CPSC 599.27 Term Project Report: Reverse Jeopardy!

SHAEMUS MELVIN* and VALERIE KIM*, University of Calgary, Canada

ACM Reference Format:
Shaemus Melvin and Valerie Kim. 2023. CPSC 599.27 Term Project Report: Reverse Jeopardy!. 1, 1 (December 2023), 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In our project, Reverse Jeopardy, we aim to help people create entertaining and challenging Jeopardy games for topics of their own interest. In emulating the game show Jeopardy, the user and model act together as the game show producers and host, planning out the various challenges that contestants must answer correctly. However, Jeopardy has a twist on the typical knowledge game show format. In typical game shows, the hosts presents a question (an interrogative) that contestants must answer (a declarative). In Jeopardy, the host presents a prompt (a declarative), while the contestants must respond (an interrogative). In other words, the Jeopardy host presents a phrase that looks like an answer, while the contestants respond with something that looks like a question.

Since many people have already done research on the topic of answering questions (most famously by IBM's Watson, which specifically responds to Jeopardy prompts), our focus here is to fine-tune a neural network model to create appropriate declarative prompts in response to a user's desired category and intended response. In other words, given a brief conceptual context (the category) and a mystery phrase that contestants must identify, our model should create relevant and coherent declarative prompts that hints at, but does not give away, this mystery phrase.

While we have not examined the empirical base of the following claim, it seems that many people accept Jeopardy's quirks as a largely superficial quality that exists primarily to let it stand apart from other game shows. Watson is largely considered an Open-Domain QA model despite the fact that it must answer in the form of a question. Does this mean that our Reverse Jeopardy model must then, necessarily be a Question Generation (QG) task? In this project, the Jeopardy quirk proves to be troublesome for typical QG approaches.

2 MATERIALS AND METHODS

In the process of investigating the appropriate approach for this project, we ran into challenges in defining project specification vs. other our models, we archived all necessary PyTorch files for model deployment, including weights, tokenizers, configuration, and training logs.

*Both authors contributed equally to this research.

Authors' address: Shaemus Melvin, shaemus.melvin@ucalgary.ca; Valerie Kim, minjae.kim1@ucalgary.ca, University of Calgary, 2500 University Dr SW, Calgary, Alberta, Canada, T2N 1N4.

an entry	"EPITAPHS & TRIBUTES", "2004-12-31", "And away we go", 400, "Jackie Gleason", "Jeopardy!", 4,680
desired input	"COUNTRIES | Germany"
desired target	"In 1936, in defiance of the Treaty of Versailles, this country began remilitarizing the Rhineland"
entry:	

Fig. 1. entry vs. desired input and desired target

2.1 Project Specification

It has been difficult to pin down the task of creating Jeopardy prompts as sitting solely in the realm of Question Answering (QG) nor in Question Answering (QA). The desired qualities of our model's outputs have similarities with both tasks, but our requirements of austere user input and the need for avoiding direct reference to the answer challenges some assumptions underlying the techniques found in both tasks.

Firstly, techniques in Question Generation often tasks in which the users provide involve much larger inputs than our task with Reverse Jeopardy. Rather than generate questions from a passage, knowledge base, image, or conversation, we are working with an austere user input, often as short as five words in total.

On the other hand, despite the seemingly QA interaction of our model providing a declarative statement given a user's interrogative input, they are in reality, implied questions that also require information from the category. Additionally, we have the requirement that our must not give the answer away through direct reference.

The intent and reasoning demanded of contestants by these prompts are as rich and varied as those described in many question taxonomies. [4]

2.2 Model Selection and Fine-Tuning

Our first approach drew inspiration from the Open Domain Question Answering pipeline, which involves a Retriever model designed to retrieve documents with the highest similarity vector to the entities identified in a question. These candidates would then be used as contexts for a Reader model that would identify the best answer for that question amongst those contexts. [3] We initially tried DistilBERT, hoping that an Open Domain QA process would ensure answerability due to its strengths in named entity recognition. Answerability is the criteria of whether the user's provided answer could directly answer the generated question. However, as these encoder-only models require a context to answer a question, we did not have the ability to augment the data necessary, and we were also wary as to whether existing Open Domain QA models can serve the wide variety of Jeopardy clue types.

