

COMP 10261: Statistical CS Bot Project

Sam Scott, Mohawk College, March 2023

OVERVIEW: AN FAQ BOT CREATED BY STATISTICAL NLP

For this project, you will use the existing **FAQ Bot Plus** code, text classification, and other tools either to rebuild your existing **FAQ Bot Plus** chat bot for a different purpose. The rest of this assignment assumes that you are rebuilding **FAQ Bot Plus**. If you decide to create a new chat bot, it should still meet the requirements. You should check in with your instructor to make sure your plan will meet the criteria.



For this assignment, you don't need a standalone bot, just the Discord version will do.

WeChat: cstutorcs

STEP 1: INTENT MATCHING

Convert your intent matching system so that it uses **word frequency vectors** and **similarity measures**.

1. Load your questions and convert the entire set to **document vectors** using **CountVectorizer**, **TfidfVectorizer**, or similar. You should have AT LEAST 2 versions of each question with different wordings. The more versions of each question, the better.
2. In the **understand** method, compute the Cosine or Euclidean similarity of each new utterance to the existing question documents. The best match determines the user's intent. But you should also have a baseline similarity that the intent must meet. For example, if your best-matching intent has a cosine similarity less than some value **t** then it is not a good enough match and you do not return a matching intent. You'll have to experiment to find the best value for **t**.
3. Test your **understand** method using at least the 40 questions you generated for the previous project, plus other questions if necessary. Experiment with parameters like **min_df**, **max_df**, **stop_words**, **ngram_range**, and **tf-idf** vs. **word frequencies** to try and find the best performance. You could also experiment with **stemming**, **lemmatization**, and other techniques. Feel free to add or remove words from your original questions to get better matching behavior. **Do not just use the default CountVectorizer or TfidfVectorizer vocabulary settings.**

STEP 2: TEXT CLASSIFICATION

程序代写代做 CS编程辅导

Use a **pickled** classifier object to classify the user's utterance for emotional tone or for some other feature that your bot can respond to (speech act, topic, etc). For example, you could use the **sentiment** corpus (or similar) to classify the user's utterance and respond with an apology and a suggestion of where to go to speak to a human if the user seems upset. Or you could use the **speech act** corpus (or similar) to recognize the user's speech act and respond appropriately.



1. In a separate python file, use different **classifiers**, **text representations**, and other techniques to find the best approach on the task you have chosen. (i.e., split into testing and training, measure accuracy, precision, recall, etc.)
2. When you are satisfied, train the classifier on the **entire data set** and **pickle** it.
3. Load the pickled classifier into your FAO Bot Plus and use its classifications to generate responses. Exactly how you use the result of the classification is up to you.

WeChat: estutores

STEP 3: TRANSFER LEARNING

Assignment Project Exam Help

This last part is a little open-ended. The requirement is that you implement at least two use cases of deep or transfer learning in your bot. Here are some options (you can use more than one):

Email: tutores@163.com

1. Use **word embedding vectors** from **spacy** to train and pickle an **MLP** for step 2.
2. Use the **OpenAI gpt-3 transformer API** to implement a second classification system for speech acts or sentiment or some other set of labels. (For example, maybe you could have an **on topic** vs. **off topic** classifier.)
3. Use the **OpenAI gpt-3 transformer API** to respond to user utterances that are unexpected or don't fit a predetermined intent. **Make sure you provide extra context in the prompt** to keep the bot on topic as much as possible and avoid having it produce misinformation.
4. Use the **OpenAI gpt-3 transformer API** to generate a question response when an intent cannot be determined. **Use a prompt to tune it into your topic**, but maybe preface the response with "this is just a guess", or "I'm not 100% sure of this, but..."

QQ: 749389476

https://tutores.com

IMPLEMENTATION NOTES

Your bot should respond quickly. Slow processes like classifier training, vectorization of a corpus, loading a spacy language model, etc. should be pickled or performed one time upon startup, not each time an utterance is processed.

Your bot is going on Discord, so it should not respond to every utterance. Make it so that it responds on a dedicated channel, to a command prefix, only to users who have "turned it on" with a command, etc.

OPTION: A HYBRID BOT

程序代写代做 CS编程辅导

If you're happy with the FAQ Bot Plus performance from the first project, consider augmenting it instead of rebuilding it from scratch. For example, you could have two intent-matching systems, one based on regular expressions and one based on a neural network. You could also leave some of your **spaCy** entity recognition and pattern matching classifiers and transfer learning classifiers and transfer



OPTION: DIALOG

Just because it's an FAQ bot doesn't mean it can't do some dialog management. Can you build in some capacity for the bot to remember who it's talking to, what it's said in the past, etc?

WeChat: cstutorcs

OPTION: A DIFFERENT TYPE OF BOT

If you decide to create a different kind of chat bot, you must make sure that you are using **similarity**, **text classification**, and **transfer learning** somewhere in the design. Make sure you pitch your idea to your teacher first.

Email: tutorcs@163.com

REPORT

You should create a brief report with the following sections:

QQ: 749389476

INSTRUCTIONS

- How do I run your bot? Do I need to add it to a server? Is there a function I need to run from the shell? Make sure you write complete instructions that will get me interacting with your bot, including an invitation link if necessary.

https://tutorcs.com

VECTOR REPRESENTATIONS

- What vector representation did you try for intent matching (i.e. tf-idf, word frequencies, n-grams, etc.)? Why did your chosen representation work better than the others that you tried?
- What vector representation did you use for your classifier? Why did this representation worked better than the others that you tried?

TRANSFER LEARNING

- Describe how you incorporated deep learning, including the architecture you chose for the MLP if you used one, the approach you took to the prompt text if you used GPT-3, etc. Be exhaustive and specific in describing what you did.

THE PIPELINE

- What does your chat bot pipeline look like? List the components of your pipeline and walk me through the steps you took in processing each utterance.

THINGS TO TRY

- Give me a list of 10 bot that will produce interesting behavior and show off the capabilities you bot (intent matching, emotional tone, speech act classification, gpt-3



HANDING IN

Zip up and hand in your report, your bot code, supporting data files (tokens, questions and answers, training data, etc) and (in separate folders) any other code you used in the development of your bot (e.g. testing and pickling classifiers). Hand it in to Canvas. See Canvas for the due date.

RUBRIC

Your project will be marked using the rubric on Canvas.

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>