Assignment 1:
Reflexive Web Agent with Tools Use

Student ID: 111502026, Name: 薛耀智

1. **(10%) Describe your Agentic AI application scenario, including target users, use cases, and problems to be solved.**

   Provide a comprehensive description of your application with at least 2-3 specific use cases, clearly defined target users, and concrete problems to be solved.

   This Agentic AI application FilmSeeker is designed to help different types of users solve problems related to movie identification and information retrieval. It primarily targets movie enthusiasts who struggle to recall movie titles, researchers and critics who need quick access to movie information, media content creators who produce film-related content, trivia enthusiasts participating in movie quizzes, and streaming service users looking for movies based on partial descriptions. The application handles various use cases, such as when a user remembers parts of a movie’ s plot, actors, or scenes but cannot recall the title. In such cases, the AI analyzes the input, predicts potential keywords, searches IMDB for relevant movies, cross-references the results, and provides the most probable movie title along with detailed information. Additionally, if users want to find a movie based on a specific scene or theme, such as "a detective investigating a murder in a small town," the AI can extract key elements, generate relevant search terms, and filter out the best-matching results, offering comprehensive details. For film researchers, the application can automatically gather and present structured data about a movie, including ratings, box office performance, director, and screenplay information, saving time and improving efficiency. This system addresses several key issues,

including users’ difficulty in searching for movie titles due to memory gaps, the inefficiency of traditional manual searches, the scattered nature of movie-related information, and the limitations of existing keyword-based search methods when only partial descriptions are available. By leveraging AI-powered inference and structured data aggregation, this application provides an intelligent and efficient solution for movie identification and information retrieval, significantly enhancing the user experience.

2. **(10%) Analyze at least 2 potential technical challenges in implementation and propose preliminary solutions.**

   For each technical challenge, provide detailed analysis including impact assessment and step-by-step solution proposals with feasibility evaluation.

   Challenge 1: Handling Ambiguous or Incomplete User Inputs

   Handling ambiguous or incomplete user inputs is a major challenge for a movie identification AI application. Users often recall only fragmented scenes, specific characters, or vague plot details, making it difficult for the AI to generate accurate search queries. For example, a user might enter a description such as "a guy who can time travel," which is too broad to directly match a specific movie. To address this issue, the application can leverage Natural Language Understanding (NLU) techniques by utilizing LLM models (such as GPT-4 or fine-tuned BERT) to analyze user input, extract keywords, and apply Named Entity Recognition (NER) to detect actors, directors, genres, and other relevant details. Additionally, semantic similarity analysis can help the AI identify related concepts, linking "time travel" to themes like "science fiction, paradoxes, past and future events." To further improve accuracy, the system can engage in interactive dialogue to guide users toward providing more details, such as asking, "Do you remember the main actor in this movie?" or

"Around what year was it released?" Based on their responses, the search scope can be dynamically refined. Finally, the AI can generate multiple queries, combining synonym expansion and semantic matching techniques to ensure the system does not rely solely on keyword-based searches but instead makes contextual inferences to determine the best match. This approach is highly feasible, as NLP models already support semantic analysis, and by incorporating a user feedback loop, the system can continuously learn and optimize query precision, enhancing the overall search experience.

Challenge 2: Efficient Cross-Referencing and Ranking of IMDB Search Results

When it comes to cross-referencing and ranking IMDB search results, relying solely on keyword matching can lead to inaccurate results. Multiple movies may share similar descriptions, or certain keywords may be too generic, making it difficult to pinpoint the exact film that matches the user's description. To overcome this, AI needs a more intelligent feature extraction and ranking mechanism. First, the system can extract metadata from IMDB search results, including the title, genre, director, actors, plot summary, release year, and relevant keywords. These details can be converted into vector representations, allowing the AI to evaluate semantic similarity between user input and each retrieved movie using models like SBERT. Next, the application can implement a weighted scoring mechanism, assigning different levels of importance to various metadata fields based on the user's input. For instance, if the user mentions a specific actor, the system can increase the weight of cast matching, while if the user describes a plot element, plot similarity will have a greater influence on ranking. Furthermore,

the ranking algorithm can integrate TF-IDF or BM25 keyword matching scores, along with IMDB ratings or movie popularity scores as additional ranking factors. To further improve accuracy, the application can employ adaptive machine learning, refining its ranking model based on user feedback. By asking users questions like, "Is this the movie you were looking for?" and analyzing clicks or feedback responses, the system can adjust its ranking dynamically using reinforcement learning. This approach is technically feasible, as semantic matching and vector search technologies are already well-established. For instance, FAISS can be used for efficient semantic lookup, and continuous feedback collection can help refine the model over time. As a result, this ensures that search results become more precise, ultimately providing users with the most relevant movie information.

3. **(20%) Explain how your system implements the complete cycle of environment perception, decision making, and action execution.**

   Detail the complete workflow of your system, demonstrating how each component interacts within the perception-brain-action cycle.

