Detection of Grice's Maxims Violations: A Deep Learning Approach Using the CMV Dataset

Chen Mordehai, Tamar Shaiman, Yarin Benyamin, Yuval Felendler, Ido Hersko
208383174, 313135865, 208896548, 206116030, 204809867
{mordeche, shaimant, bnyamin, yuvalfel, hersko} at post.bgu.ac.il
Team number 2

1 Introduction

Grice's maxims [1] represent four fundamental principles articulated by philosopher H.P. Grice, which form the bedrock of successful communication: the maxim of quantity, quality, relevance, and manner. These principles aid in understanding implicature, which involves grasping meanings that go beyond the literal words used. Nevertheless, in actual communication scenarios, these maxims may be breached either knowingly or unknowingly, resulting in misinterpretation or the conveyance of supplementary nuances. Detecting violations of Grice's maxims is a complex and challenging task for Natural language processing (NLP) models. It requires navigating the inherent ambiguity and complexity of human language, understanding subtle contextual cues and pragmatic inferences, and recognizing diverse expressions of violations. Despite advancements in NLP technology, accurately detecting violations of Grice's maxims remains an obstacle in replicating human understanding within machine systems. This approach facilitates a deeper understanding of the relationship between human language and computational systems, making it highly pertinent to the subject matter of the course. Our research[1] aims to create a computational model that can identify breaches of Grice's maxims in conversational discourse, focusing specifically on relevance and manner. These two maxims are chosen due to their comparatively straightforward detection process. The other two maxims, quality and quantity, pose greater complexities, as discerning whether a statement is true or false, or determining the appropriate quantity of information, often requires subjective judgment beyond the capabilities of current computational models and even human comprehension. Therefore, our focus is on developing a robust detection system for relevance and manner violations.

[1] github.com/ChenMordehai/Grice-s-Maxims-Violations-Deep-Learning-Approach-for-Detection-for-CMV-Dataset

2 Data

The dataset selected for this project consists of tagged data sourced from the ChangeMyView (CMV) forum on Reddit [2], comprising 10559 comments in 101 conversations(trees) with an average of 104.5 comments per conversation. Each tree has a root comment which is the first comment that starts the conversion and sets the main topic. The decision to employ this dataset is grounded in its wealth of relevant tags, encompassing categories such as 'Sarcasm', 'Ridicule', 'Aggressive', 'BAD', 'Irrelevance', etc. These tags facilitate a nuanced comprehension of the conversations, enabling the detection of potential violations of the maxim principles. As depicted in Table 1, only 12.47% and 4.56% of records are identified as violating the manner and relevance principles, respectively. This limited data is insufficient for robust analysis, and utilizing it as is may lead to poor results due to imbalanced datasets. To address this issue, we will employ various techniques such as under-sampling and over-sampling to re-balance the dataset. These tools will help mitigate the impact of skewed data distributions, ensuring more accurate and reliable outcomes in our analysis.

Table 1: Data Overview

Principle	Tag	Total records violating the principle	Percentage of all data
Manner	'Sarcasm', 'Ridicule', 'Aggressive', 'BAD'	1,317	12.47%
Relevance	'Irrelevance'	575	4.56%

3 Methodology

3.1 Manner Rule Violation

The maxim of manner focuses on how information is conveyed. It suggests that speakers should strive to be clear, concise, and orderly in their communication, avoiding ambiguity, obscurity, or unnecessary complexity. Comments violating the Manner rule may exhibit poor detection of sarcasm, ambiguous or aggressiveness, leading to a misunderstanding of the user's intended tone. In this study, detecting this violation involves fine-tuning a language model, FLAN-T5-base [3], using the tagged data in order to detect these certain linguistic nuances in users' comments.

3.1.1 Manner Violation Detection

We tried fine-tuning two models –
(1) FLAN-T5-base [3].
(2)"mrm8488/t5-base-finetuned-sarcasm-twitter"[2] (pre-trained on Twitter data for sarcasm detection).
Both models are fined tuned using tagged data from the Reddit CMV forum for detecting Manner principle violations. First, we created a new dataset in the form "conversation", "manner violation". 'conversation' was formated by combining parent comment and current comment.

