

Proposta di Architettura

Chatbot E-commerce Multi-Agente

Tesi di Laurea in Natural Language Processing

Studente	Paolo Alberti (matr. 0070103)
Relatrice	Prof.ssa Cristina Giannone
Corso	Natural Language Processing
Data	Febbraio 2026

1. Scenario Applicativo

1.1 Dominio: E-commerce di Abbigliamento

Propongo di focalizzare il chatbot su un e-commerce verticale di **abbigliamento e accessori moda**. Questa scelta offre diversi vantaggi:

- **Ricchezza di attributi:** colore, taglia, materiale, stagione, stile, brand
- **Visual Recommending naturale:** gli utenti spesso cercano "qualcosa simile a questa foto"
- **Cross-selling intuitivo:** outfit completi, accessori coordinati
- **Dataset disponibili:** Amazon Fashion, H&M Personalized Fashion Recommendations (Kaggle)

1.2 Caso d'Uso Principale

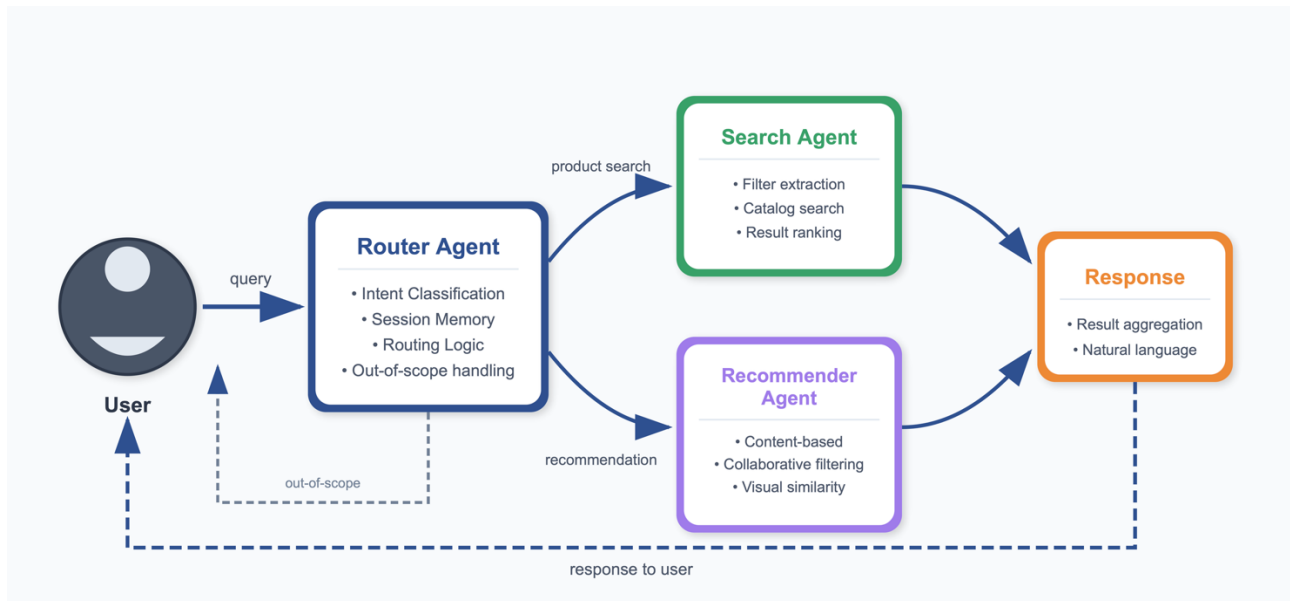
L'utente interagisce con il chatbot per:

1. **Ricerca guidata:** "Cerco una giacca invernale blu sotto i 100 euro"
2. **Consigli personalizzati:** "Cosa mi consigli da abbinare a questi pantaloni?"
3. **Visual search:** "Voglio qualcosa simile a questa foto" (elemento distintivo)

Scope del chatbot: Il sistema è focalizzato esclusivamente su ricerca e raccomandazione prodotti. Per richieste fuori ambito (es. spedizioni, resi, account), il Router risponde cortesemente indicando che può assistere solo nella ricerca e scelta dei prodotti.

2. Architettura Multi-Agente

L'architettura si basa su **LangGraph** per l'orchestrazione degli agenti, con un pattern **Router + Agenti Specializzati**. Il sistema è composto da **3 agenti** con responsabilità ben definite.



