che non è necessario fornire un pulsante apposito per l'autenticazione degli amministratori, bensì scegliamo di separare la pagina principale dell'applicativo dal login degli amministratori.

L'admin, una volta che ha acceduto alla propria area riservata, deve essere in grado di vedere una serie di statistiche sull'utilizzo dell'applicativo da parte degli installatori. Per visualizzare tali statistiche, ci deve essere un'area del DB dedicata al salvataggio di dati specifici relativi alle statistiche che si vogliono visualizzare. Se l'applicativo deve funzionare in locale, com'è possibile fare in modo che i dati di interesse per le statistiche possano essere scritti nel DB e, in seguito, siano visibili all'admin?
L'idea che caratterizza il progetto è quella di disporre dello stack applicativo (prodotto a cui stiamo tuttora lavorando) con alla base un server per la condivisione dei dati. Il senso di una struttura del genere e la richiesta di creare un'app che funzioni principalmente in locale risiedono nel fatto che, qual'ora si decidesse di mettere in produzione il sito, l'applicativo si appoggia ad un server. Al mero utilizzatore dell'applicazione verrà fornito un semplice link per collegarsi al server e fare le domande. Inoltre si richiede che l'app non utilizzi servizi esterni (come l'interrogazione di altri modelli oltre a quello preso in considerazione dall'applicativo).

È previsto un account di accesso per l'installatore? Se sì, in che modo l'installatore ottiene le proprie credenziali? È prevista una modalità di registrazione o si occupa l'azienda di distribuire le credenziali?
No, il sito è liberamente accessibile. L'utente ideale (guest) è un installatore che fa delle domande al sito. La gestione della cronologia delle chat senza le credenziali consiste nell'avere una sessione diversa per ogni dispositivo di accesso e, pertanto, se l'utente fa delle domande da due dispositivi diversi, queste non saranno sincronizzate.

Nella "base" del PoC è stata utilizzata la versione da 1B di parametri di Llama 3.2. Pur essendo il modello più piccolo che abbiamo trovato (che implica anche che sia estremamente "scarso", soprattutto in italiano, e di certo non è adatto ad un MVP) ci risulta molto lento a causa della dimensione di embedding unita allo storico della chat che vengono aggiunti come contesto al prompt. Abbiamo provato anche Llama 3.1 (nella sua versione da 8B di parametri): le risposte sono più passabili ma i tempi di risposta diventano inaccettabili (il tutto assumendo di avere abbastanza RAM disponibile per caricare il modello, non scontato su PC con ∼8GB). Ha dei suggerimenti da darci su questo frangente? Perché al momento ci sembra utopistico riuscire a far girare in locale un modello che sia allo stesso tempo veloce ed efficace.
Nonostante tutte le limitazioni che abbiamo, dobbiamo assicurare al proponente che l'applicativo funzioni correttamente, quindi, anche con un modello poco potente come Llama 3.2 1B e nonostante l'applicativo richieda delle risposte corrette e precise, è utile capire se le limitazioni risiedano semplicemente in un limite fisico di risorse. Nel caso in cui il problema risieda in altro, come per esempio non riuscire a recuperare le informazioni in modo corretto, allora spetta a noi provvedere ad una soluzione.

All'interno del gruppo abbiamo visioni discordanti sul funzionamento dell'embedding. Al LLM, come contesto, vanno consegnati direttamente i vettori più vicini oppure le informazioni ad essi corrispondenti? In altre parole, assumendo di avere una tabella nel database "— ID — JSON — VECTOR —" , al LLM vanno date le informazioni contenute nella 2ª o nella 3ª colonna?

Il processo di embedding da noi proposto risulta corretto, pertanto, per una corretta formulazione della risposta da parte del LLM è utile fornire a quest'ultimo sia la domanda originale sia i chunk di informazioni prodotti dall'embedding. In questo modo la domanda originale viene arricchita da altri parametri, che concorrono a determinare il contesto corretto. I vettori, di per sé, non producono una risposta.

3 Todo

Durante la riunione sono emersi i seguenti task da svolgere.

Assegnatario	**Task Todo**
Davide Martinelli, Matteo Gerardin, Alessandro Benin	Completare stesura parziale della documentazione e rilasciarla
Ion Bourosu	Completare scrittura dell'interfaccia grafica
Tommaso Zocche, Matteo Piron	Completare implementazione del chunking
Davide Martinelli	Stesura Verbale Esterno 07/01/2024

Firma del referente Vimar S.p.A.: _____