3.2 Relevance Rule Violation

The principle of relevance, as outlined by Grice, emphasizes the importance of staying on topic during conversations. This principle aims to avoid random or incoherent discussions lacking continuity by encouraging relevance. Determining relevance is indeed a challenging task, especially when compared to manners in conversation. It involves constant comparison with both the main topic of discussion and preceding comments to gauge whether a speaker's contribution aligns with the ongoing conversation. In our scenario, this means evaluating each comment not only in relation to the main topic but also in reference to previous comments. This allows for the conversation to diverge from the main topic when necessary, enriching the overall discussion.

3.2.1 Relevance Violation Detection

To detect relevance violation we will compare 2 approaches:

(1) train a binary machine learning model based on the semantic score features of each comment. The semantic score features are combinations of cosine similarity and euclidean distance calculation based on the current comment RoBert Embedding vector and its k previous comments and the conversion root, also after preform embeddings using RoBert. the model will classify each comment to whether its relevant or not to its conversion.

(2) Fine-Tune the FLAN-T5-base on the relevance tagging.

Both of the models are facing the imbalance dataset problem, most of the dataset comments are relevant (95.44%), so before performing the 10-fold CV, we will use under-sampling and over-sampling to examine the model performance.

3.2.2 Semantic score Machine learning model

In order to use machine learning model for the problem we first need to define the problem as we are facing a n supervised binary classification problem with 2 classes, Relevant (class 0) and Irrelevant(class 1). We will need to train the supervised algorithm with samples from the dataset, each sample will be represented as a collection of features. The raw data is a collection of comments which are text. To encode the text we will use a pre-trained Transformer called RoBert. the model Will tokenize the sentence and will embed it into a numerical representation. Then we will be able to measure the Cosine similiarty and Euclidean distance between each comment and its previous comments along with the root comment. To ensure relevance to both the root(the first comment and the main topic of the conversation) and the k previous comments we will use our scoring function:

$\text{Score}(i, r, k) = W_0 \cdot \text{Calc}(i, r) + \sum_{n=1}^{n=k}(W_n \cdot \text{Calc}(i, i-n))$

the i-th comment score will be calculated using a set of weights which will be multiply by the Calc metric. The calc metric will be used for the root and for the K previous comments. The calc metric can be either cosine similiarty or euclidean distance. The main difference between the two metric is that cosine similiarty is used to measure the angle and direction between the comments while euclidean distance is used to calculate the distance and magnitude. The sum of all the wights will be sum up to 1. Because we dont actually know what is more importent the root or the comments and to check the relevance to both, for each value of K we will use 3 sets of params. (1) Equal set - all the weights

[2]https://huggingface.co/mrm8488/t5-base-finetuned-sarcasm-twitter

are equal. (2) Root set - the root will get most of the weight, and the previous comments will split the rest. (3) Comment Set - the root will get a small amount of the wieght compared to the comments.

For example, given K = 3, we will have four weights in each set. one for the root and three for the three previous comments. the sets will be:

Equal set = 0.25, 0.25, 0.25, 0.25

Root set = 0.5, 0.25, 0.15, 0.10

Comment set = 0.20, 0.4, 0.3, 0.10

Because there are many Hyper-Parameters such as the Calc metric, K, whether use the root or not and even the weights, we will use permutations of all those hyper-parameters to create 48 features.

We will also add some features that are only comment depended such as the length of the text, Readability score, sentiment and confidence. all of those features can be extracted using the python TextBlob library.

By representing each comment as a 52 features sample, we will try to train different models and measure its performance on unseen samples.

4 Experiments

The manner and the Relevance models are trained separately on the same data but with different target variable. Data labels will refer to either Manner tags or Relevance tag. The models will be evaluated on both the under-sampled and over-sampled datasets, by measuring the mean precision, mean recall, mean F1-score and mean AUC-ROC after using 10-fold CV.

In the Manner detection we trained the FLAN-T5-base and the "mrm8488/t5-base-finetuned-sarcasm-twitter".

In the relevance detection we trained 4 different classic machine learning algorithms each operates in a different way - Linear SVM(margin and support vectors based), XGBoost(boosting based), Logistic Regression(regression and sigmoid based) and K-Nearest-Neighbors(distance based).

