DOES THE GOLDEN RULE APPLY TO LLMS: ANALYZING THE EFFECT OF POLITENESS WHEN PROMPTING LLMS

John Bernardin, Amogh Gynaeshwar, Bain McHale
11-711 Advanced Natural Language Processing
Carnegie Mellon University
Pittsburgh, PA
{johnbern, agyanesh, dmchale}@andrew.cmu.edu

ABSTRACT

We propose a robust analysis of how politeness of prompts affects the performance of LLMs. In the past, papers such as "Should We Respect LLMs?" introduce this problem, but we expand the techniques beyond simple templates by introducing a methodology to evaluate the effect of politeness on arbitrary datasets using politeness classification networks. In addition to showing performance metrics, we attempt to explain why this behavior emerges. Papers such as "Demystifying Prompts in Language Models via Perplexity Estimation" may suggest that certain levels of politeness may simply have lower perplexities. We show that perplexity does not have any correlation to output, but politeness does. This is the largest scale anlaysis of how politeness in English affects the performance of LLMs, concretely concluding that being polite does in fact yield better results for most LLMs. These results inform how individuals moving forward can engage with language models more effectively.

***K*eywords** Natural Lanaguage Processing · Prompt Engineering · Politeness Classification

1 Introduction

Just a few years ago, language models could only be interacted with by those with the technical skills to set up and use instances of them. Now, simple chat tools are ubiquitous, with ChatGPT alone reaching 200 million weekly active users[1]. As these generative AI tools become increasingly integrated into daily tasks, the importance of prompt engineering has emerged as a critical area of study. Prompt engineering refers to the art and science of crafting precise inputs to guide AI models toward generating desired outputs. As these AI systems become more sophisticated, the quality of prompts directly influences the effectiveness and relevance of their response. It has been shown that some styles of prompts will consistently yield better or worse results [10]. This makes understanding how to create good prompts essential for maximizing the utility of AI in various applications, from customer service to complex problem-solving.

An emerging theme embedded in articles offering tips and tricks for prompt engineering is the idea of politeness [4]. Some speculation suggests that polite prompts can improve the interaction between users and AI by fostering clearer communication and intent [14]. Alternatively, some believe this behavior could emerge from the RLHF training loop conditioning LLMs to behave politely [7]. As AI continues to evolve, understanding how factors like politeness affect model performance will be vital for optimizing human-AI collaboration.

2 Background

The core idea for our research was sparked by the paper "Should We Respect LLMs?", which addressed this topic by systematically evaluating LLM performance in a variety of languages at different politeness levels [19]. The study finds that impolite prompts often lead to poor performance, while overly polite prompts do not necessarily improve results. The optimal level of politeness varies depending on the language, indicating that LLMs are influenced by cultural norms

in communication. This cross-lingual analysis highlights the importance of considering politeness when designing prompts for different languages and cultures, as it can significantly affect model outcomes and user experience.

2.1 Politeness Classification

The topic of quantifying politeness in natural language extends before large language models. Since 2013, classification models on manually annotated datasets have been created [5]. In "A Computational Approach to Politeness with Application to Social Factors", the authors used support vector machines to develop a system that operated on detected lexical features which may be as simple as using words like "please". Language modeling was able to catapult this work forward using fine-tuned encoder-only BERT based models and zero shot prompting of decoder-only LLMs [18], achieving superior results to the support vector machines. As this arms race continued, networks such as the graph-induced transformer network (GiTN) used even larger networks, stacking LSTM layers and a Graph CNN on top of a fine-tuned BERT implementation [6]. Datasets for this field were also created for specialized tasks, such as goal-oriented conversations and mental health counseling [15][13]. These datasets are typically manually annotated by humans. They may have larger variance than other datasets, as politeness is, to some extent, subject and circumstantial.

