ECE 1786 Creative Applications of Natural Language Processing Final Report

Project: Political Jokes

Gan Yang 1000909163 Yimin Chen 1005138799

> Word Count: 1888 Penalty: 0%

1. Introduction

Our innovative project, PoliticalJokes, utilizes the powerful GPT-4 Turbo technology to generate politically-themed jokes. Notably, it features an advanced joke rating system. By inputting a series of keywords, users can expect to receive a personalized joke that incorporates those words, accompanied by a corresponding funniness rating. Remarkably, even with the same set of keywords, the system produces a unique joke with each run. Our primary objective is to bring happiness and amusement to individuals, while also educating them on the historical context of the oppressive regime in the USSR.

2. Illustration



Figure 1. Architecture

3. Background

Previous research has investigated humor generation and evaluation using GPT, with authors in [1] comparing different models, including GPT-3, for generating Chinese crosstalk. Additionally, Mittal et al. [2] and Tian et al. [3] employed GPT-3 and GPT-2 respectively to generate puns, while Witscript 3 [4] utilized GPT-3 for inserting jokes into conversations. However, none of these studies utilized GPT-4 for joke generation, nor did they explore political jokes. Therefore, our project aims to be the first to explore the potential of GPT-4 in generating political jokes. In terms of joke evaluation, Goes et al. [5] introduced a method using GPT-3 as 4 Al judges to assess joke funniness. It was found that these Al judges rated jokes higher than human judges, albeit with similar trends. Further study by Goes et al. [6] continued this work using GPT-4, and successfully achieved a positive correlation between Al score and human score in evaluating

joke funniness. In our project, we differ from previous research as we will evaluate political jokes generated by our own generator.

4. Data and Data Processing

4.1 Raw Data Collection

We collected 141 Q&A format Russian jokes from [7]. After manually removing duplicates and non-political jokes, there are 92 political jokes. We also collected 25 Q&A format facts from [8] as a control group. All these data are converted to a uniform format (Q: ... A: ...) and put into csv files with column name "Text".

4.2 Manual Labeling

We manually read through each of them and provided an integer funniness rating from 1 to 5 for each joke. Since funniness is subjective, we both provided ratings individually and then averaged and rounded our ratings for each joke. The non-joke texts were rated with funniness 0. After some initial experiments, we found that the performance for GPT-4 evaluator is not good enough for 0 - 5 ratings, so we decided to rescale the ratings to 0 - 3. We did not do manual labeling again, but multiplied 0.6 on our 0 - 5 ratings. All raw, averaged and scaled ratings are shown in columns "Gan's Rating", "Yimin's Rating", "Rating" and "Rating out of 3" in the csv files.

4.3 Data Sorting and Splitting

We used OpenAI's "tiktoken" package to calculate the number of tokens of each text and added a new column "Number of Tokens". All data were sorted by "Rating out of 3" and "Number of Tokens" in a descending order. Then sorted data were splitted equally into generator set (even indices) and evaluator set (odd indices).

4.4 Data Example

Text	Number of Tokens	Rating out of 3	Rating	Gan's Rating	Yimin's Rating
Q: What is chaos? A: We do not comment on national economics.	16	2	4	3	4

Table 1. Data example

4.5 Issues in Data

The distribution of the data within each rating is skewed. We have 10 data points with rating 3, 69 data points with rating 2, 13 data points with rating 1 and 25 data points with rating 0.

5. Architecture and Software

As shown in Figure 1, the flow starts with a series of keywords input from users, which will prompt the generator to create a joke based on these words. Then the generated joke will be sent to the evaluator for funniness rating. Finally, both the joke and the rating will be output to users.

We applied prompt engineering for both the generator and the evaluator. The prompt for the generator (Appendix I) contained 3 parts: the instruction, examples, and the keywords for those examples. The prompt for the evaluator (Appendix II) also had 3 parts: the instruction, examples, and the ratings for those examples. The instructions were carefully designed and iteratively improved for the best performance. For the generator, we selected 46 Russian political jokes, manually picked the keywords for each political joke, and fed all 46 keyword-joke pairs to the generator as examples. Similarly, for the evaluator we gave 4 Russian political jokes as examples, and for each example we paired it with a rating out of 3.

