

# Design, ingegnerizzazione e realizzazione di un sistema di dialogo basato su LLM nel dominio delle tecnologie assistive

Tesi di Laurea Magistrale



---

Relatore: Prof. Alessandro Mazzei

Co-Relatori: Dott. Pier Felice Balestrucci, Dott. Michael Oliverio

Candidato: Dott. Stefano Vittorio Porta

2 Aprile 2025

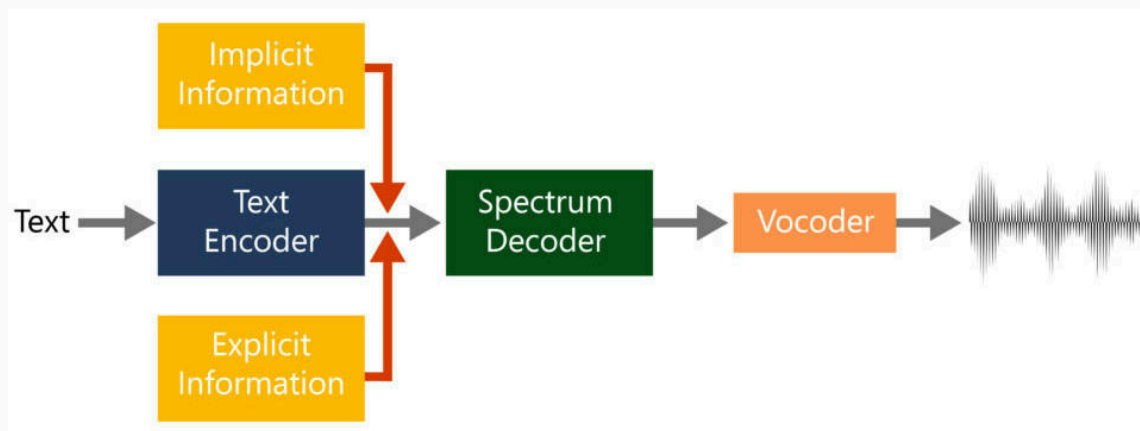
Università degli Studi di Torino, Dipartimento di Informatica - Anno Accademico 2023/2024

# 1. Contesto

---

Iniziamo ad ambientarci:

- **Lettura di contenuti testuali**: da 30 anni sono disponibili sistemi di sintesi vocale integrati in smartphone e computer.
- Essenziali per le persone con **disabilità visive**: consentono di accedere a contenuti testuali in modo autonomo e senza l'ausilio di un lettore umano.



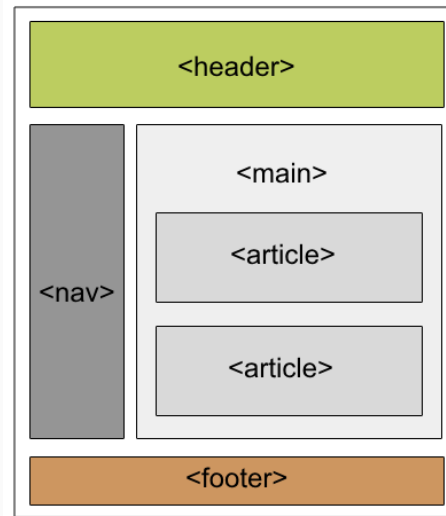
- Pagine web. I sistemi TTS sono in grado di
  - Leggere il testo
  - Interpretare la struttura del documento
  - Questo permette di seguire il flusso di lettura per fornire un'esperienza di qualità

## 1.2 Problema!



Tutto funziona, a patto che...

- La pagina web sia strutturata in modo semantico



- Le immagini siano accompagnate da un testo alternativo

`<img alt="...">` o `<img aria-label="...">`

**Queste due condizioni dipendono da chi prepara il contenuto!**

## 1.2 Problema!



I sistemi di TTS non sono in grado di interpretare il significato di un'immagine o di un elemento puramente visivo.



Se non viene fornita un'alternativa testuale contenente delle informazioni utili, l'utente non potrà comprendere appieno il contenuto della pagina o di uno specifico elemento!

Una di quattro immagini sul web non ha una descrizione testuale o non è informativa <sup>[1]</sup>.

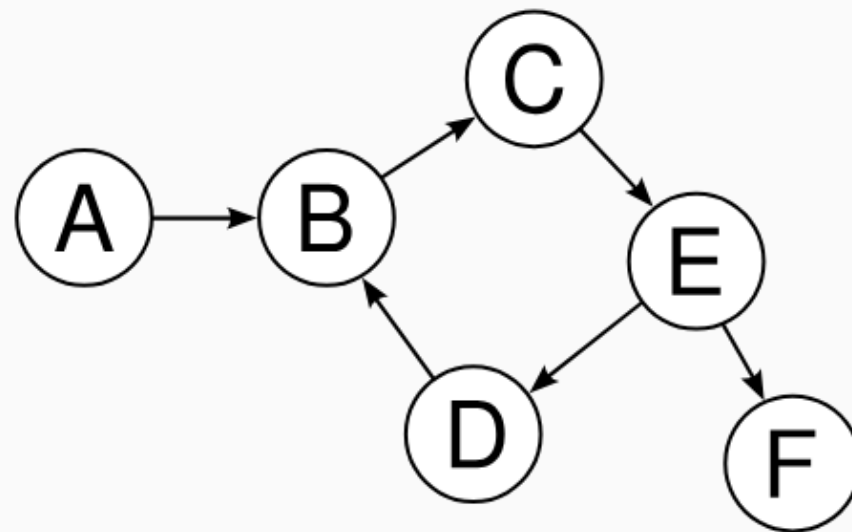
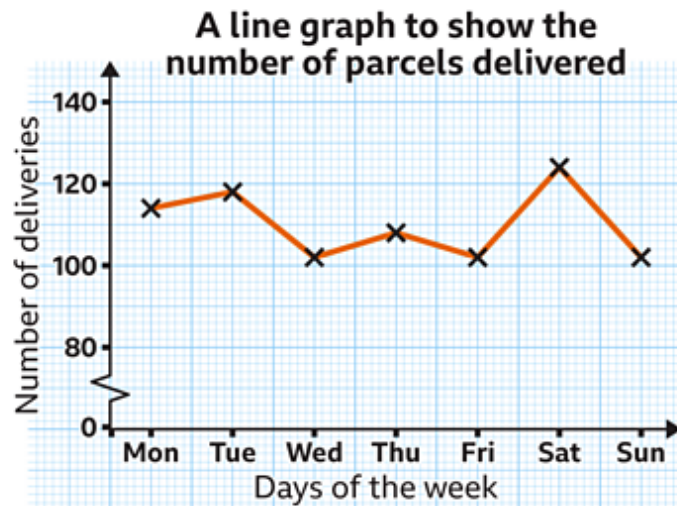
---

<sup>[1]</sup>WebAIM, «The WebAIM Million: The 2024 Report on the Accessibility of the Top 1,000,000 Home Pages». [Online]. Disponibile su: <https://webaim.org/projects/million/>

## 1.3 Anche peggio...



Se per le immagini possiamo utilizzare tecniche di **Computer Vision** o **Reti Neurali** per generare automaticamente un testo alternativo, per le rappresentazioni grafiche di dati (grafici, diagrammi, mappe) non è così semplice.



- Il Progetto NoVAGraphS si propone di rendere più accessibili questi contenuti, mediante la costruzione di sistemi di dialogo (Chatbot).
- Con essi è possibile interagire per ottenere informazioni sui dati presenti in grafi o strutture simili, per avere una comprensione **profonda** del contenuto.
- Il Progetto originale fa fondamento su AIML <sup>[1]</sup> (Artificial Intelligence Markup Language), un linguaggio di markup per la creazione di chatbot.

---

<sup>[1]</sup>«Artificial Intelligence Markup Language». [Online]. Disponibile su: <http://www.aiml.foundation/doc.html>



```
<category>
  <pattern>MI CHIAMO *</pattern>
  <template>
    Ciao <star/>, piacere di conoscerti!
  </template>
</category>
```

Codice 1: Esempio di AIML per la gestione di un saluto

- Le strategie di wildcard e pattern matching restano **prevalentemente letterali**: Se una frase si discosta dal pattern previsto, il sistema fallisce il matching
- La **gestione del contesto** (via <that>, <topic>, <star>, ecc.) è rudimentale
- L'integrazione (via <sraix>) con **basi di conoscenza esterne** (KB, database, API) è possibile implementando funzioni personalizzate, ma è di difficile gestione
- Le risposte generate sono **statiche e predefinite**, e non possono essere generate dinamicamente in base a dati esterni o a contesti più ampi in modo automatico

- Sviluppare un sistema di dialogo che superi le limitazioni di AIML evidenziate
- Integrare tecniche di **Natural Language Understanding** (NLU) e **Retrieval-Augmented Generation** (RAG) per migliorare l'esperienza d'uso
- Assicurare una elevata facilità di estensione e personalizzazione per diversi domini e applicazioni

## **2. Natural Language Understanding**

---

Il primo elemento dello stack di NLP rispetto ad AIML che vogliamo migliorare è il riconoscimento delle intenzioni dell'utente.

