

```

words_of_message = speech_text.split()
if set(check).issubset(set(words_of_message)):
    return True
else:
    return False

if check_message(['who', 'are', 'you']):
    general_conversations.who_are_you()

elif check_message(['how', 'i', 'look']) or check_message(['how', 'am', 'i']):
    general_conversations.how_am_i()

elif check_message(['tell', 'joke']):
    general_conversations.tell_joke()

elif check_message(['who', 'am', 'i']):
    general_conversations.who_am_i(name)

elif check_message(['where', 'born']):
    general_conversations.where_born()

elif check_message(['how', 'are', 'you']):
    general_conversations.how_are_you()

else:
    general_conversations.undefined()

```

The main change in this file is the code edit in the `check_message()` function (in addition to the additions of the conversation snippets in the `if/else` ladder). Let's analyze the changes in `check_message`. First, you split the `speech_text` string and store it in a variable called `words_of_message`. This results in an array of words that are present in the speech.

Note that the `check` argument in the updated `brain.py` file refers to an array of strings (not a string, as in the previous version). You then make a set of `check` and `words_of_message`, which removes any duplicate words. Finally, you check whether the set `check` is a subset of the set `words_of_message`. If it is a subset, then it returns `True`; otherwise, it returns `False` (see Figure 3-2).