SI 650 / EECS 549: Homework 3 – An Information Retrieval-based Chatbot

Part 1 Due: Wednesday, October 28, 4:30pm
Part 2 Due: Wednesday, November 11, 4:30pm

Introduction

Information Retrieval (IR) has many applications outside of search engines. In Homework 3, you'll be witnessing this in action by developing a chatbot using IR techniques.[1] While chatbots, also known as conversational agents, seem like a far stretch from IR, they have a natural synergy if you start from a collection of conversations. In the IR formulation, you can consider a message from a user m as a query and the IR system must rate the set of possible responses $R = \{r_1, \ldots, r_n\}$ according to their relevance.

This assignment consists of two parts that have separate deadlines. In Part 1, you will collective create a ground truth dataset of conversations ranked for relevance. **The deadline for Part 1 is October 28th and no late days are allowed on this part (without prior exception or extreme circumstances)** In Part 2, you will use this dataset to build and evaluate your chatbot. Part 2 is due November 11th.

1. Learn how to apply IR to a non-traditional IR task
2. Develop skills in working with messy text data
3. Learn how to annotate relevance data, when working with a large dataset
4. Become more familiar with different ranking and retrieval methods for specific IR applications
5. Develop skills at annotating and manually ranking for IR
6. Improve software development skills
7. Practice working with larger datasets

[1] To interact with one good example of a web-based chatbot, see `https://www.pandorabots.com/mitsuku/`.

Conversation Data We have provided you with a large corpus of 2,326,394 conversation pairs from Reddit. Here, each message m has one or more replies to it, $r_i, \ldots, r_k$. Both messages and replies have unique identifies that you will use later. Your job will be to use the message data in this dataset to match with the user queries and to reply to the user with one of the responses.

Please note that we've done our best to filter this data to remove offensive comments. However, as is the unfortunate nature of social media data, offensive language is often present and missed by automated methods. Please be aware that you *might* see offensive messages or those on sensitive topics, despite our efforts to remove them. If you do see something offensive, let us know and we can get it removed.

Part 1: Rank Chat Replies– Due October 28 (30 points)

In the first part of the assignment, you will help create the ground truth for this dataset. This process is intended to help you learn how to create good ground truth datasets and to understand the challenge of determining relevance when you (as an annotator) cannot possibly rank all documents for relevance.

For Part 1, each student is assigned 10 messages (specified as IDs) from the dataset and asked to submit their relevance judgments for *at least* 50 replies (also specified as IDs). Reply relevance should be scored on the following scale, keeping in mind that relevance for a chatbot:

- **2** – A great response to this message that is salient, self-contained, and clear for use in a chatbot conversation
- **1** – A good response to this message that is on-topic, but potentially not self-contained for use in a chatbot conversation. Users will likely continue the conversation after such a reply
- **0** – A mediocre response that may or may not be on topic, or may be vacuous (e.g,. "ok"). A user may be confused from this response when used in a chatbot conversation or may be less likely to continue the conversation.
- **-1** – An off-topic response to a message in a chatbot conversation, which would likely lead the user to be confused
- **-2** – A toxic, bad, or otherwise off-topic response to a message in a chatbot conversation

Note that many messages in the dataset do not have 50 replies, so you will need to search the dataset for potentially-relevant replies. This may seem like a daunting task, but is a real demonstrating of the challenge in this kind of annotation.

Most responses are not relevant to a message and are likely to be rated a 0 or -1. However, submitting annotations that are entirely 0, -1, or -2 is not particularly useful for IR. For Part 1, you should identify *at least one* response that has a rating of 1 or 2 for each query.

Ideally, there will be multiple quality responses (ratings 1 or 2), so you should go looking for these. There is no right way to evaluate; however, we do offer a few suggestions for making the task easier:

- Consider loading a subset of the data into a Google sheet and using search capabilities to find responses that have relevant words

- Find a good response in one of the immediate responses and use some text similarity tools to identify similar replies to annotate (not all will be potentially relevant)
- Use command line tools like `grep` to filter down subsets of the data
- Load a subset of the data into a Pandas data frame (or all of it if you can!) and try identifying candidates to annotate programmatically based on attributes like keywords, textual similarity, length, language model similarity, etc.

We emphasize that this step can feel challenging because there is a *lot* of data. However, the process you'll go through is realistic in scope to what you might face in industry (or academia!) as you try to evaluate internal systems.

Important Motivating Factor: While you might be tempted to skimp on annotations, remember that we will be using these annotations to score your chatbots. Bad annotations will make it *much* harder for your chatbot to get a good score.

Extra Credit (10 points) As a limited opportunity for extra credit, you can receive one point for each additional message annotated (with at least 50 reply annotations). These messages must be selected from a list linked on Canvas with the homework, where you'll pick up to 10 IDs.[2] These extra credit assignments must be done at the same time as the required assignments and have the same deadline (otherwise we can't use them).