- Non useremo più un sistema basato su pattern matching ed espressioni regolari
- Riconosceremo la categoria di interazione affidandoci ad un classificatore basato su LLM
- Le parti variabili della frase (slot) verranno estratte tramite un sistema di Named Entity Recognition (NER)

- Essendo un task supervisionato, bisogna partire con l'etichettatura dei dati.
- Il dataset utilizzato proviene dalle precedenti pubblicazioni del progetto NoVAGraphS, e contiene 350 interazioni degli utenti prodotte durante precedenti sperimentazioni.
- L'annotazione dei dati:
  - Inizialmente è stata effettuata automaticamente
  - Successivamente è stata completamente riveduta ed effettuata manualmente
- Sono usati due livelli di granularità per la classificazione:
  - 7 **classi principali**
  - Sono state introdotte le **classi secondarie** per ogni classe principale, per un totale di 33.

| Classe     | Scopo   | Numero di Esempi |
|------------|---|------------------|
| transition | Domande che riguardano le transizioni tra gli stati                 | 77               |
| automaton  | Domande che riguardano l'automa in generale                         | 48               |
| state      | Domande che riguardano gli stati dell'automa                        | 48               |
| grammar    | Domande che riguardano la grammatica riconosciuta dall'automa       | 33               |
| theory     | Domande di teoria generale sugli automi                             | 15               |
| start      | Domande che avviano l'interazione con il sistema                    | 6                |
| off_topic  | Domande non pertinenti al dominio che il sistema deve saper gestire | 2                |

Tabella 1: Classi principali per la classificazione delle domande

| Sottoclassi       | Scopo  | Numero di Esempi |
|-------------------|--|------------------|
| description       | Descrizioni generali sull'automa                               | 14               |
| description_brief | Descrizione generale (breve) sull'automa                       | 10               |
| directionality    | Domande riguardanti la direzionalità o meno dell'intero automa | 1                |
| list              | Informazioni generali su nodi e archi                          | 1                |
| pattern           | Presenza di pattern particolari nell'automa                    | 9                |
| representation    | Rappresentazione spaziale dell'automa                          | 13               |

Tabella 2: Classi secondarie per la classe primaria dell'**Automa**



- Del dataset originale sono rimasti solo 229 esempi, divisi in classi sbilanciate.
- Per assicurare che il modello abbia buone prestazioni, è stato necessario aumentare il numero di esempi.
- Sono state generate 851 nuove domande, utilizzando LLM alle quali è stato fornito un insieme di quesiti di una certa classe.
- Per le domande off-topc è stato adoperato il dataset SQUAD <sup>[1]</sup>

---

<sup>[1]</sup>P. Rajpurkar, J. Zhang, K. Lopyrev, e P. Liang, «SQuAD: 100,000+ Questions for Machine Comprehension of Text», in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, e X. Carreras, A c. di, Association for Computational Linguistics, nov. 2016, pp. 2383–2392. doi: 10.18653/v1/D16-1264.

## 2.3 Data Augmentation

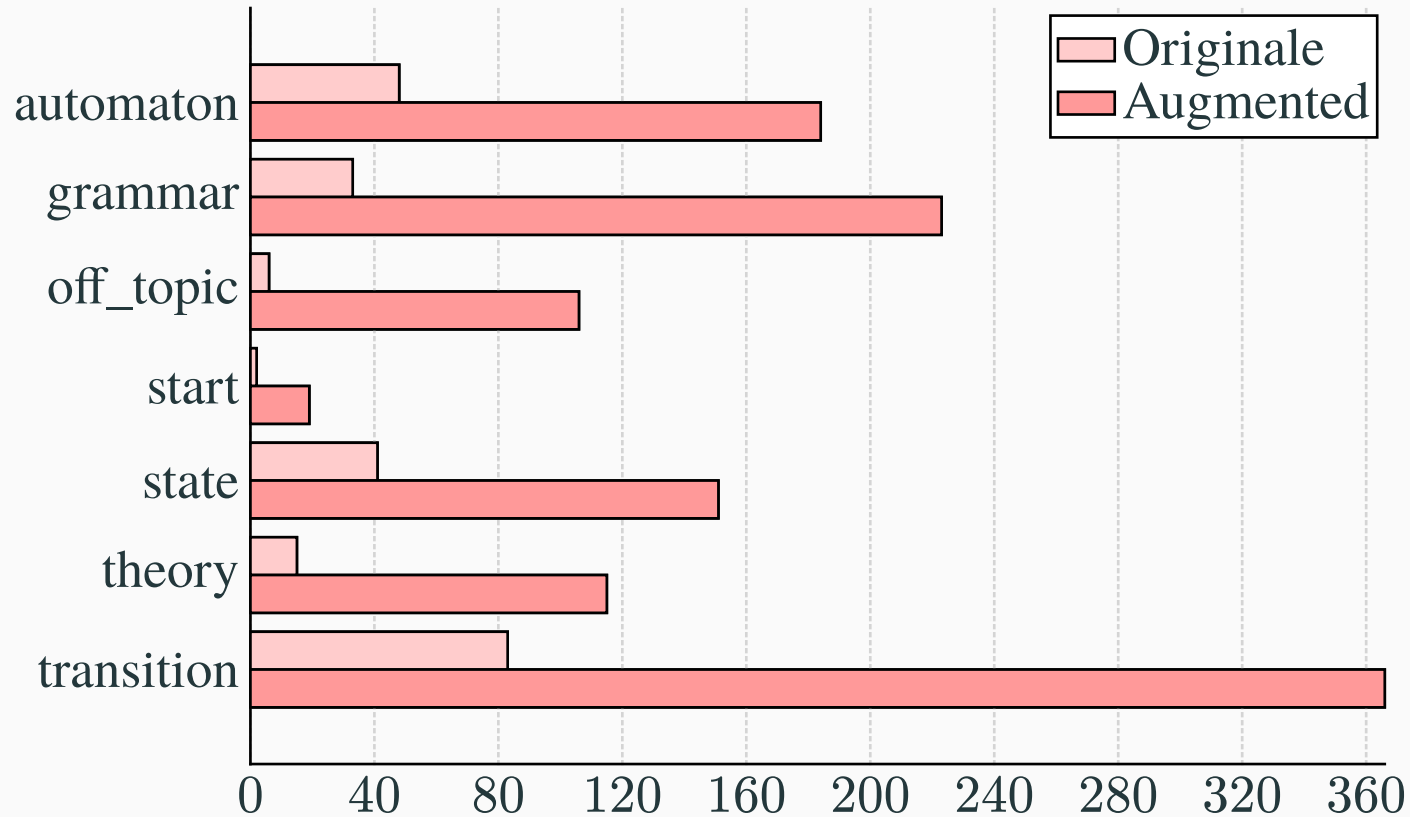


Figura 1: Numero di esempi per classe originale e aumentata

- Partendo LLM pre-addestrate e con una buona padronanza della lingua inglese, l'addestramento è piuttosto rapido e non richiede molte risorse.
- È stato eseguito un fine-tuning per adattare il modello alla classificazione delle domande.
- In questo modo il modello apprende le particolarità e sfumature dello scenario applicativo specifico.
- La metrica massimizzata è stata la **F1 score** (media armonica fra Precision e Recall)

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

I modelli utilizzati per il fine-tuning sono stati:

- google-bert/bert-base-uncased, versione da 110 milioni di parametri <sup>[1]</sup> (Google)
- distilbert/distilbert-base-uncased <sup>[2]</sup>, versione distillata di BERT, con circa il 40% in meno di parametri <sup>[3]</sup>
- google/mobilebert-uncased <sup>[4]</sup> con 25 milioni di parametri; per dispositivi mobili
- google/electra-small-discriminator <sup>[5]</sup>, da 14 milioni di parametri

---

<sup>[1]</sup>*Bert uncased model by Google*. [Online]. Disponibile su: <https://huggingface.co/google-bert/bert-base-uncased>

<sup>[2]</sup>*Distilbert base model*. [Online]. Disponibile su: <https://huggingface.co/distilbert/distilbert-base-uncased>

<sup>[3]</sup>V. Sanh, L. Debut, J. Chaumond, e T. Wolf, «DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter», 2020. doi: 10.48550/arXiv.1910.01108.