2.2 Prompt Engineering and Sensitivity

Studies have shown that well-designed prompts can significantly enhance model performance, even in few-shot or zero-shot learning scenarios, by providing the necessary context and structure for the model to interpret tasks effectively [20][3]. Several techniques have emerged within prompt engineering to improve LLM outputs. For example, chain-of-thought (CoT) prompting encourages models to break down complex tasks into intermediate reasoning steps, which has been shown to improve performance on logical reasoning and problem-solving tasks [20][2]. These techniques highlight how prompt engineering can drastically influence model behavior. Moreover, prompt engineering has practical implications for user experience and resource efficiency. By reducing ambiguity and focusing the model's attention on specific tasks, well-crafted prompts can minimize unnecessary computation and improve response times, making AI systems more scalable and cost-effective [3]. As LLMs continue to evolve, mastering prompt engineering will be essential for maximizing their potential across diverse applications. The grammar structure and word choice also impacts performance, as referenced earlier [10]. An interesting development in prompt engineering is the idea that perplexity has a correlation to performance [8]. It appears that lower perplexity outputs tend to lead to better performance.

Additionally, Zhou et al. investigated how expressions of certainty, uncertainty, and evidentiality affect language model behavior[21]. They found that LLMs are highly sensitive to epistemic markers in prompts, with accuracies varying by more than 80% depending on the phrasing used. Surprisingly, expressions of high certainty resulted in decreased accuracy compared to expressions of low certainty. The authors developed a typology of epistemic markers and injected 50 markers into prompts for question answering tasks to evaluate their impact. Specifically in terms of politeness this is valuable, as different cultures and languages can be more or less sensitive to polite language [9].

2.3 Politeness Generation

Politeness generation is an emerging task in natural language processing that focuses on automatically creating more polite or impolite variations of text, particularly questions or prompts. This task has gained attention due to its potential applications in improving human-computer interactions and understanding the impact of linguistic politeness on language model performance. The task of politeness generation stems from broader research in computational politeness, which aims to analyze, detect, and generate polite language using computational methods[16]. There are three specific ways we can perform politeness generation. First, we can perform a style transfer, which involves transforming non-polite sentences into polite ones, often using encoder-decoder architectures or fine-tuned language models[11]. Next, we can do controlled-text generation, which involves using politeness as a control variable in language generation tasks[12]. Finally and most popularly, we can simply use prompt engineering, where we design templates that induce politeness by incorporating politeness markers into prompts[17].

3 Problem Definition

We are proposing a more robust method of evaluating the effect that politeness has on LLM performance. The original "Should We Respect LLMs?" paper used a set of predesigned templates[19], wherein each one was a different level of "polite". We propose using existing Q&A datasets where we will quantify the politeness of each prompt, then determine the performance of LLMs across prompts. Additionally, we will evaluate the perplexity of prompts in an attempt to draw a correlation between perplexity and politeness, potentially explaining why polite prompts may outperform neutral

I have a sentence, and I want to generate {count} variations of it. Each variation should retain the original tone, meaning, and intent but use different phrasing or structure. For example:
Input: 'What is the best way to learn programming?'
Ouptut: ['How can I effectively learn programming?',
'What are the most effective methods to learn programming?']
Now, generate {count} {tone} variations for this sentence:
'{original question}'

Figure 1: The prompt template used for generating polite, impolite, and neutral variations of each question

or impolite prompts. We utilize politeness generation via prompt engineering to create variations of prompts to analyze. Finally, we will perform prompt based politeness classification in order to determine if LLM's are able to identify polite, impolite, or neutral tone in prompts. Having completed these tasks, we will have designed a framework for comparing the politeness-performance correlation for any arbitrary question/answering task as well as the effectiveness of LLM politeness classification, which does not currently exist in the literature.

4 Datasets

For our dataset, we chose a random subset of 668 answerable questions from the SQuAD 2.0 dataset. We then generated three neutral, polite, and impolite variations of each question using GPT-4o with the prompt in Figure 1. The neutral variations were included to avoid contamination, as GPT-4o has almost certainly been trained on the SQuAD dataset. We also choose three variations as this helps ensure assessment is not biased towards a particular way of asking questions, and this can give us some insight into the sensitivity of the politeness classifier. Here is an example entry from our dataset:

original question: What is the nickname of the historic football rivalry between Yale and Harvard hosted annually in New Haven?

neutral question 1: What is the nickname for the yearly football rivalry between Yale and Harvard held in New Haven?

neutral question 2: How is the annual football rivalry between Yale and Harvard in New Haven commonly referred to?

neutral question 3: What do people call the historic football rivalry between Yale and Harvard that takes place every year in New Haven?

polite question 1: Would you kindly tell me the nickname of the historic football rivalry between Yale and Harvard that is hosted annually in New Haven?

polite question 2: Could you please inform me of the nickname for the historic football rivalry between Yale and Harvard that takes place annually in New Haven?

polite question 3: Might you be able to tell me what the historic football rivalry between Yale and Harvard, held yearly in New Haven, is nicknamed?

impolite question 1: What's the nickname for that Yale-Harvard game? Tell me now.

impolite question 2: What's the damn nickname for that Yale-Harvard game? Spill it.

impolite question 3: Hurry up and tell me the nickname for that stupid Yale-Harvard game.