To compare our model performance we will use a baseline model which is the chat-GPT 3.5 turbo, an transformer-Decoder model developed by Open-AI mainly for text generation. The model is given the root comment and the comments before the current comment and then asked to either this comment is relevant or not, and does it violating the manner or not. Before the experiments we fed the GPT with the definition of manner and relevance as define by Grace.

4.1 Fine-Tuning Process

The fine-tuning process was uniform across both models and comprised several steps: (1) Loading the FLAN-t5 model from Hugging Face Transformers, (2) Defining training parameters such as batch size, learning rate, and the number of epochs, (3) Pre-processing the data to ensure compatibility with the model, including tokenization, padding, and truncation, (4) Splitting the dataset into training and testing subsets(based on the current fold), (5) Implementing over/under sampling techniques on the training set, (6) Leveraging the Hugging Face Trainer API (Seq2SeqTrainer) for fine-tuning, (7) Employing the trained model to classify new comments, and (8) Assessing the model's performance on an independent test dataset.

5 Results

5.1 Manner

Fine-tuning FLAN-T5-base results are summarized in Table 4 and Table 5. Performing under-sampling produced better results than over-sampling(F1 score of 0.31 compared to 0.28, and AUC ROC index of 0.64 compared to 0.596). We tried to use the FLAN-T5-base that fine-tuned on Twitter Sarcasm Dataset and Fine-tune it on our data. The best model got the results of F1 score of 0.31 and AUC ROC index of 0.634, yet these results are not better than last model.

5.2 Relevance

By trying different Machine leaning algorithms with both under and over sampling, we searched for the best model performance and hyper-parameters. As shown in table 6, linear SVM achieved the best performance in terms of class 1 recall, precision and f1-score. Class 1 is the Irrelevance tag, meaning how well the model can detect and classify Irrelevant comments based on the features we created and the train set which the model has learned from. We can also learn that XGBoost has the worst performance because it is just classifying most of the test set as Relevant and because of the major imbalance in the test set, its accuracy will be high.

Because the results were not good enough,we also tried to use the Fine-tuned FLAN-T5-base here too. This time the model was tuned by the Relevance tag as the target variable.

As shown in Table 7 and Table 8, By performing over-sampling we got the results of F1 score of 0.24 and AUC ROC index of 0.59. a much better

results then the Linear SVM in the feature based approach.

5.3 Baseline model - Chat GPT 3.5

We employed the GPT 3.5 turbo API as our baseline model for analyzing comments and identifying violations. We engineered prompts for each rule to detect comment violations, aiming to improve accuracy. The process involved iterating over data conversions separately for each rule by utilizing the same dataset used for each task for consistency. We aggregated the results and transformed them into a binary target variable which are the predictions and then calculated the metrics using the ground truth as done for our models. As can be seen in Table 2, We achieved improved performance compared to the baseline model. Additionally, it was observed that relevance detection is more challenging than manner detection, as evidenced by the differing results. This discrepancy may arise from either the problem definition or the higher data imbalance present in the relevance dataset.

5.4 Statistical Significance

The final test we examine is whether the results are statistically significant, meaning, is there a real difference between the model performances on the same dataset. we used the t-test which used to determine if there is a significant difference between the means of two groups. We set our significant level alpha to 0.05 and if the alpha of the t-test is smaller then 0.05 then we can say that there is a significant difference. As can be seen in Table 3, in relevance detection, there is no significant difference between the GPT and the Linear SVM, but there is an significant difference between the GPT and the Flan-T5 model. The same occurs in Manner detection where there is an significant difference between the GPT and Flan-T5.

6 Conclusion

In conclusion, our study focus on the detection of violations of conversational maxims using various natural language processing (NLP) tools applied to the CMV dataset. We adopted distinct approaches to detect each type of violation, with a particular emphasis on relevance and manner rather then quantity and quality. Despite our efforts, quantity and quality remained challenging and unresolved, prompting us to allocate further resources towards refining our methodologies on relevance and manner.

Throughout our experiments, we explored different models to identify violations effectively. One of the primary challenges we encountered was the imbalanced nature of the dataset, which we addressed through the implementation of various sampling techniques aimed at mitigating bias.

A main aspect of our study involved assessing our performance against a baseline model, ChatGPT, across multiple metrics. Leveraging the FLAN T5 model, we achieved notably improved results in terms of precision, recall, and F1-score, signifying the efficacy of our approach.

