

1 Eliza - The First Chatbot

There were 4 major limitations that I improved in the Eliza Python code that is explained below:

1.1 Text Contractions

Why It's a Problem: ELIZA often fails to understand contractions, such as "I'm", "don't", or "you're". This is a problem because contractions are common in everyday language. Misunderstanding or not acknowledging these contracted forms can lead to awkward responses or a communication breakdown, impairing the illusion of understanding.

Why It Happens: This happens because ELIZA uses pattern-matching and predefined rules, which may not account for contractions.

Improvement: I incorporated a contraction expansion module into the ELIZA code called *contractions*. Before processing the user's input, this module expands contractions into their full forms (e.g., "I'm" becomes "I am"), enabling ELIZA to understand and respond to them correctly.

1.2 Spell Check

Why It's a Problem: ELIZA has trouble understanding misspelled words. This can lead to incorrect or nonsensical responses, breaking the illusion of an understanding conversational partner.

Why It Happens: ELIZA is based on pattern matching with predefined rules. If a word is misspelled, ELIZA won't be able to match it with its rules, leading to a failure in understanding.

Improvement: To resolve this, I integrated a spell-checking library called *TextBlob* into the ELIZA code. Before processing the user's input, this library checks for spelling mistakes and corrects them, allowing ELIZA to understand and respond more accurately.

1.3 Lack of Lemmatization

Why It's a Problem: ELIZA doesn't consider words' root form (lemma) in the user's input. This can lead to an inability to recognize different forms of the same word. For example, "run", "running", "ran", and "runs" are treated as different words, leading to potentially incorrect responses.

Why It Happens: The original ELIZA code doesn't incorporate lemmatization. It simply matches the input with the rules defined in its database.

Improvement: I added a lemmatization process to ELIZA's preprocessing pipeline using NLTK's WordNetLemmatizer. With this, each word in the user's input is reduced to its lemma before matching patterns. This allows ELIZA to recognize different forms of the same word, leading to more accurate and appropriate responses.

1.4 Generic Greeting

Why It's a Problem: ELIZA starts every conversation with the same generic greeting, regardless of the time of day or previous interactions with the user. This lack of context-specific greeting can make the conversation feel impersonal and artificial.

Why It Happens: The greeting is hardcoded into the ELIZA code and doesn't change based on context.

Improvement: To make ELIZA's greetings seem more natural and not generic, I implemented a greeting module that generates random greetings making the chatbot feel more engaging and natural.

Note: Demo Video for the Limitations is in the eliza_v2 folder with the name demo.mp4