5 Methods

5.1 Evaluating Question Effectiveness

We evaluate how effective each question variation is by passing the each of the nine questions for each entry through GPT-4o-mini with the following prompt, setting the max tokens to 25:

{"role": "system",

"content": (

"You are a helpful assistant. Please provide a

concise and accurate ""answer to the following question.

Keep your answer under 20 words."), },

{"role": "user", "content": f"Context: {context}\nQuestion: question"},

We then evaluated the resulting answers against the reference answer, using the F1 score, exact match score, and contains score.

5.2 Classifying Politeness

We previously experimented with classifying politeness as a binary (polite or impolite) using fine tuned Bert, Ling, and GPT-3.5 to replicate a baseline paper. This paper was fairly old, so we want to use GPT-4o for this task now. Additionally, we are looking to change how we classify politeness from a binary to a scale from 1-5, with 1 being the most impolite, 5 being the most polite, and 3 being neutral. We decided to do this because some questions may not be inherently polite or impolite, so a binary classifier isn't best suited for this problem. We used the following prompt in GPT-4o with 1 max token to get these results: Please rate the politeness of the following question on a scale from 1 to 5, where: 1 = Very impolite 2 = Impolite 3 = Neutral 4 = Polite 5 = Very polite Only provide the number without any additional text. Question: "{question}"\n\nRating:

5.3 Computing Perplexity

We computed perplexity by passing each question into Davinci-002, as this model supports log probabilities. We then used these log probabilities to calculate the perplexity using the following equation:

$$P = e^{-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i)}$$

We did this to find both the perplexity per sentence and the perplexity per token for each question, then use this to determine if perplexity is correlated with performance and to examine if neutral, polite, or impolite questions have a tendency to be more or less perplex than each other.

6 Results

Our datasets were evaluated on 4 different models: Llama 2-7b-chat-hf, Mistral-7B-Instruct-v0.1, 4o-mini, and Falcon -7b instruct. As different models have different training approaches, this allows us to see if they react differently to politeness.

6.1 Answer Correctness by Question Type

Once we generated the neutral, polite, and impolite variations on each questions, the accuracy of these different question types can be calculated. Accuracy is calculated as the answer from the dataset being contained within the exact answer. Our approach did no additional system prompting or fine-tuning so as maintain the default behavior of LLMs. Because the LLMs tend to give lengthy responses and the chosen dataset does not, we decided to use the metric of determining if the answer is simply exactly contained in the output. Additionally, we find by examining the original question that our concerns about data contamination were not unfounded.

Table 1: Politeness Accuracy (Contains) Scores Across Models

Type	Overall	LLaMA	Falcon	Mistral	4o-mini
Original	0.736961	0.729730	0.627628	0.768421	0.821856
Neutral	0.689319	0.703148	0.550775	0.730635	0.772455
Impolite	0.624156	0.651303	0.444389	0.677323	0.723054
Polite	0.711879	0.722777	0.563872	0.792914	0.767964

To further analyze the effectiveness, we also looked at the F1 scores. An F1 score of 1 shows when the model was able to achieve an exact match. The graphs for F1 scores can be found in Appendix Section 10.1. The results indicate that the Polite scores tend to have a slightly higher average F1 score, but a lower perfect F1 score of 1.

6.2 Question Perplexity vs F1 Score of Answers

We decided to evaluate the correlation between perplexity and performance.