6. Comparison

The metric employed in this study assigns a numerical rating ranging from 0 to 3 to evaluate the level of humor in Q&A format Russian political jokes. A score of 0 indicates that the text lacks humorous elements, while a score of 3 indicates high comedic value. Both the evaluator and human judges will utilize this metric to assess the quality of jokes generated by the generator. Additionally, the evaluator's performance can be evaluated by comparing their ratings with those given by human judges through accuracy and root-mean-squared-error (RMSE) metrics.

7. Quantitative Results

7.1. Test Based on the Dataset

To assess the performance of the evaluator in development stage, we conducted tests using a dataset comprising 30 data points generated from our collected data. The testset consisted of a total of 30 instances, comprising 4 jokes rated with a score of 3, 16 jokes rated with a score of 2, 5 jokes rated with a score of 1, and 5 non-jokes rated with a score of 0. The ground truth were averaged and rounded ratings from both team members.

rating	precision	recall	f1-score	support
0	1	1	1	5
1	0.33	0.2	0.25	5
2	0.7	0.88	0.78	16

3	1	0.5	0.67	4
accuracy			0.73	30
macro avg	0.76	0.64	0.67	30
weighted avg	0.73	0.73	0.71	30

Table 2. Classification report for the last test on dataset

The initial evaluation results displayed suboptimal performance (accuracy: 0.47, RMSE: 0.73), despite the implementation of formatted prompts and few-shot examples. The subsequent incorporation of chain-of-thought reasoning led to an improvement in accuracy to 0.57, and a decrease in RMSE to 0.66. Subsequently, by further refining the prompts, we managed to attain an accuracy of 0.73 and an RMSE of 0.52. Following this, the final classification report, presented in Table 2, demonstrated that the evaluator was capable of accurately distinguishing between jokes and non-jokes, achieving a 100% success rate. Although the accuracy may not be exceedingly high, it is noteworthy that the RMSE remains relatively low, indicating that the discrepancies primarily fall within a range of one rating. As an illustration, a joke rated at 3 by a human would most likely receive a rating of either 3 or 2 from the evaluator.

7.2 Test with Generator and Evaluator

Once the generator and evaluator were prepared, a test was conducted using six different sets of keywords. For each keyword set, five jokes were generated by the generator and subsequently evaluated by both the evaluator and human judges. The results of the generator's performance can be found in Table 3, which reveals that there were no instances of low-quality jokes with ratings of 1 or 0 in thirty iterations. Additionally, according to human ratings, the jokes generated by the generator were considered to be funnier than those produced by GPT. In terms of the evaluator, it achieved an accuracy of 0.57 and an RMSE of 0.66 when compared to human ratings for the thirty jokes.

Rating	Count in Human Rating	Count in Evaluator Rating
3	19	10
2	11	20
1	0	0
0	0	0

Table 3. Count of generated jokes with different ratings

rating	precision	recall	f1-score	support
0	0	0	0	0

1	0	0	0	0
2	0.45	0.82	0.58	11
3	0.8	0.42	0.55	19
accuracy			0.57	30
macro avg	0.31	0.31	0.28	30
weighted avg	0.67	0.57	0.56	30

Table 4. Classification report for the test on generated jokes

8. Qualitative Results

8.1 Generator

The generator functions effectively when the keywords incorporated form a part of a political joke, as illustrated in Figures 2 and 3. However, its performance is not always reliable when provided with non-political keywords, as demonstrated in Figures 4 and 5.

keywords	
Stalin	
Clear	Submit

Figure 2. Political-related keyword example

Generat	zed joke:
Q: What	's Stalin's idea of a balanced diet? A: When every Soviet citizen has a ration card in both hands.
Rating:	3/3
Explana	ition: The joke plays on the historical context of rationing in the Soviet Union, highlighting the disparity and
hortag	es in food supply with a sarcastic notion of balance. The humor comes from the irony of considering a
hortag	es in food supply with a sarcastic notion of balance. The humor comes from the irony of considering a

Figure 3. Generated joke, rating and explanation for keyword "Stalin"

keywords	
grammar, childcare	
Clear	Submit

Figure 4. Non-political-related keyword example

Generated jo	oke:
Q: Why is gra	ammar important in childcare? A: Because you don't want your child growing up saying "I has" instead o
"I have" like	the rest of us.
Rating: 0 / 3	
Explanation:	This joke plays on the common concern for proper language development in children while making a
self-depreca	ting commentary on the current state of grammar skills in the population. However, the political
dimension is	s missing, and it doesn't tie in with the Russian political context as expected for the evaluator role. Its
humor migh	t be appreciated in a general context, but as a Russian political joke, it lacks relevance

