

**Università degli Studi di Torino**

SCUOLA DI SCIENZE DELLA NATURA

Corso di Laurea Magistrale in Informatica



Tesi di Laurea Magistrale

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

RELATORE

**Prof. Alessandro Mazzei**

CANDIDATO

**Stefano Vittorio Porta**

859133

Anno Accademico 2023/2024



### **Dichiarazione di Originalità**

*Dichiaro di essere responsabile del contenuto dell'elaborato che presento al fine del conseguimento del titolo, di non avere plagiato in tutto o in parte il lavoro prodotto da altri e di aver citato le fonti originali in modo congruente alle normative vigenti in materia di plagio e di diritto d'autore. Sono inoltre consapevole che nel caso la mia dichiarazione risultasse mendace, potrei incorrere nelle sanzioni previste dalla legge e la mia ammissione alla prova finale potrebbe essere negata.*

## **Ringraziamenti**

Todo

## **Abstract**

Todo.

## **Parole chiave**

«classificazionexNLU», «data annotation/augmentation», «NLGBasataSuParafrasi»,  
«ingegnerizzazione»



# Indice

<b>1 Introduzione</b>	<b>1</b>
1.1 Contesto generale .....	1
1.2 Motivazioni e obiettivi della tesi .....	1
1.3 Struttura del documento .....	1
<b>2 Origini e Stato dell'Arte</b>	<b>2</b>
2.1 Chatbot rule-based .....	2
2.1.1 Le Origini con ELIZA .....	2
2.1.2 AIML .....	5
2.2 Natural Language Understanding e Stato dell'Arte .....	8
2.2.1 LSTM e RNN .....	8
2.2.2 Transformers .....	8
2.2.3 LLM .....	8
2.3 Framework e strumenti moderni .....	8
2.3.1 LangChain e Haystack .....	8
2.4 Conclusioni e gap da colmare .....	8
<b>Riferimenti</b>	<b>9</b>





# **1 Introduzione**

## **1.1 Contesto generale**

## **1.2 Motivazioni e obiettivi della tesi**

## **1.3 Struttura del documento**

## 2 Origini e Stato dell'Arte

In questo capitolo verrà fornita una panoramica sui principali approcci per la realizzazione di chatbot e sistemi di interrogazione automatica basati sul linguaggio naturale. Si partirà dall'analisi dei sistemi **rule-based**, che hanno segnato le prime tappe nella ricerca nel campo degli agenti conversazionali, per poi passare ai moderni approcci basati sull'apprendimento automatico con le reti neurali, prestando particolare attenzione ai modelli basati sull'architettura dei transformer.

Il Natural Language Understanding è un campo di ricerca in continua evoluzione, con tecniche e algoritmi costantemente aggiornati e nuovi modelli che vengono proposti e testati.

### 2.1 Chatbot rule-based

Fin dalle origini dello studio dell'elaborazione del linguaggio naturale, i cosiddetti chatbot *rule-based* (basati su regole) hanno costituito il primo approccio per simulare un interlocutore con cui gli utenti possano interagire col linguaggio naturale.

In questi sistemi le risposte del chatbot sono generate sulla base di pattern di input definiti in modo esplicito e di regole o template codificati dall'autore del programma. Tutte le possibili ramificazioni delle conversazioni sono previste e scelte in anticipo, al momento della progettazione.

Nonostante le moderne tecniche di apprendimento automatico abbiano guadagnato terreno, i chatbot rule-based mantengono un ruolo significativo, soprattutto quando è cruciale avere un controllo totale sulle risposte, mantenendo il flusso logico esplicito e operando in un dominio ristretto.

#### 2.1.1 Le Origini con ELIZA

Uno dei primi e più noti esempi di chatbot rule-based è ELIZA, sviluppato da Joseph Weizenbaum al MIT nel 1966 [1], [2].

ELIZA simulava le risposte di uno psicoterapeuta rogersiano<sup>1</sup>, basandosi su un semplice algoritmo di riscrittura (rewrite) delle frasi in input.

L'effetto era sorprendentemente convincente in alcuni casi, poichè ELIZA trasformava abilmente le frasi dell'utente in domande di ritorno, mantenendo un'apparenza di comprensione.