   The Environment phase involves gathering information and understanding the context of the user's input. The system receives a description of a movie from the user, which may include details such as actors, plot fragments, release year, or themes. This input serves as the foundational data for the AI to extract keywords and contextual clues.

   In the Perception phase, the system utilizes advanced Natural Language Processing (NLP) techniques to analyze and interpret the user’ s input. It employs a Large Language Model (LLM) (such as

GPT-4o-mini) to extract key movie-related information, including actors, genres, themes, and keywords. Based on the extracted information, the system generates multiple movie title keywords.

The Brain phase represents the system's reasoning and planning capabilities. At this stage, the system performs an IMDB search and processes the retrieved results. It sends multiple optimized queries to the IMDB search engine and retrieves the top five search results based on the generated keywords. The system then extracts key metadata from these results, including title, cast, and plot summary, and determines which movie best matches the user's description based on this metadata.

In the Action phase, the system executes decisions and presents the most relevant movie information to the user. It displays detailed movie data, including the movie title, director, screenwriter, cast, plot summary, IMDB rating, and popularity score.

By following this structured Environment → Perception → Brain → Action cycle, the system provides a dynamic and intelligent movie identification process. Each phase helps optimize search results, improve accuracy, and enhance the user experience through adaptive learning. This approach ensures that the system efficiently processes ambiguous user input, generates meaningful search queries, and continuously improves its predictive capabilities through interactive feedback, offering a more accurate and intelligent movie identification solution.

4. **(30%) Design and execute 3 test tasks, analyze the results, and propose potential improvements based on the current implementation.**

   Document the execution of three test cases with comprehensive analysis of

results and specific improvement suggestions.

After analyzing the three test cases from the system logs, I evaluated the accuracy of movie identification, query processing, and result optimization. Below are the detailed results and potential improvements.

Test Case 1: “I am looking for a movie that a robot pretending to be a human and trying to assassinate the main character. The movie is the second installment in a series.”

The system successfully identified the movie Terminator 2: Judgment Day based on the provided plot description and series information. However, some issues were found in the response, including redundant entries in the cast list, with certain actors being listed multiple times. Additionally, the IMDb description emphasized that the Terminator is a "protector" in this film, while the user described a "robot assassin," which could cause confusion in cases where the description is ambiguous. To resolve these issues, the system should ensure that cast lists do not contain duplicate entries. Furthermore, it should enhance its ability to distinguish between villains and heroes, performing a plot-based contradiction check and adjusting the response or prompting the user for clarification. In future versions, the system should also improve its ability to recognize sequel numbering and use IMDb's movie series lists to ensure that the movie provided aligns with the user's intended installment.

Test Case 2: “I am looking for a movie that starring Robert Downey Jr., Chris Evans, and Scarlett Johansson.

The system correctly identified Avengers: Endgame, which featured all the actors specified by the user. However, these actors also appeared together in several other movies, such as The Avengers, Captain America: Civil War, and Avengers: Infinity War. The system did not provide these options or explain why Endgame was chosen as the best match. To address this, the system should improve its ranking mechanism for multiple matching movies. If several films meet the criteria, the system should list them in order of relevance, allowing the user to choose. Additionally, the system should explain its selection, such as stating, "This is the latest movie featuring all three actors together and a key milestone in the Marvel Cinematic Universe." If multiple matches exist, the system should prompt the user with a clarification question like, "These actors have appeared together in multiple movies. Are you referring to the Avengers series?" to ensure greater accuracy.

Test Case 3: "I am looking for a movie starring Keanu Reeves and the movie is about virtual reality."

The system successfully identified The Matrix based on the keywords "Keanu Reeves" and "virtual reality." However, the response also included sequels such as The Matrix Reloaded, The Matrix Revolutions, and The Matrix Resurrections, without explaining their relevance to the query. Additionally, the system did not confirm whether the user was referring to the original movie or one of the sequels, which could lead to confusion. To improve this, the system should clearly mark the best match, stating explicitly, "This is the best match for your query." It should also prompt the user to specify if they are referring to a sequel by asking, "Are you referring to the first movie (1999), or a sequel?" Moreover, refining the system's franchise-awareness logic would ensure that when a

user describes a concept introduced in an early film, the system prioritizes the correct installment.

Based on the test cases, the system should implement the following improvements:

1. Enhance the ranking mechanism for multiple matching movies to ensure reasonable ordering when multiple films fit the criteria, helping users make an informed choice.
2. Improve query expansion and refinement by prompting users for additional details when their input is vague, ensuring more accurate results.
3. Eliminate redundant information in responses by ensuring that cast lists and other movie data are concise and free of duplication.
4. Optimize handling of sequels and franchises by clearly marking the best match and prompting users to specify if they are referring to a particular sequel.

The system successfully identified the correct movies in all three test cases, demonstrating strong keyword extraction, IMDb search execution, and metadata retrieval capabilities. However, improvements in handling multiple matches, recognizing franchise installments, and refining responses would enhance accuracy and user experience. By implementing these optimizations, the system will provide smarter, more precise, and more user-friendly movie identification results in the future.