Based on our results on that particular dataset, we can carefully say that manner is easier to detect compare to relevance, perhaps because of the challenging nature of relevance which include looking back and comprehension with past comments, or from either the problem definition or the higher data imbalance present in the relevance dataset.

Looking ahead, we recognize the need to expand our labeled dataset in order to make the learning model more robust, particularly by incorporating additional positive samples. Furthermore, we aim to explore ensemble techniques employing a variety of transformers to enhance real-time alerting capabilities, thus advancing the practical application of our research in detecting conversational maxims violations.

References

[1] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975.

[2] Stepan Zakharov, Omri Hadar, Tovit Hakak, Dina Grossman, Yifat Ben-David Kolikant, and Oren Tsur. Discourse parsing for contentious, non-convergent online discussions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 853–864, 2021.

[3] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Appendices

Detection	**Model Name**	**Class 0 - Precision**	**Class 0 - Recall**	**Class 0 - F1-Score**	**Class 1 - Precision**	**Class 1 - Recall**	**Class 1 - F1-Score**	**Accuracy**
Relevance	GPT API	0.923	0.431	0.593	0.071	0.523	0.12	0.441
Relevance	Over Sampling - Linear SVM	0.949	0.428	0.59	0.078	0.656	0.133	0.441
Relevance	**Over Sampling - Flan-T5**	0.96	0.95	0.95	0.28	0.21	**0.24**	0.91
Manner	GPT API	0.8	0.5	0.64	0.16	0.59	0.25	0.51
Manner	**Under Sampling - Flan-T5**	0.92	0.66	0.77	0.21	0.62	**0.31**	0.66

Table 2: Final results with baseline model

Name	t-statistic	p-value
Relevane GPT vs Relevance Linear SVM	-1.33819344	0.229313725
Relevane GPT vs Relevance Flan-T5	-3.019219636	0.023420635
Manner GPT vs Manner Flan-T5	-5.060608273	0.002309293

Table 3: T-test results

Hyper Parameter	**Class0 Precision**	**Class0 Recall**	**Class0 F1-Score**	**Class1 Precision**	**Class1 Recall**	**Class1 F1-Score**	**Accuracy**	**AUC-ROC**
learning_rate=3e-4, batch_size=8	0.89	0.9	0.9	0.24	0.23	0.23	0.82	0.563
learning_rate=3e-4, batch_size=16	0.9	0.88	0.89	0.24	0.24	0.24	0.8	0.567
learning_rate=1e-4, batch_size=8	0.9	0.92	0.91	0.29	0.24	0.26	0.84	0.581
learning_rate=1e-4, batch_size=16	0.9	0.8	0.85	0.24	0.34	**0.28**	0.75	**0.596**
learning_rate=1e-3, batch_size=8	0.89	0.86	0.87	0.2	0.25	0.22	0.78	0.555
learning_rate=1e-3, batch_size=16	0.92	0.14	0.24	0.13	0.91	0.22	0.23	0.524
learning_rate=1e-2, batch_size=8	0	0	0	0.12	1	0.22	0.12	0.5
learning_rate=1e-2, batch_size=16	0.88	1	0.94	0	0	0	0.88	0.5

Table 4: Manner Results with FLAN-T5-base Over Sampling

Hyper Parameter	**Class0 Precision**	**Class0 Recall**	**Class0 F1-Score**	**Class1 Precision**	**Class1 Recall**	**Class1 F1-Score**	**Accuracy**	**AUC-ROC**
learning_rate=3e-4, batch_size=8	0.91	0.14	0.24	0.13	0.91	0.23	0.23	0.521
learning_rate=3e-4, batch_size=16	0.9	0.38	0.53	0.14	0.72	0.24	0.42	0.547
learning_rate=1e-4, batch_size=8	0.93	0.39	0.55	0.15	0.8	0.26	0.44	0.595
learning_rate=1e-4, batch_size=16	0.92	0.68	0.78	0.19	0.56	0.29	0.66	0.621
learning_rate=1e-3, batch_size=8	0.89	0.21	0.34	0.13	0.82	0.22	0.28	0.513
learning_rate=1e-3, batch_size=16	0.92	0.66	0.77	0.21	0.62	**0.31**	0.66	**0.64**
learning_rate=1e-2, batch_size=8	0	0	0	0.12	1	0.22	0.12	0.5
learning_rate=1e-2, batch_size=16	0.87	0.86	0.87	0.12	0.13	0.12	0.77	0.495