<sup>[4]</sup>*MobileBERT on Huggingface*. [Online]. Disponibile su: <https://huggingface.co/google/mobilebert-uncased>

<sup>[5]</sup>*Electra on Huggingface*. [Online]. Disponibile su: <https://huggingface.co/google/electra-small-discriminator>

## 2.4 Fine-tuning

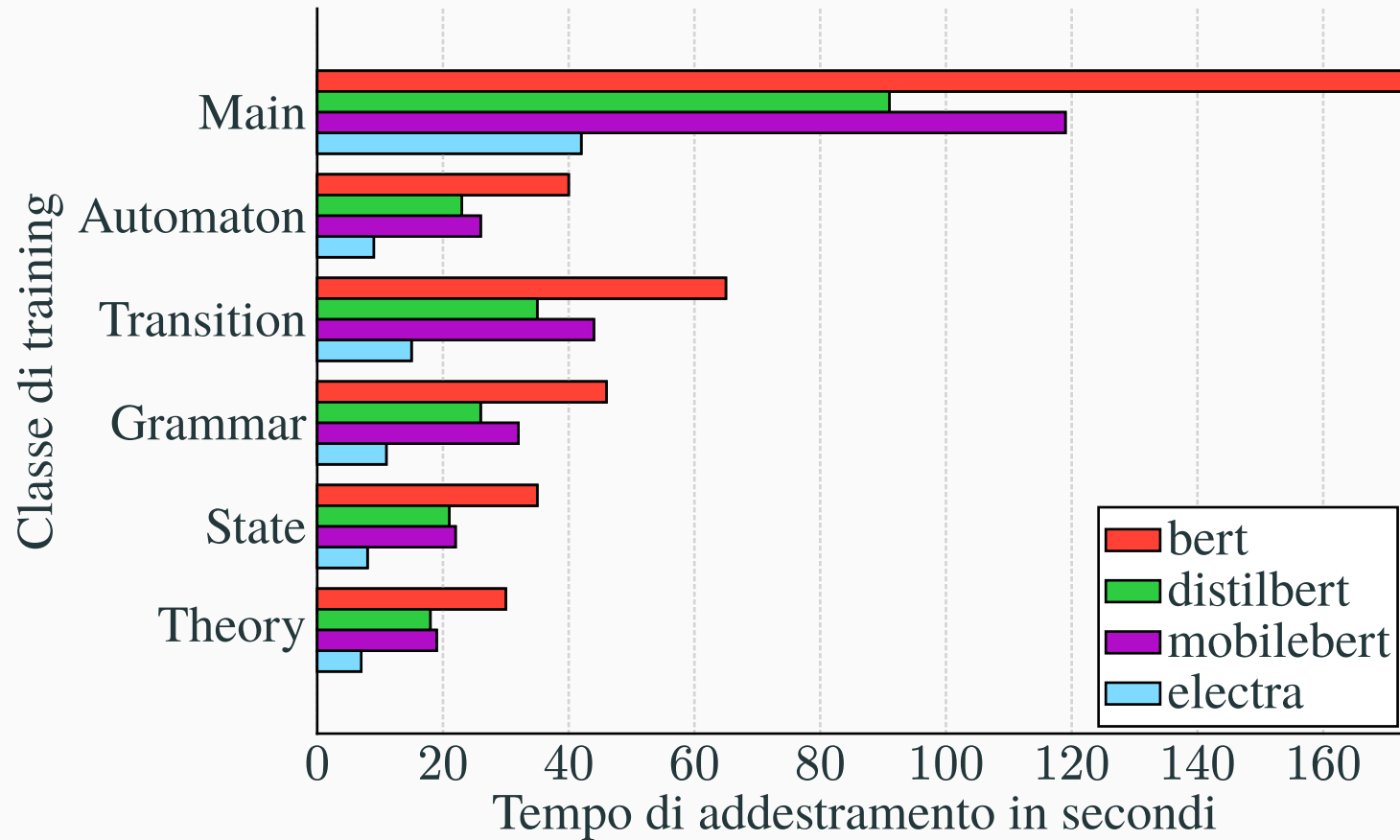


Figura 2: Tempo di addestramento per BERT, DistilBERT, MobileBERT ed ELECTRA

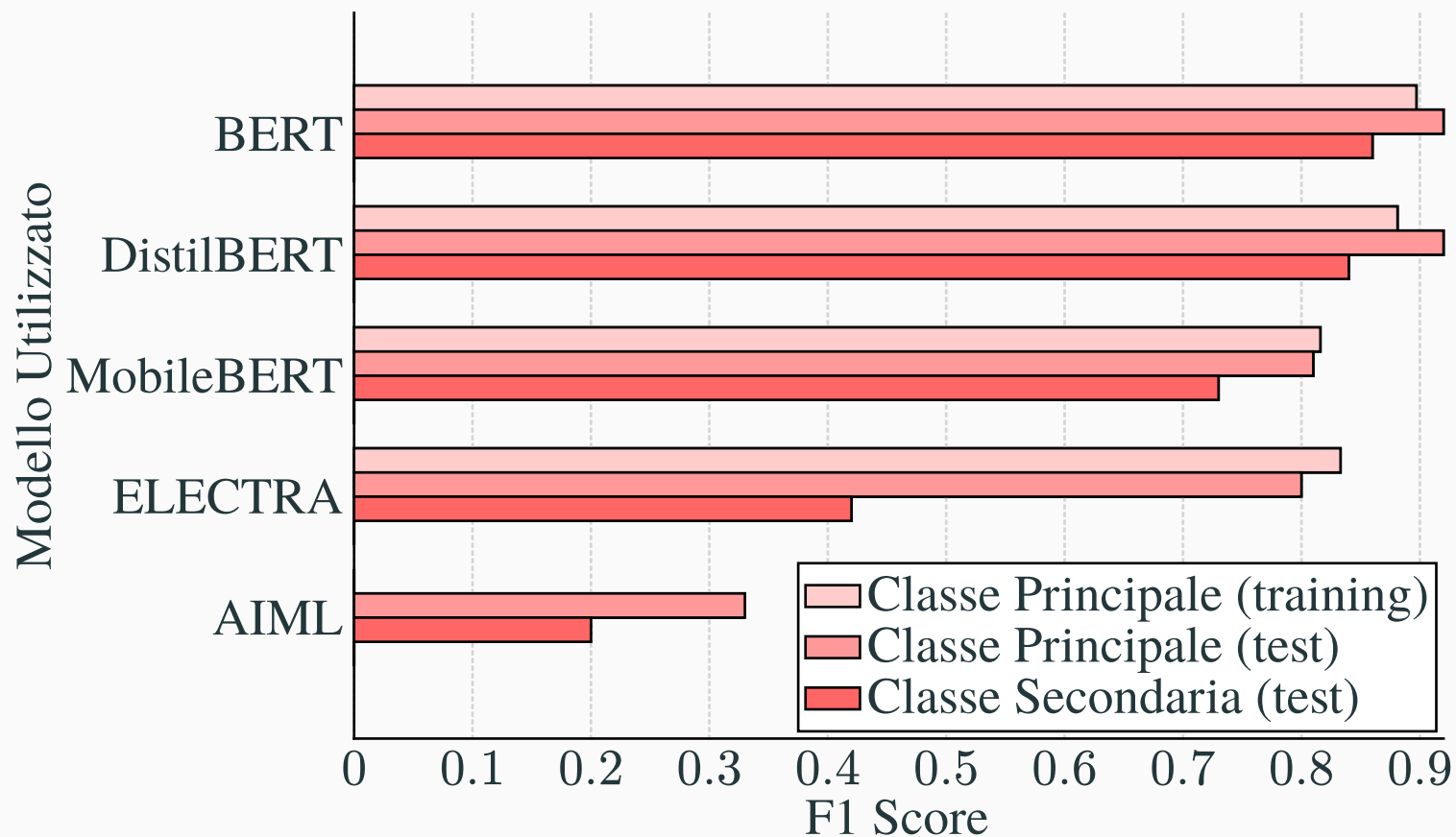


Figura 3: Performance su test set bilanciato da 468 ulteriori domande confrontato con AIML

## 2.4 Fine-tuning



|              | Performance AIML |        |          | Performance BERT |        |             | Esempi |
|--------------|------------------|--------|----------|------------------|--------|-------------|--------|
|              | Precision        | Recall | F1-score | Precision        | Recall | F1-score    |        |
| Automaton    | 0.52             | 0.19   | 0.27     | 0.93             | 0.93   | <b>0.93</b> | 75     |
| Grammar      | 0.90             | 0.13   | 0.23     | 0.78             | 0.83   | <b>0.81</b> | 70     |
| Off-Topic    | 0.26             | 0.82   | 0.39     | 1.00             | 0.96   | <b>0.98</b> | 100    |
| Start        | 1.00             | 0.53   | 0.69     | 1.00             | 0.90   | <b>0.95</b> | 40     |
| State        | 0.17             | 0.19   | 0.18     | 0.96             | 1.00   | <b>0.98</b> | 43     |
| Theory       | 0.00             | 0.00   | 0.00     | 0.57             | 0.57   | <b>0.57</b> | 30     |
| Transition   | 0.67             | 0.28   | 0.40     | 0.97             | 0.99   | <b>0.98</b> | 110    |
| Accuracy     |                  |        | 0.35     |                  |        | <b>0.92</b> | 468    |
| Macro avg    | 0.44             | 0.27   | 0.27     | 0.89             | 0.88   | <b>0.88</b> | 468    |
| Weighted avg | 0.53             | 0.35   | 0.33     | 0.92             | 0.92   | <b>0.92</b> | 468    |