What you provide We will give you a list of message IDs corresponding to the `message_id` column in the data we provide. You should annotate at least 50 responses for each message using the rating scale and then upload a .csv file containing three columns, with names `message_id`, `response_id`, `rating`. The first two columns must match IDs from the data we provide and the rating should be one of the numbers from the rating scale. If you've done additional annotations for the ground truth, please include these in the same file. Your file should be named `firstname_lastname.csv` so we can easily process it.

Part 2: Implement an IR-based Chatbot (70 points)

In Part 2, you will actually build the chatbot! Since building a full chatbot is too complicated for a homework, we've simplified the task to the simplest IR-based setup: the user will type a message m and you will respond with a single statement. We've put together the skeleton code for a web-based front-end for you, so you can actually test it out if you want! Your job will be to implement a chatbot-relevance function based on the dataset that will match the relevance scores produced by the class.

Your implementation should use the dataset provided by the class as its source of replies. You are free to use any form of relevance scoring, including using off-the-shelf IR libraries to compute relevance. However, you cannot use any existing chatbot library. To help with this effort, we will

[2] We've pre-selected these messages to increase diversity in the ground truth data, which is why you need to pick from a list.

release half of queries you annotated with aggregated relevance scores for responses. You can use these to train and evaluate your system.

Part 2 consists of two sub-parts. In Part 2a, you will implement your scoring function so that it can be used with the interactive web framework. This step will let you debug your chatbot in real time to see how well it performs and we recommend doing it first. In Part 2b, we will give you a list of `message_id`s for remaining annotated queries and you will upload your top 10 most relevant responses (with their `response_id`) for each `message_id`. The Kaggle leaderboard will show your system's NDCG@10 on half of this held-out data and your grade will be based on the private leaderboard.

The Kaggle competition for Part 2b will be released shortly after we get all the responses from Part 1, which is why no late days are allowed (otherwise we can't integrate these annotations into Kaggle!). Please be sure to check Piazza.

There are *many* ways to build your relevance ranker for a chatbot. We encourage you to try a few and see how they do. To help get you started, here are a few ideas:

- Index all of the messages and use BM25 to retrieve the most relevant message (according to BM25); then reply with a random response to this message in your dataset.
- Use any of the text similarity methods to find the most similar message to the user's message (e.g., compare tf-idf vectors) and use a random response to that message.
- Build a language model on the responses in the dataset as documents and treat the messages as queries. Use $P(Q|D)$ to score responses according to their likelihood of being generated by that message, then find the response that is the most likely to be generated by the user's message.
- Take a Learn-to-Rank approach that given a message and response, tries to predict the relevance score. You will need to use the class's relevance scores for this and need to figure out how you'll want to encode the message and response.

These ideas can help get you started and hopefully give you some intuition about how you can create a chatbot using many of the IR techniques we have talked about so far. If you have questions on the appropriateness of certain libraries, please let us know on Piazza.

What if I don't do python?

If you program in some other language other than python, you are still welcomed to do this homework in that language. You will still need to implement Part 2a, though you can potentially use Python to call your program to do the ranking, which might be easier via some REST-like interface. We're happy to brainstorm this idea too in office hours. If you have questions on concerns, please reach out to the instructors ASAP.

What to submit?

You need to submit four things, most of which are due November 11th:

1. **By October 28th**, upload your relevance rankings in .csv file format to Canvas
2. Please submit your code in a runnable format to Canvas; .ipynb files are acceptable but .py files are preferred since we need to run it. If you use a language other than python (which is fine), please include a note on how to run your code.
3. Please submit the rankings for Part 2b to Kaggle. The link for Kaggle will be announced on Piazza.
4. Please submit a text (pdf/Word) file that describes your choices and implementation for Part 2. We will need to understand this to make sense of your code. **Be sure to include your Kaggle username in this file** so we can figure out which score is yours.

Everything needs to be submitted to Canvas.

Late Policy

Throughout the semester, you have three free late days total. These are counted as whole days, so 1 minute past deadline result sin 1 late day used (Canvas tracks this so it's easier/fair). However, if you have known issues (interviews, conference, etc.) let us know at least 24 hours in advance and we can work something out. **Special Covid Times™ Policy**: If you are dealing with Big Life Stuff®, let the instructor know and we'll figure out a path forward (family/health should take priority over this course). Once the late days are used up, the homework cannot be submitted for a second, though speak with the instructor if you think this is actually a possibility before actually not submitting.

Academic Honesty Policy

Unless otherwise specified in an assignment all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others must be clearly identified as a quotation, and a proper citation provided. Any violation of the University's policies on Academic and Professional Integrity may result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program. Violations of academic and professional integrity will be reported to Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed.

Please be aware, that we know that many implementations of BM25 and Pivoted Indexing exist out there. We have collected many (many) of these and are able to check your implementation against theirs, as well as against other students. Please do you own work and do not submit code you found online (or anywhere else).