Table 5: Manner Results with FLAN-T5-base Under Sampling

Model Name	**Class 0 - Precision**	**Class 0 - Recall**	**Class 0 - F1-Score**	**Class 1 - Precision**	**Class 1 - Recall**	**Class 1 - F1-Score**	**Accuracy**
Original data - Linear SVM	0.939	1	0.969	0	0	0	0.941
Random Over Sampling - KNN	0.937	0.692	0.796	0.056	0.283	0.094	0.668
Random Over Sampling - Linear SVM	0.949	0.428	0.59	0.078	0.656	0.133	0.441
Random Over Sampling - Logistic Regression	0.941	0.526	0.675	0.062	0.487	0.11	0.524
Random Over Sampling - XGBoost	0.94	0.97	0.955	0.086	0.044	0.058	0.914
Random Under Sampling - KNN	0.946	0.494	0.649	0.067	0.566	0.12	0.499
Random Under Sampling - Linear SVM	0.949	0.465	0.624	0.077	0.631	0.133	0.474
Random Under Sampling - Logistic Regression	0.942	0.51	0.662	0.063	0.513	0.113	0.51
Random Under Sampling - XGBoost	0.941	0.475	0.631	0.062	0.54	0.112	0.479

Table 6: Relevance Results based on different ML algorithms and Sampling

Hyper Parameter	**Class0 Precision**	**Class0 Recall**	**Class0 F1-Score**	**Class1 Precision**	**Class1 Recall**	**Class1 F1-Score**	**Accuracy**	**AUC-ROC**
learning_rate=3e-4, batch_size=8	0.95	0.97	0.96	0.25	0.15	0.18	0.93	0.560
learning_rate=3e-4, batch_size=16	0.96	0.93	0.94	0.19	0.20	0.19	0.89	0.575
learning_rate=1e-4, batch_size=8	0.96	0.95	0.95	0.28	0.21	**0.24**	0.91	**0.590**
learning_rate=1e-4, batch_size=16	0.95	0.96	0.96	0.21	0.16	0.19	0.92	0.565
learning_rate=1e-3, batch_size=8	0.95	0.92	0.94	0.15	0.25	0.19	0.88	0.583
learning_rate=1e-3, batch_size=16	0.95	0.99	0.97	0.11	0.03	0.04	0.93	0.507
learning_rate=1e-2, batch_size=8	0.00	0.00	0.00	0.06	1.00	0.11	0.06	0.5
learning_rate=1e-2, batch_size=16	0.94	1.00	0.97	0.00	0.00	0.00	0.94	0.5

Table 7: Relevance Results with FLAN-T5-base Over Sampling

Hyper Parameter	**Class0 Precision**	**Class0 Recall**	**Class0 F1-Score**	**Class1 Precision**	**Class1 Recall**	**Class1 F1-Score**	**Accuracy**	**AUC-ROC**
learning_rate=3e-4, batch_size=8	0.97	0.36	0.53	0.06	0.81	0.12	0.39	0.588
learning_rate=3e-4, batch_size=16	0.97	0.51	0.67	0.07	0.70	0.13	0.52	0.607
learning_rate=1e-4, batch_size=8	0.96	0.19	0.32	0.05	0.84	0.10	0.23	0.517
learning_rate=1e-4, batch_size=16	0.97	0.56	0.71	0.08	0.67	0.14	0.56	0.616
learning_rate=1e-3, batch_size=8	0.97	0.25	0.40	0.07	0.89	0.13	0.29	0.568
learning_rate=1e-3, batch_size=16	0.97	0.72	0.83	0.11	0.59	**0.19**	0.71	**0.654**
learning_rate=1e-2, batch_size=8	0.94	1.00	0.97	0.00	0.00	0.00	0.94	0.5
learning_rate=1e-2, batch_size=16	0.94	1.00	0.97	0.00	0.00	0.00	0.94	0.5

Table 8: Relevance Results with FLAN-T5-base Under Sampling