## 2.4 Fine-tuning

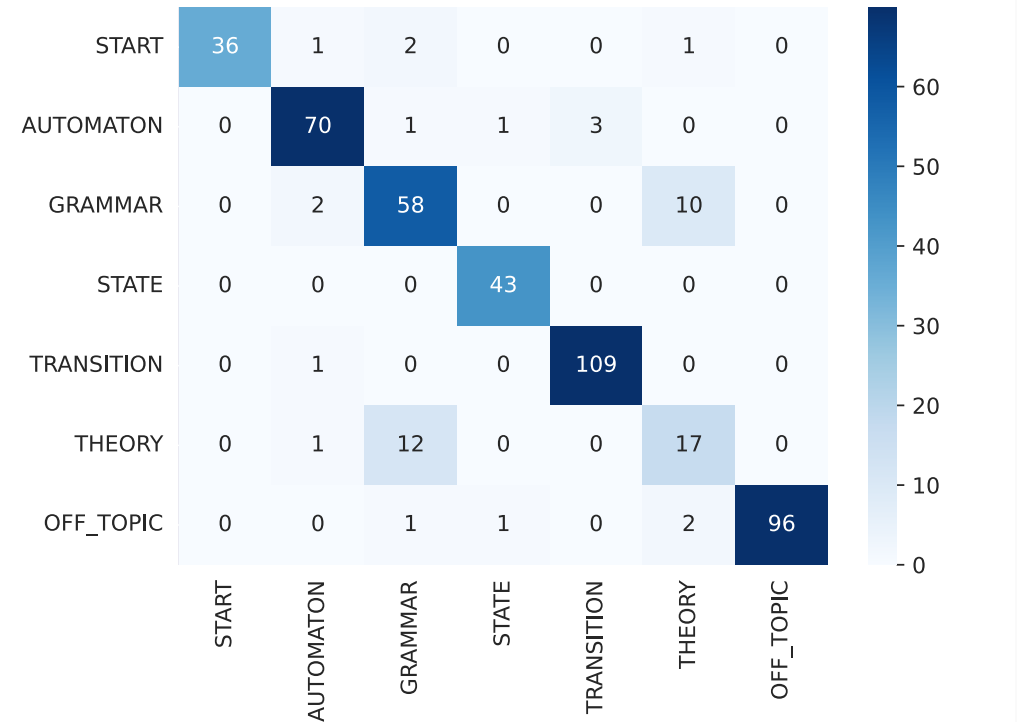
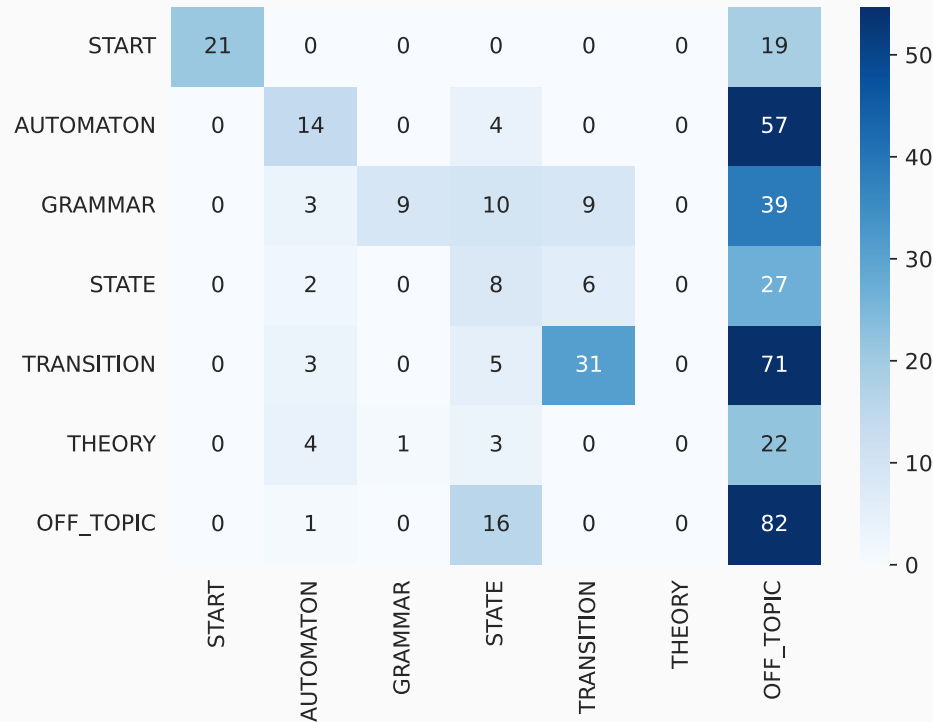


Figura 4: Matrici di confusione per le classi principali con AIML e BERT



## 2.5 Named entity recognition



Dobbiamo poter estrarre le parti variabili delle domande, in modo da capire quali informazioni l'utente sta cercando. È stato usato *Doccano* per l'annotazione delle frasi.

The screenshot shows the Doccano web interface for Named Entity Recognition. The main text area contains the sentence: "The Hitchhiker's Guide to the Galaxy (sometimes referred to as HG2G, HHGTTGor H2G2) is a comedy science fiction series created by Douglas Adams. Originally a radio comedy broadcast on BBC Radio 4 in 1978, it was later adapted to other formats, including stage shows, novels, comic books, a 1981 TV series, a 1984 video game, and a 2005 feature film." Each entity is highlighted with a colored box and labeled below it: "The Hitchhiker's Guide to the Galaxy" (Title), "HG2G" (Title), "HHGTTGor H2G2" (Title), "comedy science fiction" (Genre), "Douglas Adams" (Person), "BBC Radio 4" (Others), "1978" (Date), "stage shows" (Genre), "novels" (Genre), "comic books" (Genre), "1981" (Date), "1984" (Date), and "2005" (Date). The interface includes a top navigation bar with a menu icon, a sun icon, and the word "PROJECTS". Below the text area are three icons: a close button (X), a filter icon, and a document icon. On the right side, there is a table with the headers "Key" and "Value", and a message "No data available".

| Key               | Value |
|-------------------|-------|
| No data available |       |

Figura 5: Interfaccia di Doccano per l'annotazione dei dati di NER.

In seguito all'etichettatura sono risultate tre classi di entità:

- input: per input o sequenze di simboli.  
«Does it only accept 1s and 0s?»  $\Rightarrow$  [20, 21, "input"], [27, 28, "input"]
- node: per i frammenti di testo che contengono nodi o stati dell'automa.  
«Is there a transition between q2 and q0?»  $\Rightarrow$  [30, 32, "node"], [37, 39, "node"]
- language: informazioni sulla grammatica accettata dall'automa.  
«Does the automaton accept strings over the alphabet {0,1}?»  $\Rightarrow$  [53, 58, "language"]