The remainder of this report focuses on our alternative approach: fine-tuning the decoder-only DialoGPT and encoder-decoder T5. These two models have much stronger capabilities in text and sequence generation based on prior information, which served as an excellent starting point for iterating on our project specification.

2.2.1 Data Preparation. The jeopardy dataset, scraped from j-archive and available on HuggingFace and Reddit, [1] comprised initially of 216,930 entries. One entry contains: a clue, a response, a category, a dollar amount, air date, round, and show number. We use only the clue and category for user input, and response for target output. Considering an example entry (fig **??**)

After following existing exploratory data analysis, [2], we found that some data cleaning was necessary to make it more appropriate for our text-only approach. First, the data was cleaned by removing audio clues (denoted "(audio clue)"), video clues (denoted "(video clue)"), and clues that included a link to a photo (implemented by removing clues with "a href="). Second, we removed all entries where a quotation mark is present in the clue or category, as these often

Initial Total	216,930
Post-Cleaning Total	127,183
Training	103,017
Validation	12,719
Unused	11,447

Fig. 2. Data Counts

Model	Batch Size	Epochs
T5 (500 entries)	8	15
T5 (50,000 entries)	32	3
T5 (all entries)	32	3
DialoGPT (50,000 entries)	8	3
DialoGPT (all entries)	8	3

Fig. 3. Models vs. Training Hyperparameters

indicate some sort of wordplay (e.g. "colorful" in a clue indicates that a colour is in the response, e.g. "greenland", and a letter in quotation marks in the category indicates the response begins with that letter), which we deemed to be outside the scope of the project.

2.2.2 Fine Tuning. Due to the size of the dataset vs. resource constraints, we could only train the models on a substantial percentage of available training data for 3 epochs. Batch size was 32 in the T5 model, and 4 in the DialoGPT model.

The setup for training the DialoGPT model was taken mostly from the starter code given for the assignment 4 chatbot generator, with some minor modifications to the input and output. The primary modification to the output is that, contrary to typical chatbot structure, the overall conversation is not considered when generating output, i.e. the only tokens that are used to generate output are those given in the most recent user "utterance". This is done since the interaction the user is having with the bot isn't so much of a "dialogue" or "chat" as much as it is trying to get a single response. In addition to running contrary to the purpose of the model, the training data also isn't conducive to a multi-turn dialogue, as clues (with rare exception) are independent, with clues generally not sharing a common theme that isn't denoted by the category.

The models were trained on Google Colab, which proved to be a fairly significant issue due to the fact that free GPU access on Google Drive is limited weekly. With over 100,000 entries over the course of data exploration and experimenting with various models, a single epoch lasted as long as 80 minutes, exhausting Colab resource limits in the meanwhile.

3 RESULTS

Likely due in large part to the models being trained based on pre-trained models and being given exorbitant amounts of data, the prompts generated by both models were generally well-formed grammatically speaking. There were no instances of incomprehensible prompts that did not make grammatical sense. The actual quality of questions was less so.

Since the models are generative, testing involves some degree of subjectivity. Particularly since the models are generating trivia questions, regular generative evaluation must be performed checking for things like grammatical correctness, but particular to the Reverse Jeopardy models are factuality, and answerability. Factuality is a model's

ability to determine and output correct information about the response, and is not necessarily related to answerability. Answerability is a model's ability to generate a prompt from which a human is able to deduce the answer. While the two are often correlated, they are not necessarily the same thing.

Some things should be noted about both of these metrics: First, it is possible for a prompt to be partially factual - if a high school student were prompted to write about the collapse of the Soviet Union and wrote that *"In 1991, the Berlin Wall fell and the USSR collapsed", there are certainly problems with what is written, but it wouldn't be correct to say that what the student wrote was entirely incorrect. For this reason, factuality is rated on a scale from 1-5, with 5 being completely factual ("Canada/This country's capital is Ottawa"), and 1 being completely incorrect and/or nonsensical ("Canada/This man sailed the ocean blue in 1492"). Likewise, answerability is also graded on a scale, and is not necessarily related entirely to factuality.