---

<sup>1</sup>Carl Rogers è stato uno psicologo statunitense, noto per gli studi sul counseling e fondatore della «psicoterapia incentrata sulla persona». Questo approccio portava alla costruzione di una relazione collaborativa e di fiducia tra terapeuta e paziente (per Rogers «cliente» dal momento che secondo lui non aveva un ruolo passivo ma attivo nella guarigione), basata sull'empatia, ritenuta essenziale per poter raggiungere il benessere psicologico [3].

Gli uomini sono tutti uguali.  
 IN CHE MODO  
 Ci infastidiscono sempre di una cosa o dell'altra.  
 PUOI PENSARE A UN ESEMPIO SPECIFICO?  
 Beh, il mio ragazzo mi ha fatta venire qui.  
 IL TUO RAGAZZO TI HA FATTA VENIRE QUI.

Listato 1: Un esempio di conversazione con ELIZA da J. Weizenbaum [1]

### 2.1.1.1 Composizione

ELIZA si basava su **script di regole** scritte originariamente in SLIP (un dialetto di Lisp, un linguaggio basato sulla manipolazione dei simboli [4]), organizzate in una serie di **pattern** (decomposition rules) e **risposte template** (reassembly rules). L'idea fondamentale era che ogni regola si attivasse se l'input dell'utente conteneva una certa parola chiave (keyword); ad ogni keyword era associato un insieme di “trasformazioni” più o meno generali, che permettevano di riformulare la frase dell'utente.

```
(HOW DO YOU 00. PLEASE TELL ME YOUR PROBLEM)
START
(SORRY ( ( 0 ) (PLEASE DON'T A P O L I G I Z E )
(APOLOGIES ARE NOT NECESSARY) (WHAT FEELINGS
DO YOU HAVE WHEN YOU APOLOGIZE) ( I I V E TOLD YOU
THAT APOLOGIES ARE NOT REQUIRED)))
(DONT = DON'T)
(CANT = CAN'T)
(WONT = WON'T)
(REMEMBER S
( ( 0 YOU REMEMBER 0) (DO YOU OFTEN THINK OF 4)
(DOES THINKING OF ~ BRING ANYTHING ELSE TO MIND)
(WHAT ELSE DO YOU REMEMBER)
(WHY DO YOU REMEMBER 4 JUST NOW)
(WHAT IN THE PRESENT SITUATION REMINDS YOU OF ~)
(WHAT IS THE CONNECTION BETWEEN ME AND ~))
((0 DO I REMEMBER 0) (DID YOU THINK I WOULD FORGET 5)
(WHY DO YOU THINK I SHOULD RECALL S NOW)
(WHAT ABOUT 5) (=WHAT) (YOU MENTIONED S))
((0) (NEWKEY)))
```

Codice 1: Un frammento delle regole che compongono ELIZA da J. Weizenbaum [1]

#### 2.1.1.1.1 Parole chiave e priorità

Le regole di ELIZA erano raggruppate attorno a delle keyword, che definivano l'argomento o l'elemento su cui il sistema doveva focalizzare l'attenzione. Ad esempio, se la keyword era “REMEMBER”, tutte le regole associate si occupavano di reinterpretare le frasi in cui l'utente accennava a un ricordo.

- Ogni keyword poteva avere una priorità numerica, indicando quanto fosse importante rispetto alle altre. In caso di input multiple che attivassero più keyword, veniva scelta quella con la priorità più alta.

- Se l'utente menzionava "IO NON RICORDO" ("I DON'T REMEMBER"), ELIZA cercava innanzitutto la keyword "REMEMBER" per attivare le regole di corrispondenza più specifiche.

#### 2.1.1.1.2 Decomposition Rules

Ogni keyword era legata a una serie di decomposition rules che descrivevano come scomporre la frase dell'utente in componenti rilevanti. Questo avveniva tramite pattern contenenti placeholder e occasionalmente wildcard. Ad esempio, una decomposition rule per la keyword "REMEMBER" poteva apparire così (in forma semplificata)

```
( 0 YOU REMEMBER 0 )
```

Qui, il simbolo 0 rappresenta dei placeholder che potevano catturare qualunque parte del testo. Se la frase dell'utente era «I think **you remember** when I was a child», il sistema isolava le porzioni di testo prima e dopo la stringa "YOU REMEMBER".