## 2.5 Named entity recognition

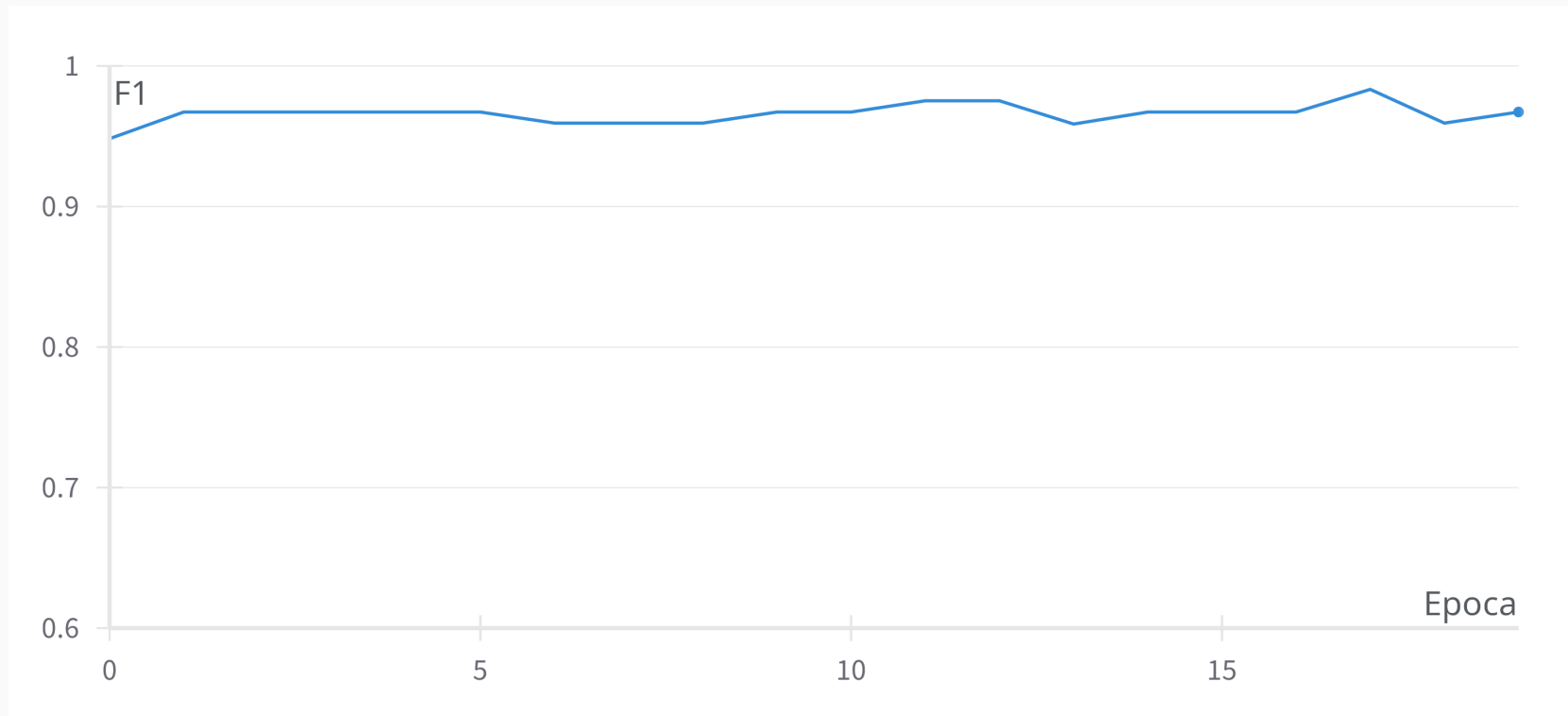


Figura 6: Performance di F1 del modello di NER durante l'addestramento tramite SPACY <sup>[1]</sup>.

<sup>[1]</sup>M. Honnibal e I. Montani, «spaCy: Industrial-strength Natural Language Processing in Python». [Online]. Disponibile su: <https://spacy.io/>

### **3. Retrieval Augmented Generation**

---

- Per fornire risposte complete e pertinenti, il sistema deve essere in grado di interrogare una base di conoscenza esterna.
- Esistono diverse tecniche, che dipendono dal tipo di dati e dalla loro rappresentazione:
  - Basi di conoscenza strutturate: SQL, SPARQL, ecc.
  - Corpus: ricerca full-text, TF-IDF, embeddings...
  - API e servizi esterni: REST, JMESPath

```
SELECT customer_name, order_date, total_amount
FROM orders
WHERE order_id = :orderId
      AND customer_id = :customerId;
```

Query 1: Esempio di query SQL per il recupero di dettagli di un ordine in un sistema e-commerce.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT ?director
WHERE {
    dbr:The_Incredibles dbo:director ?director .
}
```

Query 2: Query SPARQL per il recupero del regista del film *Gli Incredibili* (Brad Bird).

## 3.4 Cosine Similarity

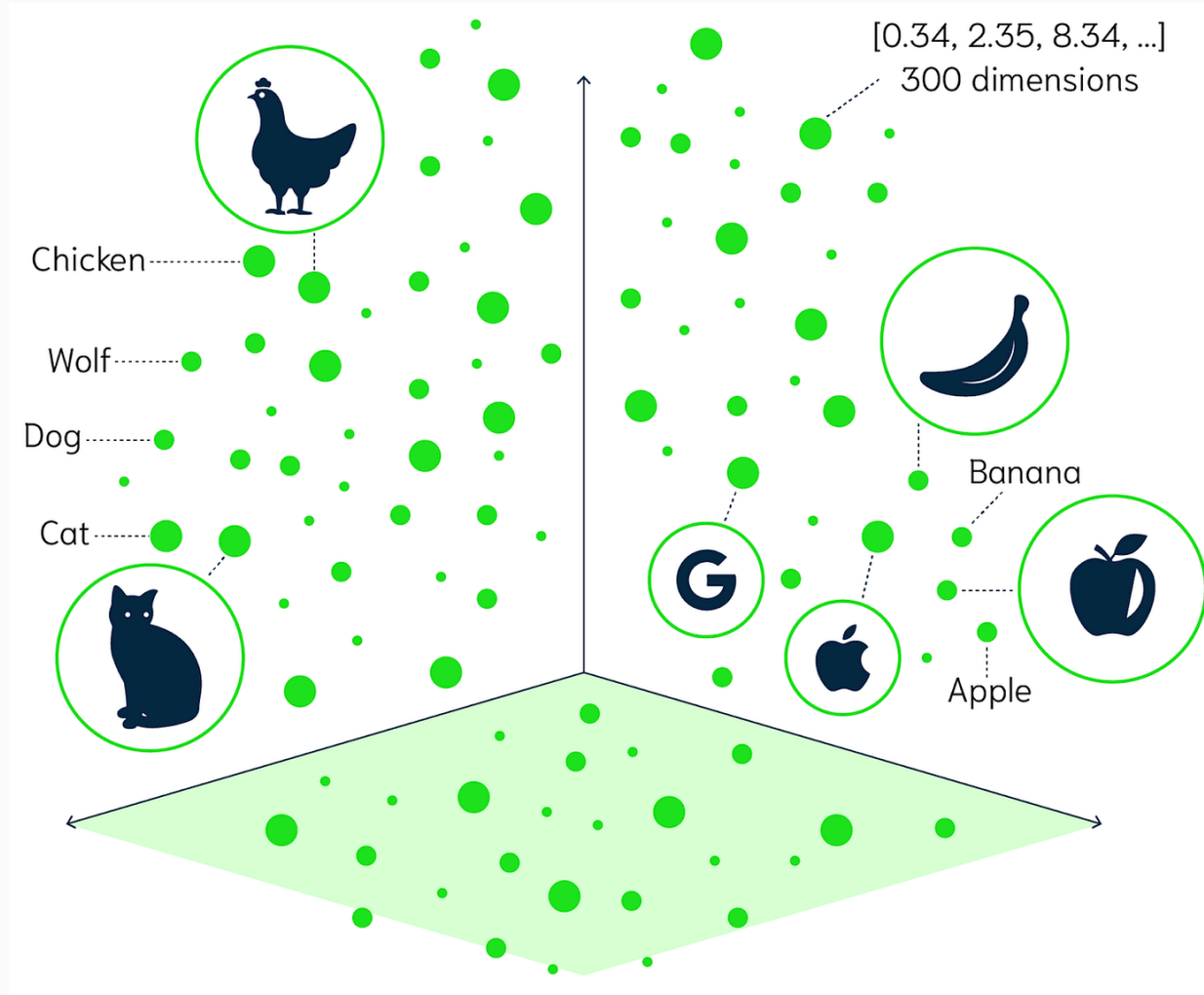


Figura 7: Illustrazione della cosine similarity.



### Prompt

Ecco i dati estratti dal database di e-commerce:

- Prezzo dell'articolo: 49,99 euro
- Tempi di spedizione: 2 giorni
- Colori disponibili: rosso, blu, verde

Genera una breve risposta da mostrare al cliente, evitando informazioni non pertinenti e senza inventare nulla.

### Risposta da GPT-4o

L'articolo è disponibile nei colori rosso, blu e verde al prezzo di 49,99 euro. I tempi di spedizione sono di 2 giorni.

### Prompt

Sei un assistente che fornisce informazioni su ordini online. Di seguito trovi una selezione parziale delle interazioni con l'utente:

**Utente:** “Qual è lo stato di avanzamento del mio ordine?”

Il sistema ha reperito i seguenti dati:

- Ordine n. 1357
- Stato: in spedizione
- Previsione di consegna: 10/04/2025

Ora, rispondi alla domanda dell'utente in modo chiaro e conciso, mantenendo la coerenza con le interazioni precedenti.

- Dobbiamo assicurarci che il sistema sia in grado di rispondere in modo non solo coerente e pertinente, ma anche efficace alle domande degli utenti.
- Per effettuare le valutazioni sono stati utilizzati diversi LLM
- Una volta generate tutte le risposte con i vari modelli, sono state tutte annotate
- I risultati delle annotazioni hanno fornito dettagli essenziali per la scelta del modello finale

Sono state scelte delle LLM open-weights dati i multipli vantaggi che offrono:

- Trasparenza e accountability
  - Accesso ai pesi e alla struttura dei modelli per identificare bias e anomalie.
  - Maggiore controllo sulla sicurezza e affidabilità dell'AI.
- Verifiche indipendenti e replicabilità per ricercatori e sviluppatori. Maggiore fiducia.
- Innovazione e competizione
  - Riduzione delle barriere d'ingresso per startup e centri di ricerca.
  - Non sono necessarie per forza grandi risorse hardware
  - Protezione della privacy e riduzione dei costi operativi.
- Supporto a uno sviluppo sostenibile e responsabile dell'AI.

## 3.6 Generazione delle risposte



You are a helpful assistant expert in finite state automata.

Answer the question given by the user using the retrieved data, using plain text only.

Avoid referring to the data directly; there is no need to provide any additional information.

Keep the answer concise and short, and avoid using any additional information not provided.

The system has retrieved the following data:

\ \ \

{data}

\ \ \

The user has asked the following question:

\ \ \

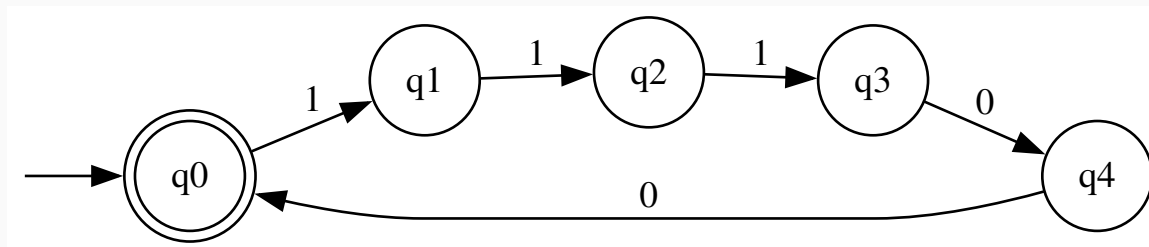
{question}

\ \ \

## 3.6 Generazione delle risposte



Per la generazione delle risposte è stato selezionato un sottinsieme di domande che richiedono informazioni riguardo questo specifico automa a stati finiti:



Se il tema della domanda è riguardante le *transizioni uscenti da un nodo*, prima di generare la risposta, il sistema recupera le i dettagli utili e li presenta al modello.

The transitions exiting from the node are the following:

- From q0 to q1, with label '1'