Some questions become much more answerable if the person responding considers the prompt-giver to be fallible - many Jeopardy! prompts don't actually entirely contain information known to the contestants, and there are often "multiple ways" to get to the response; an example would be for the response "Spain" for the prompt "This birthplace of Picasso is the only European country with a land border in Africa" - if one knows that Spain has an overseas territory in North Africa, or knows the birthplace of Picasso, they can get the response. This means that a contestant doesn't have to know both, only one - an additional important note is that if a contestant doesn't know one of the facts, it may as well be made up (as a language generation model might do sometimes) - this means that answerability and factuality aren't necessarily linked, as as long as a prompt is partially true, it may be possible to answer it (one could also say "The birthplace of Picasso and one of only two nations that border Russia", and the Picasso hint would still give it away).

There are also certain topics in Jeopardy! which are referred to as "Pavlovs", a reference to Pavlov's Dogs. A Pavlov is a topic which, when the prompt mentions a certain related topic, the answer is always the same (or it is often enough that one can confidently assert the response). An example would be if a prompt mentions a "Austrian Psychologist", the response is almost always Sigmund Freud. This factor pretty much completely divorces factuality from answerability, since (assuming the responder can assume fallibility), the prompt could read "This Austrian psychologist defeated the Turkish forces in the Battle of Crimea in 1094" and "Austrian psychologist" would still tip off that the response is Freud. Since responding to certain prompts like these requires knowledge that the prompt generated may contain incorrect information, answerability is also rated on a scale of 1-5.

Both models created were used to generate a list of 30 prompts, with categories and responses being picked by Shaemus in fields that he knows well enough to identify factuality and answerability. The prompts generated were then rated based on the metrics listed above. The data is presented alongside this report, containing the category, response, prompt, model, factuality (hand-judged), answerability (also hand-judged), whether the response was given in the prompt (the prompt given for "Star Wars" is '"Star Wars" is the first film in this 1977 series'), and any additional notes to explain answerability/factuality scortes. The important part of the data is that on average, the dialoGPT model generated questions with a factuality average of 2.44 and answerability 2.24, meaning that the questions on average weren't great. The T5 model generated prompts with a factuality average of 2.48, and answerability average of 2.44. This is still not great (answerability of 2.44 is just below what you would expect for 50 percent answerability), but there is interestingly a statistically significant difference between the average answerability of the two models, at about 10 percent.

Looking at the various topics/responses given, both models gave the same scores for the following prompts:

- Lithuania (5/5)

- Uruguay (5/5)
- Liberia (1/1)
- Justin Trudeau (2/1)
- Shinzo Abe (2/1)
- Citizen Kane (1/1)
- Gone With the Wind (1/1)
- The Godfather (1/1)

Generally, the models both performed well in Countries, and badly in At the Movies - the DialoGPT model generated two out of three of its 5/5s in Countries, and three out of four 5 answerability scores, with another answerability score of 4. The only other answerability scores above 3 were Boris Yeltsin, Genghis Khan, and Star Wars. Likewise, the T5 model generated 2 out of 4 of its' 5/5s in Countries, with the other two being Boris Yeltsin (again), and French. Answerability scores above 3 were Genghis Khan, and Leif Erikson. An interesting note is that while Lithuania and Uruguay generated good prompts in both models, Liberia resulted in unanswerable prompts with hallucinated facts, while the DialoGPT model produced fairly answerable prompts for both Germany and Ireland (with the national anthem being incorrect for Germany and Ireland's largest city being misidentified).

Languages seemed to be a mixed bag - the 7 out of 10 possible metrics were evaluated at a 1 in the DialoGPT model, with 4 out of 10 in the T5 model. DialoGPT came very close to generating a good prompt for "Dutch", identifying the language as that of the Netherlands, however the prompt asked for "this country", throwing off the answer that one would give (for what it's worth I would likely answer "Suriname", the only other country where "the language of the Netherlands" is spoken). Last note on Languages, Esperanto was misidentified by both models, with DialoGPT calling it a "Language of the Aztecs" and T5 calling it a "Spanish language" - this is of note because one might expect these results from a response that had never occurred in Jeopardy history, but there have been multiple instances where Esperanto was a correct response over the show's history (even entire categories!).