Queste regole permettevano a ELIZA di individuare la struttura della frase e di memorizzare le parti variabili (ad esempio, tutto ciò che seguiva "YOU REMEMBER").

#### 2.1.1.1.3 Reassembly Rules

Una volta trovato un pattern corrispondente, ELIZA applicava una corrispondente reassembly rule, ovvero un template che generava la risposta. Proseguendo con l'esempio della keyword "REMEMBER", se la decomposition rule catturava una parte di frase dopo "YOU REMEMBER", la reassembly rule selezionata poteva essere una delle seguenti:

```
(DO YOU OFTEN THINK OF 4?)  
(WHY DO YOU REMEMBER 4 JUST NOW?)
```

e il sistema rispondeva sostituendo dinamicamente il placeholder con la parte di testo catturata.

Così, se l'utente diceva «Io penso che tu ricordi il mio compleanno», ELIZA poteva rispondere «Perché pensi che io ricordi il tuo compleanno proprio ora?»

#### 2.1.1.1.4 Regole di Fallback

Se nessuna keyword né pattern specifico viene attivato, ELIZA passava a una serie di regole di fallback, spesso domande generiche come:

```
(TELL ME MORE)  
(WHY DO YOU SAY THAT?)
```

Queste tornano utili a mantenere viva la conversazione anche quando non vi è corrispondenza con le regole definite, evitando il silenzio o un evidente messaggio di errore.

### 2.1.1.2 Conclusioni

È evidente comunque come ELIZA non fosse realmente in grado di comprendere il significato delle frasi: si limitava a riconoscere e riformulare le frasi usando parole chiave ottenute dalla conversazione. Un utente che ne è consapevole può potenzialmente «ingannare» il sistema con frasi prive di senso, ottenendo comunque risposte coerenti.

Il mio gatto è morto.

MI DISPIACE PER IL TUO GATTO.

Ora che è morto non so cosa fare.

COSA TI FA PENSARE DI NON SAPERE COSA FARE?

Listato 2: Un listato dal poco senso. (Povero gatto)

Ad ogni modo Weizenbaum riportò (con non poca preoccupazione per ciò che potesse implicare) che diversi utenti iniziarono a fare riferimento al programma umanizzandolo, attribuendogli sentimenti e personalità.

Una personalità che era, se non proprio di un terapeuta, perlomeno di un interlocutore percepito come utile.

Esiste un episodio celeberrimo che dimostra l'efficacia del sistema: anche la segretaria di Weizenbaum aveva iniziato ad interagire con esso. In un'occasione, gli chiese al ricercatore di lasciarla sola nella stanza per poter terminare privatamente la propria consulenza con ELIZA [2].

### 2.1.2 AIML

Negli anni '90 iniziò a guadagnare popolarità il Loebner Prize, una competizione ispirata al Test di Turing [5].

Nella competizione, chatbot e sistemi conversazionali cercavano di «ingannare» giudici umani, facendo credere loro di essere persone reali. Molti sistemi presentati alla competizione erano basati su pattern matching e rule-based, a volte integrando euristiche per la gestione di sinonimi o correzione ortografica.

Tra questi, uno dei più celebri è ALICE (Artificial Linguistic Internet Computer Entity), sviluppato da Richard Wallace utilizzando il linguaggio di markup AIML (Artificial Intelligence Markup Language) da lui introdotto [6], [7]. ALICE vinse per la prima volta il Loebner Prize nel 2000, e in seguito vinse altre due edizioni, nel 2001 e 2004.

#### 2.1.2.1 Struttura di un chatbot AIML

Basato sull'XML, di base l'AIML fornisce una struttura formale per definire regole di conversazione attraverso “categorie” di pattern e template:

- `<pattern>` : la frase (o le frasi) a cui il chatbot deve reagire.
- `<template>` : la risposta (testuale o con elementi dinamici) che il chatbot fornisce quando si verifica il match del pattern.