```
digraph FSA {  
    rankdir=LR;  
    node [shape = circle];  
    q0 [shape = doublecircle];  
    q1; q2; q3; q4;  
  
    start [shape=none, label=""];  
    start -> q0;  
  
    q0 -> q1 [label = "1"];  
    q1 -> q2 [label = "1"];  
    q2 -> q3 [label = "1"];  
    q3 -> q4 [label = "0"];  
    q4 -> q0 [label = "0"];  
}
```

Snippet 1: Rappresentazione in formato Graphviz dell'automa a stati finiti utilizzato come input per le domande.

- È necessario annotare le risposte per valutare le performance degli LLM candidati
- Basato sulle ricerche di Z. Kasner e O. Dušek <sup>[1]</sup>
- Verrà utilizzato il software Factgenie <sup>[2]</sup> da loro sviluppato per le annotazioni

---

<sup>[1]</sup>Z. Kasner e O. Dušek, «Beyond Traditional Benchmarks: Analyzing Behaviors of Open LLMs on Data-to-Text Generation», in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, e V. Srikumar, A c. di, Association for Computational Linguistics, 2024, pp. 12045–12072. [Online]. Disponibile su: <https://aclanthology.org/2024.acl-long.651>

<sup>[2]</sup>*Factgenie*. [Online]. Disponibile su: <https://github.com/ufal/factgenie>



Gli annotatori sono liberi di evidenziare nelle risposte frammenti problematici semplicemente selezionandoli. Sono stati definiti quattro generi di errori:

- **INCORRECT<sup>I</sup>**: la risposta contiene informazioni che contraddicono i dati forniti o che sono chiaramente sbagliate.
- **NOT\_CHECKABLE<sup>NC</sup>**: la risposta contiene informazioni che non possono essere verificate con i dati forniti.
- **MISLEADING<sup>M</sup>**: la risposta contiene informazioni fuorvianti o che possono essere interpretate in modo errato.
- **OTHER<sup>O</sup>**: la risposta contiene errori grammaticali, stilistici o di altro tipo.

12345678910111213141516171819202122232425

What are the incoming and outgoing transition paths for q1?

Instructions

Incorrect

Not checkable

Misleading

Other

Drag your mouse over the text to highlight the span:

Incoming transition for q1 comes from q0 with label "1". Outgoing transition goes to q2 with label "1".  
Incoming: q0 → q1 (label "1") Outgoing: q1 → q2 (label "1")

Please check if you agree with any of the following statements:

☒ The text 100% accurate and does not contain any errors.

☐ The text is missing or incomplete.

☐ The text is severely off-topic (seems completely unrelated to the data).

Answer Clearness (if it is understandable)

Completely clear

Answer length (if it was too short to be 100% clear or

Figura 9: Interfaccia di Factgenie

41 / 60

Oltre ad evidenziare parti problematiche delle risposte, è stato richiesto agli annotatori anche di fornire delle valutazioni qualitative su alcune metriche:

- **Accuratezza della risposta:** da selezionare quando la risposta è corretta al 100% e non contiene errori sui dati;
- **Assenza o incompletezza di informazioni:** da selezionare quando la risposta non contiene tutte le informazioni rilevanti;
- **Totale incongruenza della risposta:** da selezionare quando la risposta appare completamente scorrelata o non pertinente alla domanda;
- **Chiarezza della risposta:** se è comprensibile e ben strutturata;
- **Lunghezza della risposta:** se la comprensione della risposta è facilitata dalla sua lunghezza (o brevità);
- **Utilità percepita della risposta:** se la risposta è utile e fornisce informazioni rilevanti;
- **Apprezzamento generale:** se la risposta è apprezzata o gradita.

In totale, 12 annotatori hanno partecipato alla valutazione delle risposte generate dai modelli.

Riguardo gli annotatori:

- 8 su 12 sono studenti del Dipartimento di Informatica;
- 2 hanno un background ingegneristico;
- I rimanenti 2 provengono dal Dipartimento di Storia e Biologia.

L'età media è di 28 anni, con un range tra i 21 e i 68 anni.

Ogni volontario ha valutato un sottoinsieme delle risposte, garantendo comunque un overlap sulle 75 risposte totali.

Oltre agli annotatori umani, sono stati utilizzati anche due modelli di LLM (GPT-o3-mini e GPT-4.5) per svolgere automaticamente una valutazione simile a quella dei volontari ( $\epsilon_{\text{hum}}$ ).

| Modello     | Incorrect               |                  | Not Checkable           |                  | Misleading              |                  | Other                   |                  | Globale                 |                  |
|-------------|-------------------------|------------------|-------------------------|------------------|-------------------------|------------------|-------------------------|------------------|-------------------------|------------------|
|             | $\epsilon_{\text{hum}}$ | $\epsilon_{4.5}$ | $\epsilon_{\text{hum}}$ | $\epsilon_{4.5}$ | $\epsilon_{\text{hum}}$ | $\epsilon_{4.5}$ | $\epsilon_{\text{hum}}$ | $\epsilon_{4.5}$ | $\epsilon_{\text{hum}}$ | $\epsilon_{4.5}$ |
| Deepseek    | <b>20.0%</b>            | <b>13.3%</b>     | 6.67%                   | <b>0.0%</b>      | <b>26.7%</b>            | 6.7%             | 66.7%                   | <b>0.0%</b>      | 83.0%                   | <b>33.3%</b>     |
| Gemma2      | 26.7%                   | 26.7%            | <b>0.0%</b>             | <b>0.0%</b>      | 33.3%                   | <b>0.0%</b>      | <b>20.0%</b>            | 6.7%             | <b>30.6%</b>            | <b>33.3%</b>     |
| Llama3.1    | 33.3%                   | 33.3%            | 6.67%                   | <b>0.0%</b>      | <b>26.7%</b>            | 6.7%             | 46.7%                   | <b>0.0%</b>      | 67.3%                   | 53.3%            |
| GPT-4o      | 20.0%                   | <b>6.67%</b>     | 13.3%                   | <b>0.0%</b>      | 20.0%                   | <b>0.0%</b>      | 40.0%                   | <b>0.0%</b>      | 36.0%                   | <b>6.67%</b>     |
| GPT-o3-mini | <b>0.0%</b>             | 13.3%            | <b>0.0%</b>             | <b>0.0%</b>      | <b>13.3%</b>            | <b>0.0%</b>      | <b>20.0%</b>            | <b>0.0%</b>      | <b>18.6%</b>            | 20.0%            |

Tabella 3: Percentuali di *risposte contenenti almeno un errore*, secondo le annotazioni umane ( $\epsilon_{\text{hum}}$ ) e le valutazioni automatiche ( $\epsilon_{4.5}$ ). Più basso è il valore, migliore è la qualità delle risposte.

| Metrica       | LLM Commerciale |     | LLM Open-Weights |     |
|---------------|-----------------|-----|------------------|-----|
| Chiarezza     | GPT-o3-mini     | 93% | Deepseek-r1:8b   | 69% |
| Lunghezza     | GPT-o3-mini     | 96% | Deepseek-r1:8b   | 78% |
| Utilità       | GPT-o3-mini     | 98% | Deepseek-r1:8b   | 82% |
| Apprezzamento | GPT-o3-mini     | 95% | Deepseek-r1:8b   | 68% |

Tabella 4: Percentuali di valutazione delle risposte generate dai modelli LLM commerciali e open-weights.

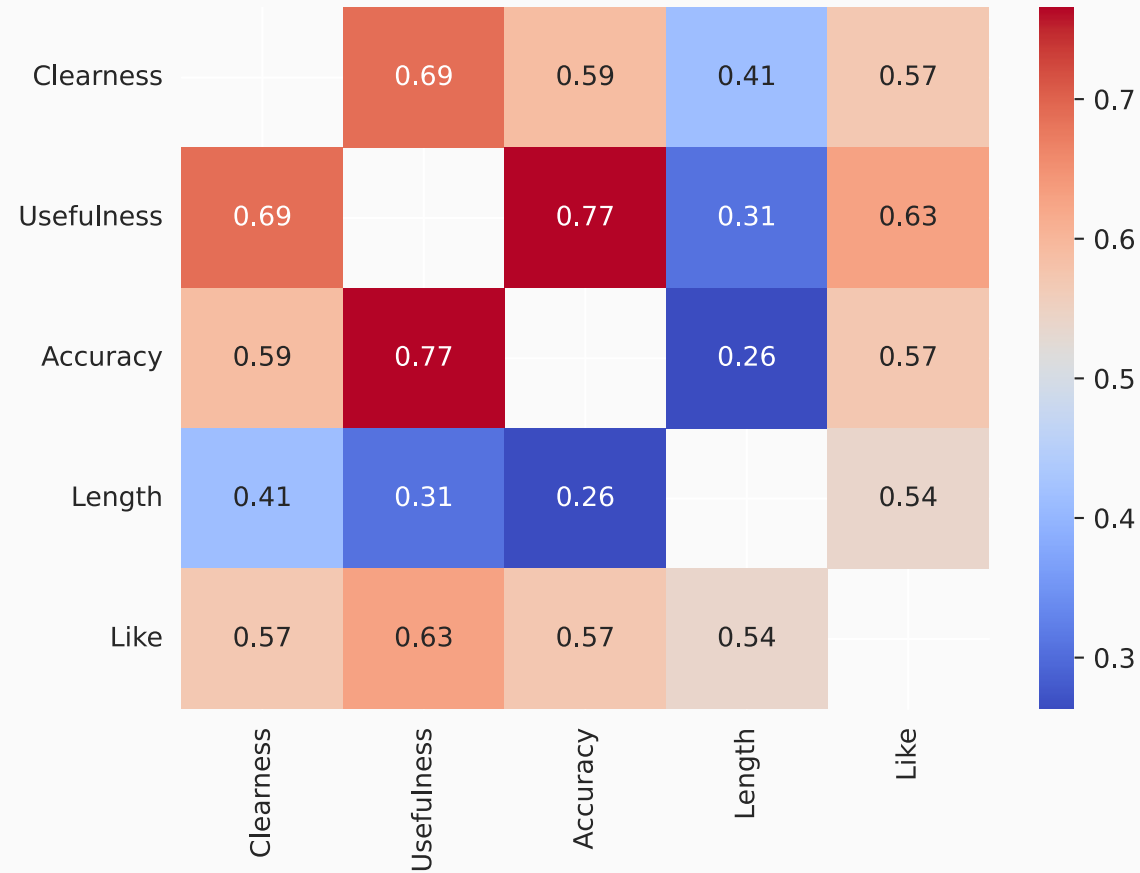


Diagramma 1: Correlazione tra le metriche di valutazione delle risposte per ogni modello.

## **4. Ingegnerizzazione del sistema**

---



Il nuovo sistema deve offrire funzionalità equivalenti a quelle di AIML, ma con un'architettura più moderna e flessibile:

1. Riconoscimento del topic:
  - Classificazione neurale
  - NER
2. Recupero delle informazioni:
  - Script
  - Database
  - API
3. Generazione delle risposte:
  - LLM locale/remota
  - Template

- Dal momento che dobbiamo lavorare con dei LLM, è necessario un sistema che effettui fine-tuning per ogni insieme di interazioni. Potremmo lasciare il fine-tuning a runtime, ma prepararlo in anticipo permette di risparmiare tempo e risorse.
- È sufficiente descrivere in modo sequenziale (pipeline) le operazioni da eseguire, e il «compilatore» si occuperà di preparare tutto il necessario

## 4.2 Compilatore