2.1 Schema Architetture

Agente	Responsabilità	Tecnologie
Router Agent	Classifica l'intent, gestisce il contesto della sessione, smista ai sotto-agenti, risponde a saluti e richieste fuori scope	LangGraph StateGraph, LLM (GPT-4 / Claude API), Intent Classifier
Search Agent	Ricerca prodotti nel catalogo basandosi su filtri estratti dalla query (colore, prezzo, categoria, taglia)	Elasticsearch / Vector DB (FAISS), NER per estrazione attributi
Recommender Agent	Suggerisce prodotti tramite content-based, collaborative filtering e visual similarity	LightFM (hybrid filtering), Trendyol DinoV2 per image similarity

2.2 Flusso di Interazione

Il **Router Agent** è il punto di ingresso. Riceve ogni messaggio dell'utente e:

4. Mantiene lo **stato della conversazione** (session memory) per gestire riferimenti contestuali
5. Classifica l'**intent** del messaggio (ricerca prodotto, richiesta consiglio, visual search, altro)
6. Instrada verso **Search Agent** o **Recommender Agent**
7. Risponde direttamente a saluti, chit-chat e richieste fuori scope
8. Aggrega e presenta le risposte in modo coerente

2.3 Gestione Contesto e Richieste Fuori Scope

Memoria conversazionale: Se l'utente dice "Cerco una giacca blu" e poi "La vorrei sotto i 100€", il Router comprende che si riferisce ancora alla giacca e arricchisce la query.

Richieste fuori scope: Per domande su spedizioni, resi, account o altri argomenti non pertinenti, il Router risponde: "Mi dispiace, posso aiutarti solo a cercare e consigliare prodotti. Per assistenza su [argomento], ti consiglio di contattare il servizio clienti."

3. Scelte Tecnologiche

3.1 LangGraph per l'Orchestrazione

LangGraph (estensione di LangChain) permette di definire il flusso come un **grafo di stati**, ideale per:

- **Routing condizionale:** decisioni basate sull'intent classificato
- **Stato persistente:** memoria della conversazione tra i nodi
- **Composizione modulare:** ogni agente è un nodo indipendente
- **Debugging:** tracciabilità del percorso decisionale

3.2 LightFM per il Recommending

LightFM implementa algoritmi **ibridi** che combinano:

- **Content-based:** raccomandazioni basate sugli attributi dei prodotti (categoria, colore, brand)
- **Collaborative filtering:** basato sulla similarità tra utenti con preferenze simili
- Gestisce bene il **cold-start problem** grazie alle feature dei prodotti

3.3 Visual Recommending con Trendyol DinoV2

Il modello **Trendyol DinoV2** suggerito è specificamente addestrato per e-commerce:

- **Embedding a 256 dimensioni:** rappresentazione compatta delle immagini prodotto
- **Similarity search:** data un'immagine, trova prodotti visivamente simili
- **Use case:** "Voglio qualcosa simile a questa foto che ho visto su Instagram"
- **Elemento distintivo:** aggiunge originalità alla tesi rispetto a un classico chatbot testuale

3.4 LLM per la Generazione

Per la componente conversazionale, propongo di usare **API commerciali** (come concordato):

- **OpenAI GPT-4 / GPT-4o-mini:** ottime performance, costi contenuti per prototipo
- **Anthropic Claude:** alternativa con buone capacità di ragionamento

4. Dataset

4.1 Intent Routing Dataset

Il dataset **ecommerce-intent-routing** proposto contiene ~2.750 esempi di query. Per il nostro sistema, mapperemo gli intent alle azioni del Router:

Intent Originale	Esempio	Azione Router
product	"Which headphones have the best noise cancellation?"	→ Search Agent o Recommender
sales	"Do you have the iPhone 15 in blue?"	→ Search Agent
support / account / bugs	"How do I track my order?"	→ Risposta fuori scope

4.2 Dataset Prodotti e Recommending

Per il catalogo e il recommending, propongo:

- **Amazon Product Data - Fashion:** recensioni e metadati di prodotti fashion
- **H&M Personalized Fashion Recommendations:** dataset Kaggle con transazioni e attributi
- **Immagini prodotto:** per generare embeddings con Trendyol DinoV2

5. Piano di Sviluppo Proposto

Fase	Attività
1	Setup e Router Agent: ambiente, dataset intent, classificatore, gestione contesto sessione
2	Search Agent + Catalogo: preparazione dataset prodotti, indicizzazione, ricerca con filtri
3	Recommender Agent: LightFM (content-based + collaborative), Trendyol DinoV2 (visual)
4	Integrazione: orchestrazione LangGraph, testing end-to-end, fix
5	Valutazione e Scrittura: benchmark, metriche, stesura tesi

6. Elementi Distintivi della Tesi

1. **Architettura Multi-Agente:** sistema modulare con 3 agenti specializzati orchestrati da LangGraph, non un semplice chatbot monolitico
2. **Triple approccio al Recommending:** content-based, collaborative filtering e visual similarity integrati nello stesso sistema
3. **Visual Recommending:** integrazione di Trendyol DinoV2 per similarity search basata su immagini