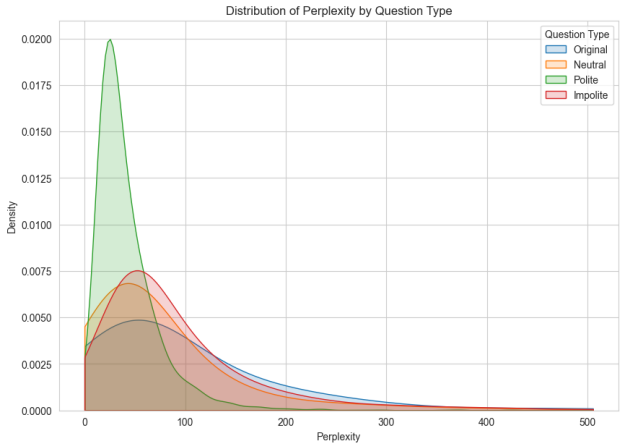


Figure 2: Perplexity per each question politeness.

Here you can see that the polite questions, on average, had much lower perplexity. Thus, we decided to plot this against F1 scores to see if any trend emerged.

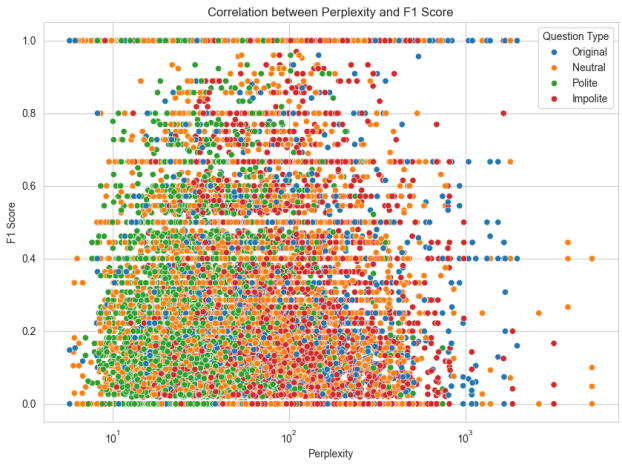


Figure 3: Perplexity vs F1 score accross all models.

In fact, the pearson correlation coefficient for F1 score was -0.0059, showing no correlation. The accuracy pearson correlation coefficient was 0.0004, which also showed no correlation.

We did consider that perplexity per token may be a better metric, as polite questions tend to have for more tokens than others. This yielded pearson correlations of 0.0011 and -0.0061 for the F1 and accuracy scores respectively.

6.3 Consistency Across Questions

We additionally looked at the consistency of different types of outputs for each model. Consistency evaluates whether or not all of the variations align on an answer. For accuracy consistency, this evaluates if all 3 variations were correct, all 3 were incorrect, or the consistency was mixed.

Table 2: Distribution of Question Consistencies

Consistency Level	4o-mini	Mistral	Falcon	LLaMA
All Correct (%)	72.604790	51.646707	39.071856	44.461078
Mixed Consistency (%)	21.856287	35.329341	41.467066	36.826347
All Incorrect (%)	5.538922	13.023952	19.461078	18.712575

A further consistency analysis was done in the appendix. It evaluates consistency across question types and other considerations such as if a model is more likely to be consistent when getting the right answer than the wrong answer.

6.4 Superscoring

We decided to incorporate a superscored version of the accuracy after finding variability in the consistency of performance. This was calculated by counting a type of question as correct if any of the 3 variations asked a version that yielded a correct answer. Given that many systems will generate multiple answers and then evaluate the best answers, we though a superscored version would have interesting implications for a system that generated variations of questions and then had an aggregator function over the outputs.

Table 3: Consistency Categories Across Models and Distributions (Percentages)

Consistency Category	Distribution	Overall	4o-mini	Mistral	Falcon	LLaMA
Consistently Right	Polite	60.778443	71.856287	71.257485	41.167665	58.832335
	Neutral	57.522455	71.407186	62.125749	40.568862	55.988024
	Impolite	50.112275	67.065868	54.491018	31.287425	47.604790
Consistently Wrong	Polite	19.797904	18.862275	13.622754	29.790419	16.916168
	Neutral	20.059880	16.916168	16.766467	29.491018	17.065868
	Impolite	26.047904	22.754491	21.107784	41.167665	19.161677
Mixed Consistency	Polite	19.423653	9.281437	15.119760	29.041916	24.251497
	Neutral	22.417665	11.676647	21.107784	29.940120	26.946108
	Impolite	23.839820	10.179641	24.401198	27.544910	33.233533