```
- name: "question_intent_classifiers"
  type: classification
  steps:
    - type: load_csv
      name: data
      path: "./multitask_training/data_cleaned_manual_combined.csv"
      label_columns: ["Global Subject", "Question Intent"]
    - type: split_data
      name: split
      dataframe: data.dataframe
      on_column: "Global Subject"
      for_each:
        - type: train_model
          name: training
          dataframe: split.dataframe
          pretrained_model: "google/electra-small-discriminator"
          examples_column: "Question"
          labels_column: "Question Intent"
          resulting_model_name: "question_intent_{on_column}"
```

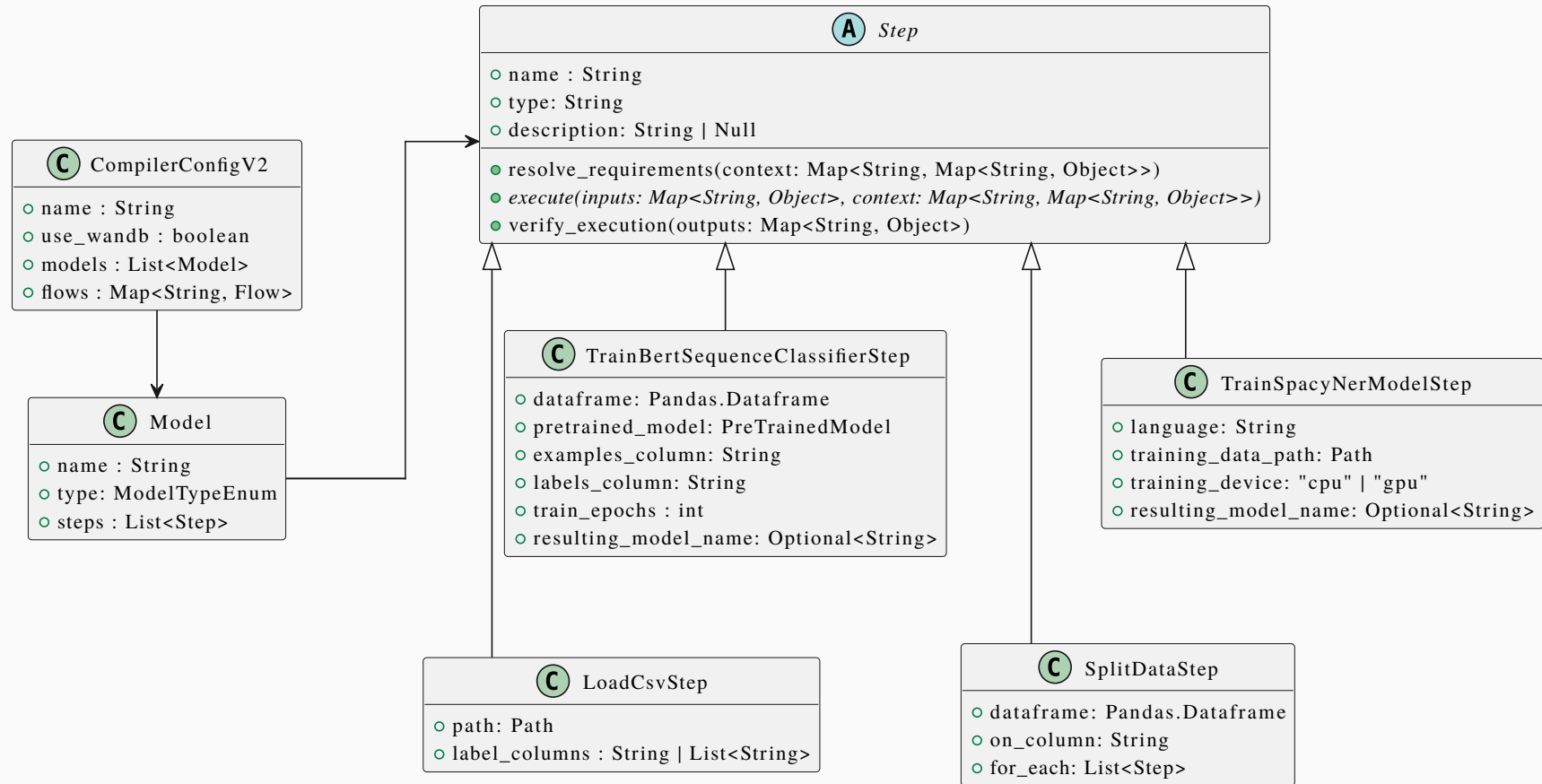
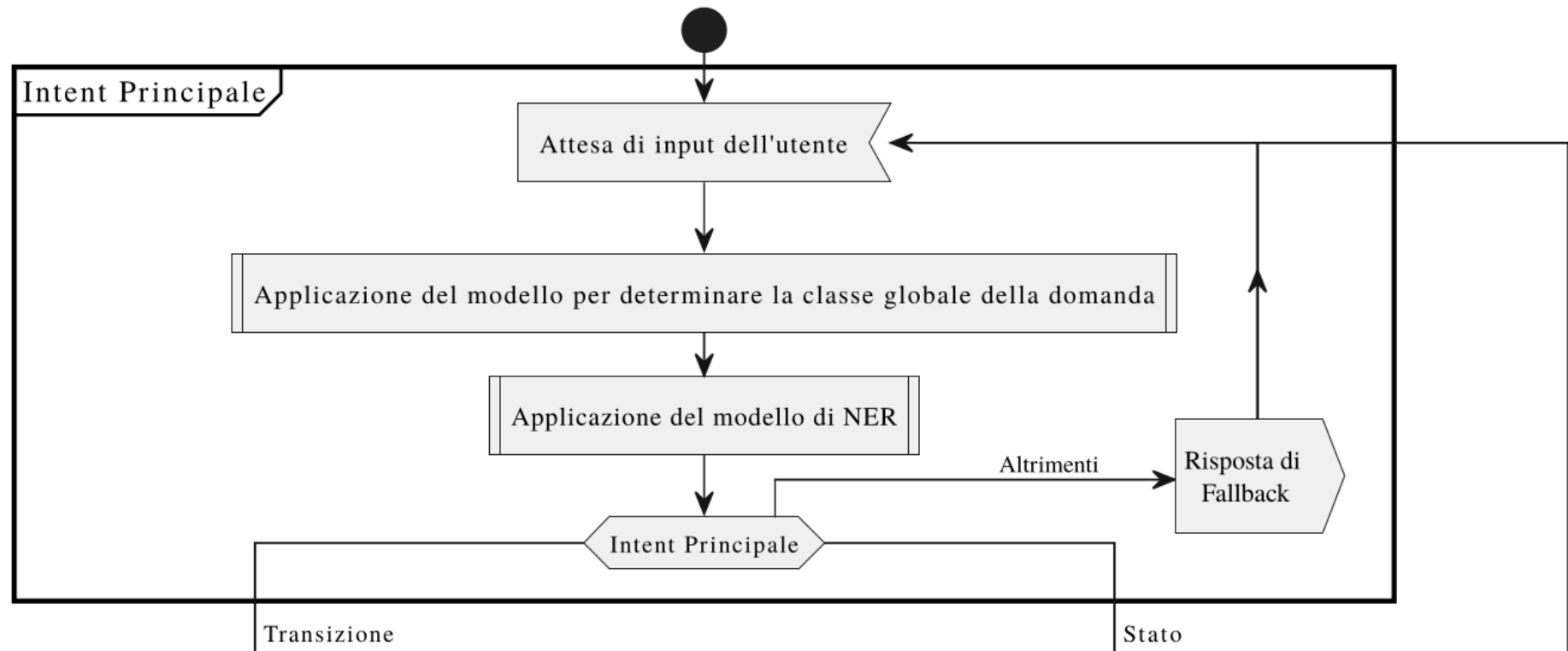


Diagramma 2: Class Diagram raffigurante le classi e proprietà utilizzate per la compilazione.

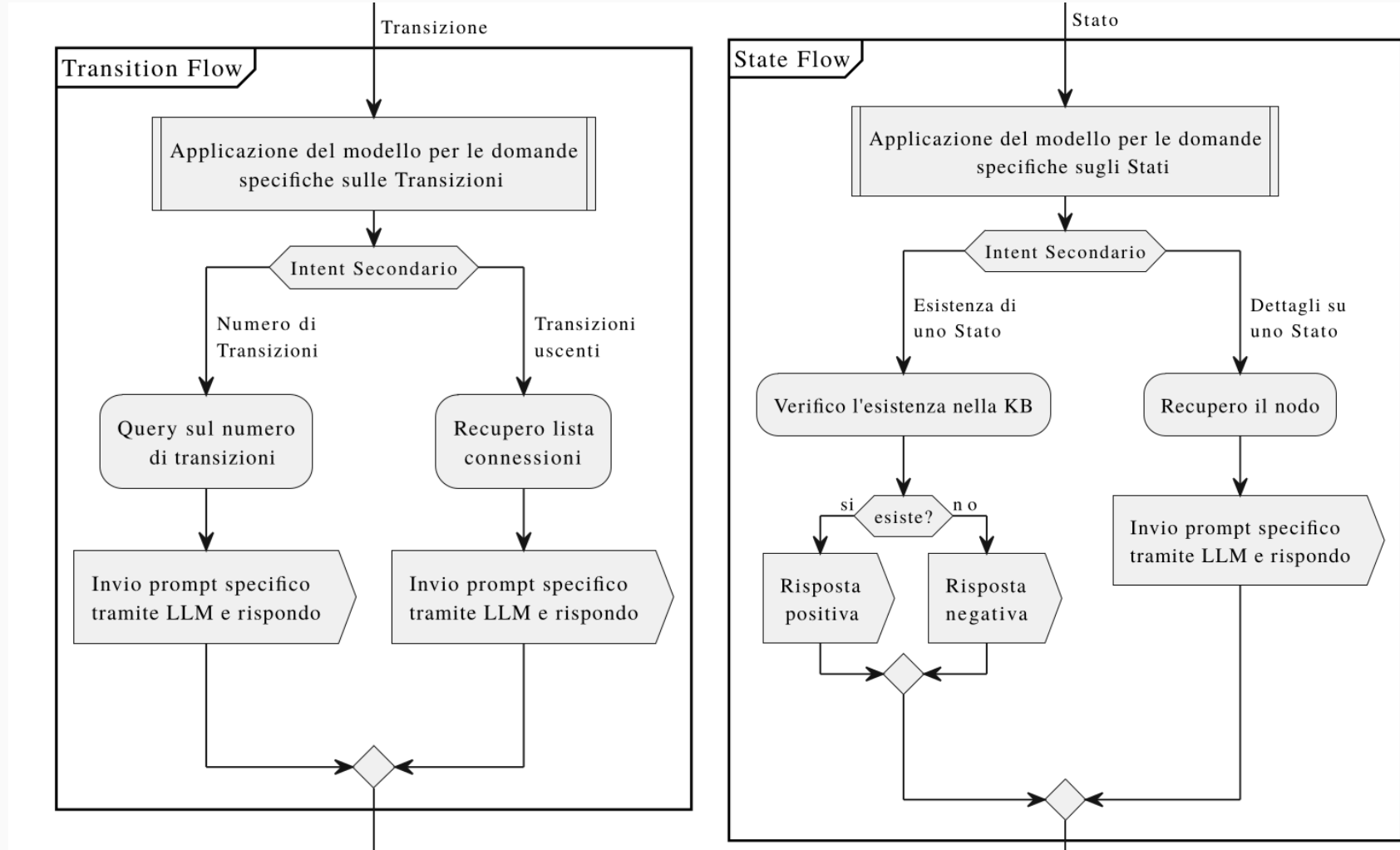
Come per AIML, è necessario un motore di esecuzione del chatbot, che si occupi di seguire il flusso di esecuzione delle interazioni e di gestire le domande degli utenti.

- Il motore di esecuzione è un automa a stati finiti
- Il percorso di interazione è un grafo: ogni nodo codifica un'azione che il motore deve svolgere:
  - Lasciare la parola all'utente
  - Recuperare informazioni da risorse (DB, API, ecc.)
  - Generare una risposta (LLM, template, default, ecc.)
- Un insieme di nodi è definito come un *flusso* di interazione (flow), che può essere richiamato in qualsiasi momento.

## 4.4 Esempio di chatbot



## 4.4 Esempio di chatbot



## **5. Conclusioni**

---



- **Riconoscimento Intenzioni:** AIML si basa solo su pattern matching, il sistema proposto supporta sia classificatori neurali che basati su regole.