History also resulted in hit-or-miss prompts, with both models identifying Pearl Harbor as an important battle to the United States that took place in 1941, but neither correctly identifying the date (December 7th), or that it was an attack by the Japanese on the United States. Julius Caesar was twice misidentified as a Roman Emperor, which is a common misconception, but technically he was never emperor - what likely caused this hallucination was the presence of the word "Caesar" in the response, a word which later came to mean emperor; if the response was instead "Augustus Caesar" the prompt would have maybe been more accurate. Genghis Khan was pavlov'd in the DialoGPT model with "Mongol Leader", other information being dubious to incorrect, and the T5 model generated an almost entirely correct prompt, with even the date of his conquest being correct - however it identified Genghis as "The Qing Dynasty", which would not exist for several more centuries. Jefferson Davis, the president of the Confederate States of America, was vaguely declared a "President" by both models, with the DialoGPT model actually generating a mostly factually correct prompt (though the description of Davis as a "President" without specifying the country made the prompt unanswerable). Leif Erikson was identified twice as Danish, which is close, but he would be more accurately described as Icelandic, Norse, or possibly even Norwegian. His notability was misidentified in the DialoGPT model (calling him an astronomer - note also that "Danish Astronomer" is a Pavlov for Tycho Brahe), but in the T5 model he was identified as "The first European to cross the Arctic Ocean" which is technically true, and "First European to cross [the ocean]" is generally what he is known for.

4 DISCUSSION AND CONCLUSION

The evaluation of the output data shows that (at least out of the 60 questions evaluated), neither model performs extremely well as a general question generator. For all but a few prompts, the only way to solve the prompts generated is to have the knowledge that the prompt may contain incorrect information, and to know the specific way that it "expects" you to find the response. This is not necessarily an issue with the design of either model itself - the fact that both are indeed capable of generating well-formed trivia prompts shows that the models aren't fundamentally broken, they simply need to be more fine-tuned.

The most obvious limitation of our method of interpreting data is the inherently subjective nature of manually grading the quality of prompts; obviously prompts like "The capital of this Baltic country is Vilnius" are good prompts, but grading the prompt "In 1997 this former Soviet president was elected president of Russia" for Boris Yeltsin is difficult, as the threshold for incorrect information has to exist somewhere while also being able to have balance with recognizing that "Russian president in the 1990s" clearly refers to Yeltsin. This method in addition to being extremely subjective and hand-wavey is very time-consuming - not only must you hand-pick categories and responses, but also grade the prompts, which can take large amounts of time depending on the number of prompts generated.

Another limitation is the scope of what is able to be tested; because categories and responses were hand-picked, there is a wide body of knowledge that the model potentially performs very well on, but may be unknown to the users, since it's either something they didn't think about or something they're not knowledgable on. This could potentially be remedied by extracting prompts and categories automatically using RNG rather than individually selecting prompts/categories. This was experimented with during the training stage, but it was never used to evaluate the models themselves.

The clearest next step from here is to determine and/or build an automated evaluation method - this was something that was touched on in the mid-point report, however due to time constraints manual evaluation was the best option. Following that, quality prompts (which I would consider to have a factuality of 5 and an answerability of at least 3) could potentially be used to augment the dataset.

REFERENCES

[1] [n. d.]. https://redd.it/1uyd0t.

[2] M. Houser. 2022. Jeopardy Data Science Project. https://github.com/mhouser2/Jeopardy-Project. Accessed: 2023-04-17.

[3] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. arXiv:2004.04906 [cs.CL]

[4] Prachi Gharpure Nikahat Mulla. 2023. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence* 12 (2023), 1–32. https://doi-org.ezproxy.lib.ucalgary.ca/10.1007/s13748-023-00295-9