Figure 5. Generated joke, rating and explanation for keywords "grammar, childcare"

8.2 Evaluator

The explanations provided are coherent and comprehensive, effectively conveying the intended meaning. However, the ratings assigned to the jokes are regarded as subjective. The evaluators possess the ability to discern between non-jokes and assign them a rating of 0. When evaluating political jokes, the ratings may not always align precisely with those determined by human judges, but they never deviate by more than one rating. This discrepancy can be attributed to the utilization of manually labeled data in the provided examples and the inherent subjectivity and potential bias resulting from only two human judges. Figure 2 and 3 illustrate an instance where the human and evaluator ratings match, while Figure 6 and 7 exemplify a scenario where the evaluator rating (3) surpasses the human rating (2).

keywords	
USSR	
Clear	Submit

Figure 6. Evaluator rating higher than human rating example keyword

Genera	ted joke:
Q: What	is the shortest horror story in the USSR? A: "Comrade, we need to have a talk at the Party office."
Rating:	3/3
0	
Explana	tion: This joke plays on the fear and paranoia of being summoned by the Communist Party for questioning
punishı	nent, a reality that could have dire consequences during the Soviet era. The brevity of the horror story
effectiv	ely conveys the terror such a statement could invoke in a citizen of the USSR

Figure 7. Evaluator rating higher than human rating example output

9. Discussion and Learnings

In general, the quality of the jokes generated in our study is substantiated by the findings presented in Sections 7 and 8. Furthermore, the explanations provided by the evaluator demonstrate a logical and coherent assessment. However, a few unexpected observations have emerged. Firstly, when non-political-related keywords are used as inputs, the generator tends to produce non-jokes, as observed in Section 8.1. Additionally, repetitive use of the same set of keywords sometimes results in generated jokes that exhibit strong similarities, albeit with slight variations in wording. Furthermore, the evaluator's performance is comparatively less accurate when evaluating generated jokes in contrast to evaluating jokes from the dataset. This discrepancy may arise from the fact that the generator did not generate non-jokes, whereas the evaluator has an impeccable accuracy rate when it comes to rating jokes as 0.

If future projects similar to this one are undertaken, several modifications should be implemented. Firstly, for subjective metrics such as human ratings, it is recommended to gather data from a larger pool of individuals rather than solely relying on the opinions of the project team. By averaging the perspectives of multiple individuals, the subjective ground truth is less likely to be influenced by bias. Additionally, this approach would address the issue related to rounding up the average opinions of just two raters. Secondly, it is advised to expand the dataset by collecting more data, ensuring an adequate number of data points for each rating. Subsequently, excess data points should be discarded to prevent dataset skewness. Thirdly, introducing statefulness into the generator can facilitate the production of unique outputs, e.g. integrating previously generated jokes that share the same keyword, originating from the same user, into the generator prompts. Finally, it is advisable to fine-tune GPT models. Utilizing additional examples during the fine-tuning process would yield better results, faster response times and less token count sent to GPT with each iteration following fine-tuning.

10. Individual Contributions

Gan

- Data collection and labeling
 - Create a dataset with 141 jokes from [7]
 - Rate the funniness of the jokes
 - Integrate political jokes and facts, count tokens, sort data and divide them into generator set and evaluator set
- Political joke evaluator development
 - Write code to call GPT-4 API to evaluate political jokes
 - Evaluate results, tune parameters and do prompt engineering
 - Wrap the code into classes for easier integration
 - Collect quantitative and qualitative results
- Gradio UI
 - Write code to call the generator and the evaluator
 - Gather the user input, display the model output
- README file of our Github repo
 - Add UI examples and minor editing
- Final test with generator and evaluator combined
 - Run the whole system 30 times with 6 sets of keywords
 - Collect quantitative and qualitative results

Yimin

- Data collection and labeling
 - Remove duplicates from above and separate non-political jokes
 - Rate the funniness of the jokes
 - Create a dataset with 25 facts from [8]
- Political joke generator development
 - Write code to call GPT-4 API to generate political jokes
 - Evaluate results, tune parameters and do prompt engineering
 - Wrap the code into classes for easier integration
 - Collect qualitative results
- README file of our Github repo
 - Write the first draft

References

[1] B. Wang, X. Wu, X. Liu, J. Li, P. Tiwari, and Q. Xie, "Can Language Models Make Fun? A Case Study in Chinese Comical Crosstalk", 36th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks, 2 Jul 2022.