6.5 Quantifying Politeness of Questions

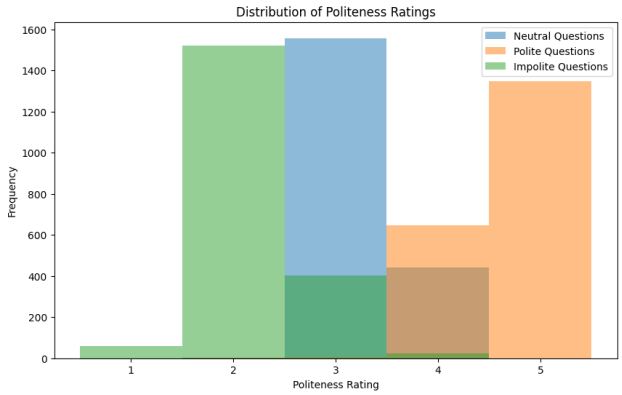


Figure 4: Politeness classification results.

Here we find that GPT-4o mini can pretty accurately identify polite questions. This is likely because polite questions have common markers, such as the word "please". It also frequently identifies neutral questions as 4 - Polite, consistently avoids going higher than 4 or lower than 3 for its ratings, indicating an ok ability to determine neutral questions. However, it doesn't perform particularly well at classifying impolite questions, only classifying a small amount as very impolite, a noticeable amount as neutral, and even a non-trivial amount as polite. This could possibly be attributed to impoliteness being heavily based on tone, which may be difficult for LLMs to pick up on.

7 Discussion

7.1 Politeness Effect on Performance

We find that polite questions consistently outperform their neutral and impolite counterparts for smaller LLMs, but in GPT-4o-mini's case, neutral questions outperform polite and impolite questions. Impolite questions also perform

the worst across the board, indicating that at the very least not being impolite to an LLM will likely improve your performance.

7.2 Data Contamination

We elected to generate neutral variations despite the original questions tending to be quite neutral. In doing so, we found strong evidence of data contamination. The squad dataset is a common benchmark which is duplicated many times on the internet, so there is a good chance the various models were at least pretrained on it. We believe this explains why the original questions, while having a lower politeness, performed better than the polite question set.

7.3 Model Variability

Though generally in smaller models the trend holds that polite questions perform best, all models' performance varies significantly. This can be better seen in section 8.1 of the appendix with the significantly varied F1 score distributions, but even from the previous analysis Falcon consistently under-performs in every category, GPT-4o consistently outperforms, and Mistral generally outperforms Llama.

7.4 Perplexity Effect on Performance

We found no correlation between F1 score and perplexity, indicating that question perplexity does not seem to matter for performance. This means it is likely politeness, or lack thereof, that is impacting performance.

8 Conclusion

We find that politeness does indeed impact the accuracy of your answers. For larger models, being neutral will likely give you better results, but for smaller models, being polite will consistently provide the best results. Additionally, we find that LLMs can accurately identify polite questions and somewhat accurately identify neutral questions, but may struggle with identifying impolite questions without fine-tuning.

9 Future Work

9.1 Politeness Variation Combinations

We believe there is work to be done in taking a question, generating many variations at different politeness levels, then evaluating the different answers generated. It is proved in this work that LLMs can be sensitive to the politeness of the questions, so pre-processing questions through this lens may allow better results. This could be especially effective in scenarios where users are not careful about how they phrase their prompts. For example, a customer service bot where people lodge complaints may regularly receive impolite questions that could be improved with rewording.

9.2 Additional Models

Our analysis only evaluates relatively small models, as we were limited by compute. It may be the case that large-scale models have robustness to politeness and can better interpret the meaning of a question despite how it is worded. As such, further analysis with models more than 10B parameters in size would be effective.

9.3 Other Tasks

We evaluated our model on question answering with a specific dataset. However, the pipeline proposed in this paper is the first to allow someone to dynamically evaluate politeness on any dataset across various different tasks. Extending this work by performing the analysis on different Q&A datasets and tasks beyond Q&A would be insightful.

