## Chatbot “Megamind” report

I developed a chatbot called “Megamind” designed to engage with users in a highly personalized manner, mainly capable of general conversations and communication of simple topics, retaining information and replying in a natural manner, recognising general mood in user input and asking dynamic follow up questions in conversation. Megamind’s primary goal is to provide a friendly and intuitive experience though understanding and responding to the user. My main focus when developing this chatbot was to develop a well-crafted robust program capable of understanding a wide range user inputs without crashing or freaking out, with a capability of understanding ranging from full user sentences to single word answers.

### Chatbot application:

Megamind’s domain of operation is general conversation, with a focus on remembering preferences and keeping a conversation going for as long as it is capable. It can recognize patterns related to frequent questions like favourite foods and colours, and is capable of remembering user answers, changing based on its current knowledge of the user. For example, if the user mentions their favourite colour, Megamind will remember this information and reference it in later conversations, providing a more personalized experience.

A use-case for this chatbot could include it as a virtual assistant or entertainment bot, where users could set preferences for the chatbot to remember for future engagements, interact with it in conversation, or make it  spit out fun facts and jokes.

Another use-case could include a learning environment, where native speakers can converse with the chatbot to practice vocabulary, grammar, and pronunciation in real-time.

An important use-case could be in Mental Health, offering conversational support for users experiencing anxiety, helping them manage emotions through empathetic and constructive dialogue.

**Test Case 1:** Detecting and Storing User Preferences

- User Input: "I love pizza!"
- Expected Chatbot Response: "Yum! I’ll remember that you like pizza."
- Behaviour: The chatbot detects that the user is expressing their food preference by matching the input to the "food" intent using a regex pattern like \b(?:my favourite food is|my fave food is|i love) (\w+). The chatbot then stores this information in its memory (i.e., memory["food"] = "pizza"), and acknowledges it by responding with a relevant message.

**Test Case 2:** Detecting and Storing User Preferences (Prompted by bot for favourite colour)

- User Input: "blue."
- Expected Chatbot Response: "Oh, blue is a beautiful colour!"
- Behaviour: The chatbot matches the input with a regex pattern that captures colour preferences specifically when awaiting a response from the user. It then updates its memory to store the colour "blue." If this input is followed by another mention of the colour, the chatbot should offer a personalized response, such as, "I know you like blue!"

**Test Case 3:** Handling of User's Name with Memory Integration

- User Input: "Call me John."
- Expected Chatbot Response: "Nice to meet you John!"
- Behaviour: The chatbot recognizes the user's introduction using a regex pattern (r"\b(?:my name is|call me) (\w+)"). It stores the name "John" in memory and uses it in subsequent conversations by replacing {name} in responses through string substitution. The bot will then use the name in personalized replies, adding a more human-like interaction style.

**Connection to Lecture Material:**

The development of this chatbot is directly related to the ideas discussed in the lecture, using many of the concepts and practices discussed and built upon later in lab. Practically in areas relating to the extensive use of dictionaries, such as Megamind's memory system, the use of complex regex expressions and search matching , the use of string substitution with name, colour and food and the use of Json files to store large intents. This chatbot being an amalgamation of all of these practices put into one.

**Advanced Techniques Used:**

**Use of full and default match types for intent detection:** Megamind uses regex patterns to detect user intent across a variety of phrasing styles. It identifies intents like "colour," "food," or "mood" while avoiding false positives. For example, it prevents misinterpreting "nothing" as a greeting due to the substring "hi" in "not-hi-ng." However, it can still identify help requests regardless of where "help" appears in the sentence.

**Dynamic responses from chatbot based on memory:** The chatbot dynamically adapts its responses based on memory. If a user has yet to share a favourite food or colour and asks Megamind for his, he will answer and then promptly ask you the same question. Now that it knows the answer Megamind will no longer do the same follow up question and will respond differently in the future. This memory-based personalization enriches the interaction and keeps it engaging. Randomized replies from predefined lists also help maintain conversational variety.

**In depth intent detection from user input both prompted and not:** Megamind employs specialized intent detection for both prompted and unprompted user inputs. When awaiting a specific answer, the chatbot uses unique regex patterns capable of interpreting both full sentences (e.g., "My favourite colour is blue") and single-word responses ("blue"). This flexibility ensures robust handling of user inputs.