* [ Dynamic Content Retrieval & Hybrid Approach ]
* Use NLP libraries (like SpaCy, NLTK, or Hugging Face’s Transformers) for more advanced text processing. This can help in better intent recognition, entity extraction, and emotion analysis from user queries.
* For queries that require detailed information retrieval, consider integrating a **custom search engine.** This engine can index the contents of the MS Applied Computer Science website and return relevant information in response to user queries.

**Framework for intents:**

1. Trying to identify different components for intent such as

* Emotion,
* Gain Information,
* Sentiment Analysis : [understanding the overall sentiment of the query, whether it's **positive, negative, or neutral**, and can be used for brand monitoring or improving customer service].
  + Utilize sentiment analysis models (**like those provided by Hugging Face or other NLP libraries**) to determine the emotional tone of the queries.
* **Urgency Detection:** Identifying whether a query requires immediate attention, based on the choice of words or context.
  + Train a model to recognize urgent language patterns or implement **rule-based triggers** for words and phrases that typically indicate urgency.
* Personalization Factors: Extracting data points that can be used for personalizing responses, such as references to **past interactions**, user preferences, or specific needs mentioned in the query.
  + Use data from previous interactions stored in a database. Implement algorithms that analyze past conversations to tailor responses.
* Question Type Classification: Distinguishing the type of question (e.g., **factual, opinion, explanatory**) which can help in formulating the most appropriate response.
  + Train a model to classify the type of question. This requires a **labeled dataset** with different types of questions.
* **Disambiguation:** Clarifying ambiguous queries by identifying multiple possible interpretations and either choosing the most likely one or asking follow-up questions for clarification.

1. Database Storage:
   * Use a relational database (like MySQL, PostgreSQL) or a NoSQL database (like MongoDB) to store and manage your intents and the associated patterns and responses. This allows for more efficient data retrieval and management.
   * Design the database schema to store intents, patterns, and responses in a structured format. This will enable easier updates and expansions of the dataset.
2. Dynamic Content Retrieval:
   * Implement a **web scraping technique** to dynamically retrieve information from the MS Applied Computer Science website. You can use libraries like **Beautiful Soup or Scrapy in Python** for this purpose.
   * **Integrate an API** that fetches the latest information from the website or a relevant data source. This ensures that your chatbot always provides the most current information.
3. Advanced NLP Techniques:
   * Use NLP libraries (**like SpaCy, NLTK, or Hugging Face’s Transformers**) for more advanced text processing. This can help in better intent recognition, entity extraction, and emotion analysis from user queries.
   * Implement machine learning algorithms to handle the complexity of various user queries and to improve the accuracy of responses over time.
4. Integration with a Search Engine:
   * For queries that require detailed information retrieval, consider integrating a **custom search engine.** This engine can index the contents of the MS Applied Computer Science website and return relevant information in response to user queries.
5. Hybrid Approach:
   * Combine pre-defined intents for common queries with dynamic content retrieval for more complex or specific questions. This hybrid approach ensures efficiency and accuracy.