9.4 Explaining Behavior

While we were able to confirm behavior on how to interface with LLMs, we were unable to provide an explanation for this behavior, as we conclusively proved that perplexity does not show a strong correlation to behavior on this task. We suspect it could be a result of the alignment process. As such, evlauting on raw models that are not aligned against the original models would be interesting.

Acknowledgments

This work is being completed in affiliation with Carnegie Mellon University's Advanced Natural Language Processing course 11-711.

References

[1] Backlinko. Chatgpt / openai statistics: How many people use chatgpt?, 2024. URL https://backlinko.com/chatgpt-stats.

[2] Bensen Chen et al. Unleashing the potential of prompt engineering in large language models: A comprehensive review. *arXiv preprint arXiv:2310.14735v5*, 2024. URL https://arxiv.org/abs/2310.14735v5.

[3] CircleCI. Prompt engineering: A guide to improving llm performance, 2023. URL https://circleci.com/blog/prompt-engineering/.

[4] Idamae Craddock and Kristen Wilson. Six prompt tips: Getting the most out of large language model ai. *School Library Journal*, 69(11):16, Nov. 2023. URL https://link.gale.com/apps/doc/A773080353/AONE?u=cmu_main&sid=googleScholar&xid=06689dff.

[5] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2013. URL https://nlp.stanford.edu/pubs/politeness.pdf.

[6] Tirthankar Dasgupta, Manjira Sinha, and Praveen Chundru Geetha. Graph induced transformer network for detection of politeness and formality in text. In *Proceedings of the 2023 Conference on Human Factors in Computing Systems*, New York, NY, USA, 2023. Association for Computing Machinery. doi: 10.1145/3543873.3587352. URL https://dl.acm.org/doi/pdf/10.1145/3543873.3587352.

[7] Lance Eliot. Hard evidence that please and thank you in prompt engineering counts when using generative ai. *Forbes*, 2024. URL https://www.forbes.com/sites/lanceeliot/2024/05/18/hard-evidence-that-please-and-thank-you-in-prompt-engineering-counts-when-using-generative-ai/.

[8] Hila Gonen, Yoav Levine, Ori Ram, Yoav Shoham, and Amnon Shashua. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2024. URL https://arxiv.org/abs/2212.04037.

[9] Jae-Hee Kim, Sung-Min Park, and Ji-Soo Lee. Politeness in prompt engineering: A cross-cultural analysis. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1823–1835, 2024. URL https://aclanthology.org/2024.emnlp-main.123.

[10] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Mind the gap: A balanced comparison of pre-trained and fine-tuned language models for few-shot learning. *arXiv preprint arXiv:2210.15303*, 2022. URL https://arxiv.org/abs/2210.15303.

[11] Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Poczós, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, 2020.

[12] Christopher Miller, Peggy Wu, H Funk, Lewis Johnson, and H Viljalmsson. A computational approach to etiquette and politeness: An 'etiquette engine™' for cultural interaction training. In *Proceedings of BRIMS 2007*, 2007.

[13] Kshitij Mishra, Mauajama Firdaus, and Asif Ekbal. Predicting politeness variations in goal-oriented conversations. *IEEE Transactions on Computational Social Systems*, 10(3):1095–1104, 2023. doi: 10.1109/TCSS.2022.3156580.

[14] David Moore. How to talk to ai part 2 - good prompt/bad prompt, 2024. URL https://davidmoore.io/how-to-talk-to-ai-part-2-good-prompt-bad-prompt/.

[15] Priya Priyanshu, Firdaus Mauajama, and Ekbal Asif. A multi-task learning framework for politeness and emotion detection in dialogues for mental health counselling and legal aid. *Expert Systems with Applications*, 218:119625, 2023. URL https://doi.org/10.1016/j.eswa.2023.119625.

[16] Priya Priyanshu, Firdaus Mauajama, and Ekbal Asif. Computational politeness in natural language processing: A survey. *arXiv preprint arXiv:2407.12814*, 2024. URL https://arxiv.org/abs/2407.12814.

[17] Diogo Silva, David Semedo, and João Magalhães. Polite task-oriented dialog agents: To generate or to rewrite? In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 304–314, 2022.

[18] Hao Wang, Ziqi Yin, Daisuke Kawahara, and Satoshi Sekine. How well can language models understand politeness? *IEEE Transactions on Computational Linguistics*, 2024. URL https://ieeexplore.ieee.org/document/10195007.