- **Gestione Contesto:** AIML usa <topic> in modo rigido, il nuovo sistema adotta un meccanismo di Flow più articolato, con un contesto di esecuzione completo.
- **Dati Esterni:** AIML ricorre a <sraix> in modo periferico, il nuovo sistema integra nativamente moduli di retrieval come componente centrale.
- **Controllo del Flusso:** AIML utilizza <condition> e <srai> in modo semplice, il nuovo sistema offre branching condizionale e passaggio dinamico tra flussi.

- **Approccio ibrido:** Combina dichiarativo (AIML) con la flessibilità di Python e reti neurali.
- **Scalabilità e adattabilità:** Ideale per chatbot complessi, riduce sforzo di progettazione e manutenzione.
- **Controllo totale:** Definire con precisione l'ordine delle interazioni evita limiti tipici dei LLM, come explainability ridotta, allucinazioni e jailbreaking.

L'utilizzo degli step permette di avere la massima flessibilità, astraendo comunque il sistema da dettagli implementativi.

- Libreria di valutazione delle espressioni (Asteval) che consente di eseguire codice Python a runtime, garantendo grande flessibilità
- Assenza di protezioni predefinite contro codice malevolo rende necessaria una sandbox
- Servono controlli più stringenti per garantire sicurezza e stabilità

- YAML spinto al limite: ottimo per flow semplici, ma diventa complesso per espressioni avanzate
- Possibilità di introdurre un DSL dedicato per maggiore chiarezza e validazione

- Definire le configurazioni direttamente in Python abbasserebbe la barriera di ingresso per sviluppatori
- Supporto IDE (PyCharm, VS Code) migliorerebbe validazione e velocità di sviluppo
- Mantenere compatibilità YAML per utenti meno esperti

Grazie per l'attenzione!  
Domande?