[2] A. Mittal, Y. Tian, and N. Peng, "AMBIPUN: Generating Puns with Ambiguous Context", Vol: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Jul 2022.

[3] Y. Tian, D. Sheth, and N. Peng, "A Unified Framework for Pun Generation with Humor Principles", Vol: Findings of the Association for Computational Linguistics: EMNLP 2022, Dec 2022.

[4] J. Toplyn, "Witscript 3: A Hybrid AI System for Improvising Jokes in a Conversation", arXiv:2301.02695, Computation and Language; Artificial Intelligence, 6 Jan 2023.

[5] F. Góes, Z. Zhou, P. Sawicki, M. Grzes, and D. G. Brown, "CROWD SCORE: A METHOD FOR THE EVALUATION OF JOKES USING LARGE LANGUAGE MODEL AI VOTERS AS JUDGES", arXiv:2212.11214, Artificial Intelligence, 21 Dec 2022.

[6] F. Goes, P. Sawicki, M. Grzes, D. Brown, and M. Volpe, "Is GPT-4 Good Enough to Evaluate Jokes?", Subject: Q Science > Q Science (General) > Q335 Artificial intelligence, 07 Jun 2023.

[7] Armeniapedia, "Radio Yerevan Jokes", armeniapedia.org. <u>https://www.armeniapedia.org/wiki/Radio Yerevan Jokes</u> (accessed Nov 18th, 2023)

[8] P. Baudis, J. Pichl, "WebQuestions QA Benchmarking Dataset", github.com. <u>https://github.com/brmson/dataset-factoid-webquestions</u> (accessed Nov 18th, 2023)

Appendix

I. Generator Prompt

instruction

Generate a USSR political joke with the keywords given. Pretend it's the twentieth century and the USSR still exists. Use dark humor. Base on the history and don't play on words. Generate the output in one line.

examples

Keywords: Nixon, Kremlin

Q: When Nixon visited Moscow, he and Khrushchev had a race around the Kremlin. Nixon came the first. How should our media report on that? A: As follows: In the international running competition the General Secretary of the Communist Party took the honorable second place, while President Nixon came in next to last.

Keywords: labor camps

Q: Is it true that conditions in our labor camps are excellent? A: In principle, yes. Five years ago one of our listeners was not convinced of this, so he was sent to investigate. He seems to have liked it so much that he hasn't returned yet.

Keywords: Soviet technology

Q: Is it true that due to the achievements of Soviet technology, we will soon be able to order food over the phone? A: Yes, our sources tell us it is true. The same sources told us the ordered food would be delivered via TV.

... (42 hidden examples)

Keywords: USSR

Q: Is there anti-Semitism in the USSR? A: We must answer in the positive: No!

II. Evaluator Prompt

instruction

Act as a Russian political joke evaluator. Evaluate the funniness with a reason and give an integer rating from 0 to 3, in a format of reason -> rating.

examples

Example #1

<user>: "'Q: Is it true that every Soviet soldier dreams of becoming a general? A: No, our soldiers are not that stupid. They know that generals may become MIA even in time of peace."
<assistant>: "'This joke is very funny because it is easy to understand and plays on the absurdities and contradictions of the Soviet regime and its leadership -> 3"'
Example #2

<user>: "'Q: Could an atomic bomb destroy our beloved town, Yerevan, with its splendid buildings and beautiful gardens? A: In principle, yes. But Moscow is by far a more beautiful city."'

<assistant>: "This joke can be rated between 3 and 1 -> 2"

Example #3

<user>: "'Q: Is there a difference between capitalism and communism? A: In principle, yes. In capitalism, man exploits man. In communism, it's the reverse."

<assistant>: "'This joke is too opaque for immediate comedic impact or is just a simple wordplay without meaningful satire against the absurdities under Soviet regime -> 1""

Example #4

<user>: "'Q: What two countries invaded poland in the beginning of ww2? A: Germany and Soviet Union."'

<assistant>: "This is a fact, not a joke -> 0""

Permissions

Team Member	Post Video?	Post FinalReport?	Post SourceCode?
Gan Yang	Wait till see	Yes	Yes
Yimin Chen	Wait till see	Yes	Yes