[19] Ziqi Yin, Hao Wang, Kaito Horio, Daisuke Kawahara, and Satoshi Sekine. Should we respect llms? a cross-lingual study on the influence of prompt politeness on llm performance. *arXiv preprint arXiv:2402.14531*, 2024. URL https://arxiv.org/abs/2402.14531.

[20] Wei Zhang, Ming Li, Xiaojun Chen, et al. Investigating the impact of prompt engineering on the performance of large language models for standardizing obstetric diagnosis text: Comparative study. *Journal of Medical Internet Research*, 26(11):e10884897, 2024. doi: 10.2196/10884897. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10884897/.

[21] et al. Zhou. Prosa: Assessing and understanding the prompt sensitivity of llms. *arXiv preprint arXiv:2410.12405*, 2024. URL http://www.arxiv.org/abs/2410.12405.

10 Appendix

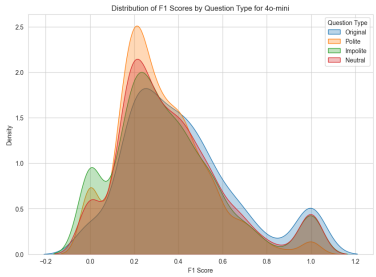
10.1 Question Type Performance Figures

The accuracy score is calculated by checking if the output contains the answer somewhere in it. It is calculated independently for each model, as well as overall.

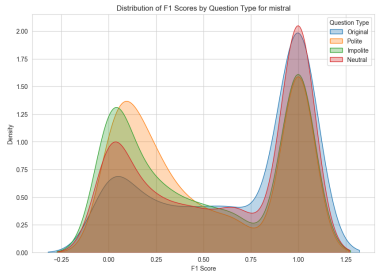
Table 4: Politeness (Accuracy) Scores Across Models

Type	Overall	LLaMA	Falcon	Mistral	4O-Mini
Original	0.736961	0.729730	0.627628	0.768421	0.821856
Neutral	0.689319	0.703148	0.550775	0.730635	0.772455
Impolite	0.624156	0.651303	0.444389	0.677323	0.723054
Polite	0.711879	0.722777	0.563872	0.792914	0.767964

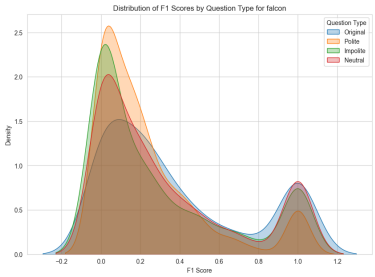
The F1 scores are additionally sample, and the distribution of F1 scores overall and for each model is graphed.



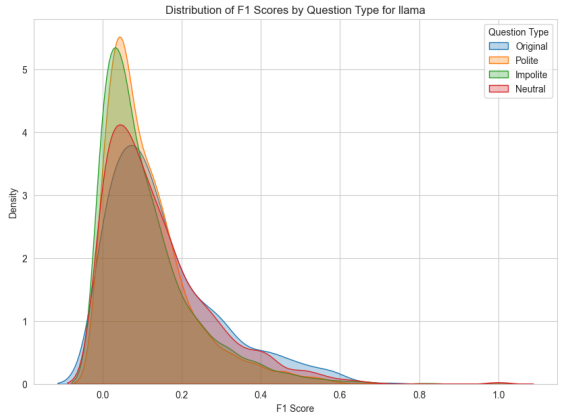
(a) 4o-mini



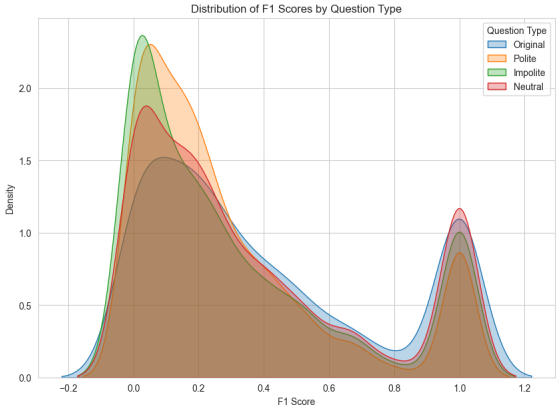
(b) Mistral



(c) Falcon



(d) Llama



(e) Overall

Figure 5: F1 Score distributions for each model across question types

10.2 Output Consistency

Table 5: Consistency across variations for each question type

Question Type	4o-mini	Mistral	Falcon	Llama
Polite (%)	87.125749	75.000000	61.826347	67.664671
Neutral (%)	85.029940	73.353293	62.574850	67.215569
Impolite (%)	81.886228	65.269461	57.185629	57.485030

