Data-science exercise: Generate an FAQ bot

Backstory

In this exercise, we will prototype a bot to answer frequent questions on a forum by analyzing said forum. We will not implement the full bot and rather focus on what the bot could say when. We limit ourselves to a simple FAQ bot which at most responds to the first post of a thread.

We imagine that this bot will assist a company that wants to validate everything which will be displayed to users. As such, the bot cannot generate answers or even use templates. Instead we have to come up with a limited set of pre-defined strings ("answer strings") which are reused. If the bot proposes an answer to a thread, it must be one of these answer strings. The number of strings we can pick will be limited and additionally each answer string can be at most 1000 characters long. Note that the these strings are arbitrary and not limited to what has been said in the forum. By choosing more general answer strings, they can be used for a number of similar questions.

We need to judge whether the bot provides good answers to questions. For this, imagine each thread starter being forced to either click a button to accept an answer, or post a response asking for a human answer. If the user prefers to do the former, the answer was accepted and otherwise rejected. When the bot does not propose an answer, it is neither accepted nor rejected. For the company, the goal is to improve customer impressions and have their staff spend less time answering common questions. The forum manager thinks it is net neutral for the company if for every one accepted proposed answer, there are ten rejected.

Example questions and answers

Sample questions in new threads and appropriate answers for questions on drip-coffee machines. Note that only two answer strings are used for three proposed answers.

Q. Getting coffee grounds in the canister - Help! Am I using the wrong kind?
A. Make sure the coffee is middle ground, the filter of the right size, and do not overfill grounds.

Q. Machine won't start. Can I fix it?
A. (pass)

Q. This machine is nasty - what's the best way to clean it?
A. Wipe the machine off inside and out, scrub if needed, and run machine with vinegar then water.

Q. There's debris in the coffee - why does this keep happening?
A. Make sure the coffee is middle ground, the filter of the right size, and do not overfill grounds.

Tasks

For the following questions, the conclusions and motivations are the most interesting. Code should however be provided which can replicate the conclusions. Python is preferred. Auxiliary data and sources can be used so long as the work can be reproduced, e.g. downloads provided. Code quality and good practices are secondary to results. Methods do not have to be sophisticated and can involve manual steps of limited work. If you think the project would benefit from stakeholder clarifications, note the queries and assumed reasonable answers. You should be the only human involved in the project.

The imagined (unaffiliated) target for the bot is this forum: https://lgcommunity.us.com/categories/4k-ultra-hd-tvs
Scrapes of relevant pages are available: https://drive.google.com/file/d/1xo_cBIUg6Ha3JremD2_eAe9P1tL8pZ43
There is no assumption in this exercise that we should understand how the product works.

Question 1. Which were the first ten threads of 2019? Which of these threads could be considered questions? Which received answers? For each answer, what is your gut judgement for how likely the answer would have been accepted by the thread starter? Why?

Question 2. How many initial posts from 2019 concerned problems connecting the TV to Wi-Fi? Do any of these have answers which potentially could have resolved the problem? If you were to only make a bot for answering Wi-Fi threads, what answer string do you think would be the best? Give in code or pseudocode a method which given the initial post determines whether to respond to it with this Wi-Fi answer string. How many of the 2019 wifi threads would your answer string be used for and what do you think would be the expected acceptance rate of these answers?

Question 3. Analyze the forum and come up with the best five answer strings for the bot. Produce and share a CSV which contains a row for each thread started in 2019 with the thread URL, the first post, and which of the five answer strings the bot should respond with, if any. Provide an estimate for how many of those responses would have been accepted respectively rejected.

Question 4. Suppose that we could do the work of Question 2 but for up to 50 answer strings. Estimate the number of accepted and rejected proposed answers for 2019 threads for the best one answer string, for the best two, etc, up to the best 50 answer strings. You do not have to come up with 50 specific answer strings nor generate any CSV. Only the estimates and motivations are needed.

Question 5. From your experiments, what automation would you propose for the company? How much value would you argue that it generates for them? Are there any other insights which you think could benefit the company? (100-character responses are good enough.)