Stage 6: Process Actions

Thomas Gagne Jon Sims

CS425, May 13, 2016

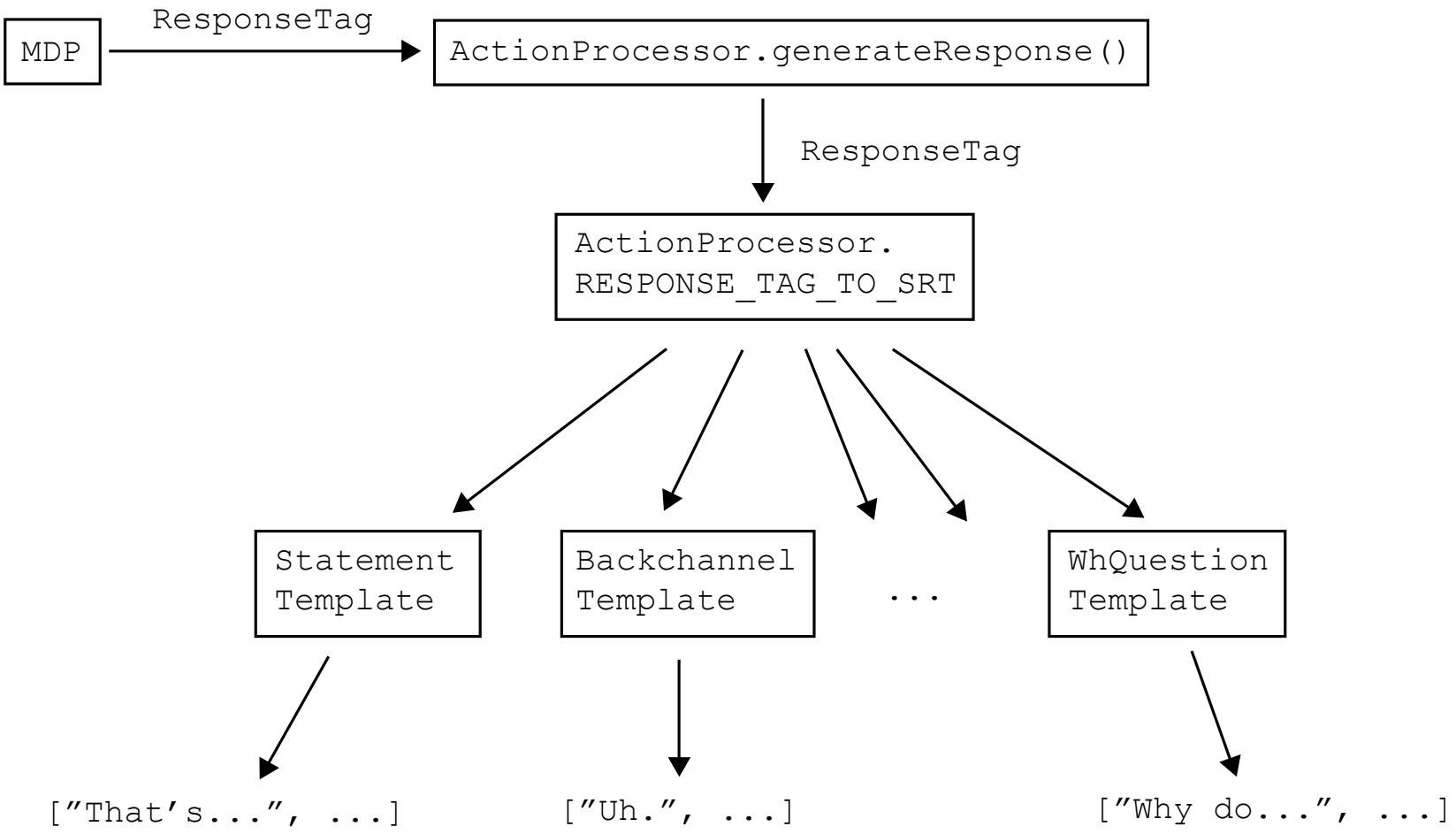
Project Purpose

1. Take in a Response Tag from the MDP which determines what type of response we should present to the user, such as: `Yes-No-Question` or `Conventional-Opening`.
2. Use the information gathered in the conversation plus the knowledge stored in the knowledge base to produce a text response.
    - Two methods of response generation:
        i) "Dumb" responses, which are simply picked from a collection of hand-written responses.
        ii) "Smart" responses, which involve using the knowledge base and the conversation so far to form a semantic response in AMR format, which is then converted to text.

What We Accomplished: Dumb Response Generation

1. We created a list of appropriate Response Tags and matched each Tag to a Response Template, which describes how to generate a response matching that particular tag.
2. For each Response Template, we developed a "dumb" way to generate responses.
   - 2.1 Most Templates have a hand-written list of responses which they choose randomly from.
   - 2.2 Some Templates have rudimentary thinking, such as the `Yes-No-Answer` Template which sends the user's utterance to the knowledge base's `yesNo()` method and uses the output to determine whether to return "Yes" or "No."

What We Accomplished: Dumb Response Generation



What We Accomplished: Smart Response Generation

- There are three required components to generating "smart" responses:
  1. Conversion of the user's input to AMR form.
  2. Algorithms for each Response Template which use the knowledge base and the user's AMR to form a semantic response in AMR format.
  3. Conversion of the AMR semantic response to English text.
- We accomplished the first and third of these.

What We Accomplished: Smart Response Generation

- To convert arbitrary sentences into AMRs, we wrote two primary utilities:
  1. A `msrsplat.py` Python script which calls Microsoft's MSR SPLAT tool which converts sentences into their AMR form, returned as a String.
  2. A function which takes in the String AMR returned by Microsoft's tool and parses it to create a corresponding `AMR` Java Object which can be manipulated.

What We Accomplished: Smart Response Generation

- What we think is necessary to complete smart response generation:
  1. For each Response Template, an algorithm needs to be developed which can analyze information from the user's input in AMR form, pass this information to the knowledge base, and construct a semantic response in AMR format.
     - We suggest that each Response Template has a collection of AMR templates which are filled with information from the knowledge base to form a semantic response.
  2. An algorithm needs to be developed which can convert an AMR into its corresponding English text.
     - We suggest using a pattern-matching scheme on the individual nodes in the AMR, where a node matching a pattern allows one to form the stub of the sentence which corresponds to that node in the AMR.
     - This approach requires deep integration with tools such as WordNet, PropBank, and FrameNet.