Table 6: Distribution of Question Consistencies

Consistency Level	4o-mini	Mistral	Falcon	LLaMA
All Correct (%)	72.604790	51.646707	39.071856	44.461078
Mixed Consistency (%)	21.856287	35.329341	41.467066	36.826347
All Incorrect (%)	5.538922	13.023952	19.461078	18.712575

Table 7: Consistency Categories Across Models and Distributions (Percentages)

Consistency Category	Distribution	Overall	4o-mini	Mistral	Falcon	LLaMA
Consistently Right	Polite	60.778443	71.856287	71.257485	41.167665	58.832335
	Neutral	57.522455	71.407186	62.125749	40.568862	55.988024
	Impolite	50.112275	67.065868	54.491018	31.287425	47.604790
Consistently Wrong	Polite	19.797904	18.862275	13.622754	29.790419	16.916168
	Neutral	20.059880	16.916168	16.766467	29.491018	17.065868
	Impolite	26.047904	22.754491	21.107784	41.167665	19.161677
Mixed Consistency	Polite	19.423653	9.281437	15.119760	29.041916	24.251497
	Neutral	22.417665	11.676647	21.107784	29.940120	26.946108
	Impolite	23.839820	10.179641	24.401198	27.544910	33.233533

10.3 Superscoring

Given that there was inconsistencies across variations, we decided to attempt superscoring the results. In doing so, the value for a question type would be counted as correct if any 1 of the 3 variations contained the correct answer. This has implications for extensions wherein somoene may try generating multiple questions, getting answers, and then doing some kind of aggregation function on them.

Table 8: Superscored Correctness Percentage Across Models and Answer Types (Percentages)

Answer Type	Overall	4o-mini	Mistral	Falcon	LLaMA
Polite	80.202096	81.137725	86.377246	70.209581	83.083832
Neutral	79.940120	83.083832	83.233533	70.508982	82.934132
Impolite	73.952096	77.245509	78.892216	58.832335	80.838323

10.4 Politeness Classification and Performance

The generated politeness is not necessarily accurate or as precise as we want. As such, we decided to further quantify these generated politeness values and determine the performance correlation for those. Once again, accuracy is calculated by checking if the correct answer is somewhere within the final answer.

Table 9: Performance Metrics Across Models and Ratings

Model	Rating	EM	F1	Accuracy
4o-mini	1	0.068966	0.347706	0.672414
	2	0.059245	0.356511	0.739583
	3	0.057616	0.365008	0.777427
	4	0.045719	0.355660	0.726517
	5	0.015567	0.322624	0.790215
Mistral	1	0.344828	0.452397	0.517241
	2	0.374837	0.521006	0.676010
	3	0.449585	0.604627	0.731119
	4	0.353577	0.522886	0.757072
	5	0.375093	0.526811	0.809489
Falcon	1	0.105263	0.225804	0.333333
	2	0.147712	0.299907	0.443137
	3	0.164226	0.326964	0.555204
	4	0.092346	0.251598	0.522463
	5	0.080801	0.248981	0.598962
LLaMA	1	0	0.076549	0.413793
	2	0.000655	0.112473	0.660118
	3	0.001581	0.140244	0.708696
	4	0	0.118562	0.681932
	5	0	0.116477	0.738325

The analysis showed yielded the following statistical significance testing from an ANOVA analysis to see if using different politeness values mattered:

Table 10: ANOVA Test Results for Each Model

Model	F-statistic	P-value
4o-mini	6.103967	6.717×10^{-5}
Mistral	21.106071	2.587×10^{-17}
Falcon	22.193086	3.177×10^{-18}
LLaMA	11.472254	2.792×10^{-9}

As can be seen, the p-value is sufficiently low that we can conclude politeness does matter.