Considerando ad esempio la seguente configurazione:

```
<category>
  <pattern>CIAO</pattern>
  <template>Ciao! Come posso aiutarti oggi?</template>
</category>
```

se l'utente scrivesse «Ciao»<sup>2</sup>, il sistema risponderebbe con «Ciao! Come posso aiutarti oggi?».

Grazie all'uso di wildcard (\*, -) e a elementi di personalizzazione ( <star/> ), AIML può gestire un certo grado di variabilità linguistica:

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

In questo caso, se l'utente scrivesse «Mi chiamo Andrea», il sistema risponderebbe con «Ciao Andrea, piacere di conoscerti!».

Esistono anche tag per permettere di memorizzare informazioni e utilizzarle in seguito, come <set> e <get>, e per gestire la conversazione in modo più dinamico, come <think> e <condition>:

```
<category>
  <pattern>CHE TEMPO FA</pattern>
  <template>
    <condition name="stagione">
      <li value="inverno">Fa piuttosto freddo, in questa stagione.</li>
      <li value="estate">Fa molto caldo, bevi tanta acqua!</li>
    </condition>
  </template>
</category>
```

È inoltre possibile raggruppare categorie simili con il tag <topic>, e riindirizzare la conversazione verso un argomento specifico con il tag <srai>:

```
<topic name="saluti">
  <category>
    <pattern>SALUTA *</pattern>
    <template>
      <srai>CIAO</srai>
    </template>
  </category>
</topic>
```

---

<sup>2</sup>Caratteri maiuscoli e minuscoli sono considerati uguali dal motore di riconoscimento.

#### 2.1.2.2 Conclusioni su AIML

Grazie ai tag previsti dallo schema, AIML riesce a gestire conversazioni piuttosto complesse. Ciononostante, presenta comunque alcune limitazioni:

- Le strategie di wildcard e pattern matching restano prevalentemente letterali, con limitata capacità di interpretare varianti linguistiche non codificate nelle regole. Se una frase si discosta dal pattern previsto, il sistema fallisce il matching. Sono disponibili comunque alcune funzionalità per la gestione di sinonimi, semplificazione delle locuzioni e correzione ortografica, che devono essere costruite manualmente.
- La gestione del contesto (via `<that>`, `<topic>`, ecc.) è rudimentale, soprattutto se paragonata a sistemi moderni di NLU con modelli neurali che apprendono contesti ampi.
- L'integrazione con basi di conoscenza esterne (KB, database, API) richiede estensioni proprietarie o script custom, poiché AIML di per sé non offre costrutti semantici o query integrate, e non permette di integrare script internamente alle regole.

Nonostante questi limiti, AIML ha rappresentato un passo importante nell'evoluzione dei chatbot, offrendo un framework standardizzato e relativamente user-friendly per la creazione di agenti rule-based. In alcuni ambiti ristretti (FAQ ripetitive, conversazioni "scriptate"), costituisce ancora una soluzione valida e immediata. In domini più complessi, in cui la varietà del linguaggio e l'integrazione con dati dinamici sono essenziali, diventa indispensabile affiancare o sostituire AIML con tecniche di Natural Language Understanding basate su machine learning e deep learning.

## **2.2 Natural Language Understanding e Stato dell'Arte**

Con **Natural Language Understanding** (NLU) si fa riferimento a un insieme di tecniche e modelli che mirano a comprendere il testo in ingresso a un livello più semantico, superando la semplice analisi di pattern o keyword. Negli ultimi anni, la ricerca si è orientata verso modelli di machine learning, e in particolare di deep learning, capaci di catturare caratteristiche sintattiche, semantiche e contestuali.

### **2.2.1 LSTM e RNN**

#### **2.2.1.1 LSTM**

#### **2.2.1.2 RNN**

### **2.2.2 Transformers**

### **2.2.3 LLM**

## **2.3 Framework e strumenti moderni**

### **2.3.1 LangChain e Haystack**

<https://haystack.deepset.ai/> ([https://www.reddit.com/r/LocalLLaMA/comments/1dxj1mo/langchain\\_bad\\_i\\_get\\_it\\_what\\_about\\_langgraph/](https://www.reddit.com/r/LocalLLaMA/comments/1dxj1mo/langchain_bad_i_get_it_what_about_langgraph/))

## **2.4 Conclusioni e gap da colmare**



## **Riferimenti**