UDACITY

Logout

PROJECT

Bookworm

A part of the Artificial Intelligence Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Absolutely Awesome work! It can be seen that a lot of hard work has been put into this submission.

Here are some useful resources that you can look at, to further improve your understanding:

- For info on Discovery API :
  - https://www.ibm.com/watson/developercloud/discovery/api/v1/
  - https://www.ibm.com/watson/developercloud/discovery.html
- For info on Conversation API :
  - https://www.ibm.com/watson/developercloud/conversation/api/v1/
  - https://www.ibm.com/watson/developercloud/conversation.html

Congratulations!! You have successfully completed the AIND project **Bookworm**.

1\. Create and configure Discovery service

Notebook connects to a Discovery service instance and creates/fetches an environment.

Awesome start!

- The environment `Bookworm` is successfully created/found.

The service processes a small sample text and returns enriched output.

Great Work!

The step `enrichments_output` executed by the service is seamlessly processing the sample text and returning the enriched output tuples.

All inline questions are answered correctly based on the output, and a word-cloud of keywords is shown.

Nicely done!

- Good job identifying the `docSentiment` and various concepts with `relevance > 0.5`.
- Great observation on identifying an ambiguous relation and identifying the solution to resolve its ambiguity.
- The WordCloud is able to create the word visualizations proportional to their respective relevance.

Suggestion:
There is another relation with much clearer/larger ambiguity. Check for disambiguity in this relation:

```
{
"sentence": " During the battle, Rebel spies managed to steal secret plans to the E
```

```
mpire's ultimate weapon, the DEATH STAR, an armored space station with enough power
 to destroy an entire planet.",
  "subject": {
    "text": "Rebel spies",
  },
  "action": {
    "text": "to destroy",
  },
  "object": {
    "text": "an entire planet",
  }
}
```

2\. Ingest documents

A collection is created and documents are added to it, one document per paragraph of text.

Awesome:

- `Story Chunks` as a new collection is created and sample documents are added to it with each paragraph being treated as a separate document in the data-set.

A simple query is made against the collection, and relevant results are returned. Inline questions are answered correctly.

Great work!

- Nice work querying the data-set for the titles pertaining to your requirement.
- Kudos on correctly writing what the query does, in plain terms.

3\. Parse natural language questions

At least 3 intents are added to the Conversation service, each with at least 5 example utterances. Inline questions are answered adequately.

Again, nice job!

- Great work adding the intents : **#who, #where, #what** and listing a fairly good set of questions targeted towards the **#who** intent.

Suggestions:

- Try increasing the number of example utterances for each intent and possibly, number of intents too.
- Increasing the number of utterances would help you get better results. You can try to give **10-12** example utterances per intent. :)

At least 3 entities are added to the Conversation service, each with at least 1 example entity. Inline questions are answered adequately.

Awesome

- Great work here too. Listing the entities with the corresponding examples.

Important Trivia

- You should notice that based on the example utterances provided for the intents, the entities types result may change.

An appropriate dialog flow has been designed using the Conversation service workspace tool, with at least 3 nodes. Inline questions are answered adequately.

Brilliant work creating the dialog nodes and setting appropriate triggers corresponding to them.

A simple 1-question dialog is demonstrated in the notebook, showing what node was triggered.

- The notebook is correctly showing the `entity` as **Person** with values **Luke** and **Luke Skywalker** with confidence **1** which were triggered through the intent **who** with confidence **0.658**. Good job there!
- Notice the response "Let me find out who for you" after the intent has been triggered.

4\. Query document collection to fetch answers

A sample question is run through the Conversation service. The intent and entities identified are extracted, and optionally the dialog node that was triggered.

Awesome work on taking up the optional task too.

- The `intent` `entity` and `dialog_node` for the query are the correctly identified.
- Great work printing the dialog_node triggered.

A query is designed based on the information extracted in the previous step, and run against the Discovery service collection.

Awesome work meeting this specification!

- Good work formulating the query and using filters on `enriched_text`.

- Your query is able to return the `document_id`, `passage_score` and `passage_text` which is brilliant.

Suggestion: You may try formulating different types of queries against the Discovery service collection, and observe the results you obtain.

Results obtained from the Discovery service are processed to provide a specific response to the natural language question that was asked.

Absolutely Brilliant work processing the results.

- Great work attaining the `Entity locations` and `Total location counts` through your implementation.

5\. Reflections

Inline question adequately answered, including strengths and weaknesses of an API-based solution like this.

Awesome work!

- Good work listing the strengths and limitations of the two IBM Watson Services, in detail. :)

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ Reviewer Agreement