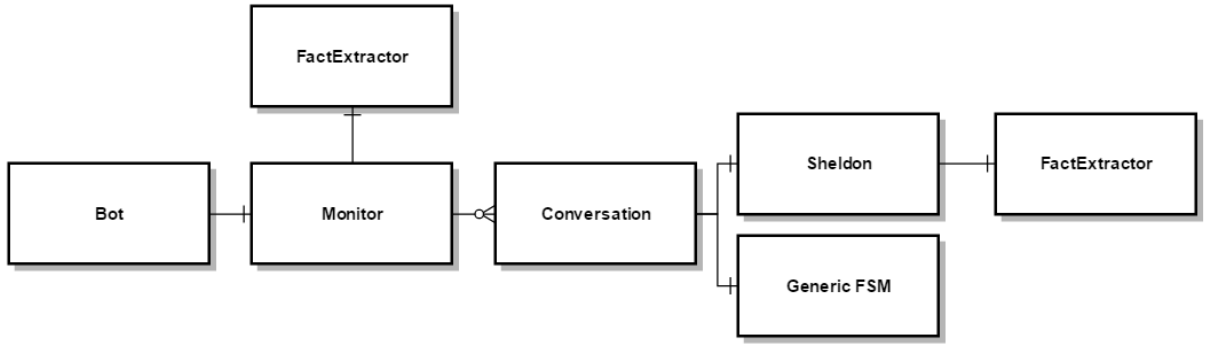
Lab4 : Sheldon Chatbot

By: Skylar Durst, Ivan Pachev, and Jeremy Kerfs

CPE 582 - Spring 2016 - Foaad Khosmood

Description

System Design



Our system design consists of several components, which are depicted in Figure 1:

- Bot
  - This component simply gets each incoming message from the IRC channel and passes it to the monitor.
  - The Bot also submits outgoing messages to the IRC channel.
- Monitor
  - This component manages Conversations by maintaining a mapping between conversation partner name and Conversation.
  - The Monitor uses this mapping to submit incoming messages from the Bot to the conversation associated with that conversation partner.
- Conversation
  - This component keeps track the personality associated with the conversation, as well as a message queue and a thread used to process the conversation.
- Personality (Sheldon/Generic)
  - These personalities dictate how the incoming response is processed and how the outgoing response is created.
  - Depending on how the conversation is started, we choose the appropriate personality.
    - Generic FSM is the phase 2 complex greeting personality.
    - Sheldon is our phase 3 personality and uses the fact extractor to extract important noun phrases from incoming messages and uses them to look up relevant data in the corpora to produce an outgoing message.

Preprocessing pipeline

We use a preprocessing pipeline in order to extract information from all incoming messages, which is illustrated in Figure 2. The pipeline takes as input a string containing the input message. The incoming message string is processed using of the following stages in our pipeline:

1. Word tokenizer
   - The word tokenizer divides messages into words for further processing

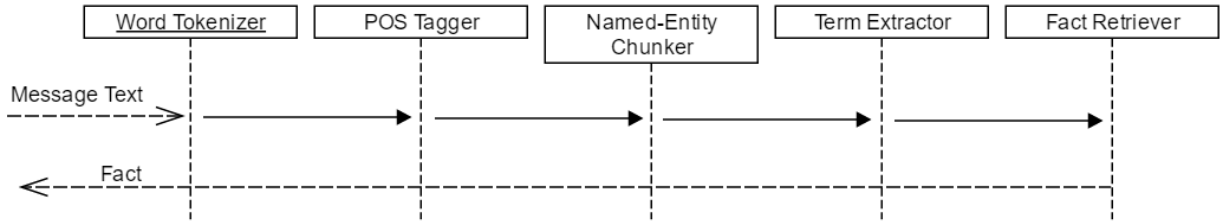


Figure 1: Fact Generation Process

2. Lemmatizer
   - The lemmatizer replaces words with their more common, synonymous counterparts.
   - This facilitates more accurate POS tagging, named entity recognition, and noun-phrase extraction.
3. POS tagger
   - The POS tagger attempts to categorize the words in the message based on their parts of speech.
   - We use the built-in, pre-trained POS tagger in NLTK.
4. Named-Entity Recognizer
   - The named entity recognizer attempts to determine if any of the tokens in the message are named entities.
   - It also combines terms that it believes are part of the same named entity.
     - For example, "Los Angeles" will be grouped as one named entity.
   - We use the built-in pre-trained NER in NLTK.
     - It does not perform very well, but it seems to do well enough for this purpose, especially combined with the next step.
     - It seems to miss obvious named entities that are not capitalized, such as "california".
5. RegExp Parser
   - We use a regular expression parser to extract noun phrases from the POS and NE-tagged tokens.
   - Our RegExp parser prioritizes noun phrases containing named entities, but also extracts normal noun phrases.
     - We are able to extract and prioritize noun phrases containing named entities because our grammar allows us to label them differently.

The output of the pipeline is a list of noun phrases, organized with the noun phrases containing named entities first. This helps us try to pick the most important noun phrase, which we can then lookup in our corpus to pull out facts.

Implementation

Originally, we were looking up data from wikipedia on-the-fly, but we decided it's much faster by using locally stored corpora.

Testing

Corpora

Table 1: Corpora used in Chatbot

Content	Url
Sheldon Quotes	http://the-big-bang-theory.com/quotes/character/Sheldon/
Physics Facts	https://en.wikipedia.org/wiki/Category:Concepts_in_physics

Content	Url
Computer Sciencist Facts	https://en.wikipedia.org/wiki/List_of_computer_scientists
Astonomy Facts	https://en.wikipedia.org/wiki/Outline_of_astronomy
Computer Science Facts	http://corpus.byu.edu/wiki/

Resources