Restaurant order analyzer service

Create a containerized http service that should implement two endpoints given below. You may use any language of choice.

GET /health -> Returns 200 when service is available

POST /detect_intent

This takes a single string parameter “message”, and responds with a string body containing the output.

The solution:

- Should be a git repo (need not be on github, local git is fine)
- Should have a pre-commit hook that runs unit tests (test cases included below)
- Should have a dockerfile that makes the service available

We will test it by:

- Validating the pre-commit hook
- Building and running the docker container, and testing it with examples from the algorithm description.

About the restaurant order analyzer service

We wish to build a service that automates ordering from pizza restaurants. We need your help to build a backend service that will classify the incoming conversation messages so that our software can respond appropriately.

We have chat logs from previously completed calls to help with classifying the incoming messages.

Each chat message contains:

- An utterance - the words that were spoken, e.g. I'd like to order a pizza for pickup please.
- An intent - a reusable string label that identifies the intended purpose of the spoken words, e.g. OrderForPickup.

The backend service should accept an incoming chat message (string), and will need to determine what intent we should use to label it. To achieve this, it will need to compare the (input) new message against the corpus of messages we've seen in our past chats. When it finds a similar message, it can return the intent (string) that was used to label the past message.

To determine similarity, we can calculate the intersection over union (*Jaccard Similarity*) – so when we receive a new message, we will need to efficiently determine the following from each historical message:

- The intersection (words that appear in both messages).
- The union (words that appear in either message).

For example, given a historical message

```
{
    utterance: "OK, and what's your order?"
    intent: "ReadyToReceiveOrder"
}
```

and an inbound message `"I'm ready for your order."`:

The intersection is the words "your" and "order".
The union is the words "OK", "I'm", "and", "ready", "what's", "for", "your", and "order".

We can then calculate 2 / 8 = 0.25. If this score is the highest, then we will return the intent "ReadyToReceiveOrder".

So at the core, we have a function called jaccardSimilarity(string s1, string s2) that returns the similarity between two strings.

You can use the following for test cases for this function:

```
"OK, and what's your order?"
"I'm ready for your order."
Output: 0.25

"I'd like a medium supreme pizza."
"Can I get a pepperoni pizza, medium?"
Output: 0.3

"Hi there, how can I help you?"
"Hi there, how can I help you?"
Output: 1

"Is your order for one large pizza?"
"Thanks, please come again."
Output: 0
```

Now we can use the function to query the prior conversations, and return the Intent with the best match.

Test cases:

```
Input: "OK, your order is a large pizza and garlic bread."
Output: "ConfirmItem"

Input: "Ready in 30"
Output: "DurationBeforePickupAnswer"
```

Additional notes

Most languages have “set” types to help with this - or if not, sets can be implemented based on maps.

Prior conversations

Here’s the list of previously analyzed conversations:

```
{
  "chatLog": [
    {
      "utterance": "Hi, Mario's what can I get you?",
      "intent": "Greeting"
    },
    {
      "utterance": "I'd like to order a pizza for pickup please.",
      "intent": "HowCanIHelp"
    },
    {
      "utterance": "OK, what would you like to order?",
      "intent": "ReadyToReceiveOrder"
    },
    {
      "utterance": "I'd like a medium supreme pizza.",
      "intent": "OrderItem"
    },
    {
      "utterance": "Anything more?",
      "intent": "AnyMoreItems"
    },
    {
      "utterance": "Also, a garlic bread.",
      "intent": "OrderItem"
    },
    {
      "utterance": "Is that all?",
      "intent": "AnyMoreItems"
    },
    {
      "utterance": "Yes, that is all thanks.",
      "intent": "EndOfOrder"
    },
    {
      "utterance": "OK, your order is a medium supreme and a garlic
bread.",
      "intent": "ConfirmItem"
    },
    {
      "utterance": "Should be ready in about 30 minutes.",
      "intent": "DurationBeforePickupAnswer"
    },
    {
      "utterance": "Thank you, goodbye.",
```

```
      "intent": "Goodbye"
    }
  ]
}
```