Stance and Persuasiveness Evaluation Using LLMs and Machine Learning

Rory Cooke, John Darnall, John Deibert, Connor Paladino
rtc25@pitt.edu, jhd36@pitt.edu, jad355@pitt.edu, cjp96@pitt.edu

abstract **-** *This paper describes an attempt to utilize LLMs as a tool to analyze social media posts containing both text and images to accurately evaluate each post's stance on controversial topics and the corresponding image's persuasiveness. This attempt uses an LLM to vectorize the dataset into a feature set and classify each post using various machine learning algorithms.*

I. INTRODUCTION

With Large Language Models (LLMS) growing in strength and decreasing in size, there are new methods emerging to classify and moderate text and images. The 2023 ImageArg workshop challenges attendees to use machine learning methods to correctly classify tweet-image pairs by stance and persuasiveness. Our attempt at this problem utilizes Llama 3.2 11B (a flagship LLM by Meta) to vectorize these tweet-image pairs.

By utilizing new technology, we were able to find better results than the previous year's contestants. Our strategy followed a simple, yet robust approach: select a LLM for local hosting, vectorize data using carefully engineered LLM prompts, and run the vectorized data through multiple different machine learning models to discern which model would work best for our challenge.

II. MODEL SELECTION

The first challenge faced when beginning this project was finding a LLM that was powerful enough to properly process the full dataset, yet small enough to run locally, and capable of image processing. Due to the limitations of the strongest available PC (10 GB VRAM, 32 GB RAM), this process proved to be difficult.

Meta's Llama 3.2 Vision met all the requirements. The model was lightweight enough to run locally, had vision capabilities and efficiently utilized the low available computing power. Although it did meet *minimum* requirements, the model still underperforms against many popular, readily available models.

Table 1 provides a comparison between Llama 3.2 11b, its stronger counterpart Llama 3.2 90b, and Open AI's Chat GPT 40-mini.

Benchmark	**Llama 3.2 11b**	**Llama 3.2 90b**	**Chat GPT 40-mini**
MMMU	50.7	60.3	59.4
MMMU-Pro Standard	33.0	45.2	42.3
MMMU-Pro Vision	23.7	33.2	36.5

Table 1
Comparison between popular LLMs.

These results highlight a known issue with our method, computing power. As more computing power is available for models, the models perform better. With our limitation on computing power, our selected method cannot perform at its highest level. The computing issues with LLMs is slowly being addressed, and we predict that models stronger than Meta's Llama 3.2 11b will be available at lower computing thresholds in the near future.

After selecting the model, we needed establish a way to run it locally. An open-source developer tool named "oLlama" was utilized for

the prompting and tokenizing of LLM requests. OLlama's Python library allowed us to build the LLM calls directly into Python scripts, automating the prompting process.

III. PROMPT ENGINEERING

Prompt engineering is the process of eliciting from LLMs high-quality, context-specific output. More specifically, prompt engineering is an art whose main objective is to come up with clear, structured, and goal-oriented inputs that help guide the model in developing responses. In our project, this was done both textually and visually, with nuanced approaches in order to ensure that the LLM would effectively differentiate between abortion and gun control topics.

The results heavily depend upon the design of clear instructions tailored according to its special characteristics. In textual cases, prompts were constructed which test the presence of topics of speech based on their defining words or sets of phrases. Examples included in a set of prompts are: "Does the content talk about women's reproductive rights, abortion-related issues? Confidence scores for 0 to 1. Similarly, the image-based prompts had to do with visual symbolism and contextual clues. Overall questions such as "Is the image about firearms, guns, or weaponry?" guided the LLM during preliminary classification. The more specific ones included protest imagery, using terms such as "pro-choice" or "pro-life" in connecting textual and visual elements. These approaches show specificity reduces ambiguity and makes models reliable across various input types.

Another point of challenge was balancing specificity at the prompt level with model capacity. Prompts that are significantly too broad create a higher risk to create irrelevant or too general results, while hyper-specific ones shrink the model's interpretive flexibility. For example, developing nuanced stances within texts and images required integrating legal and organizational terminologies to ensure relevant aspects were captured accurately without oversimplification.

Further complications were then introduced into the processing of the visual data. Images often contain information on many levels, from explicit text to implicit symbolism. These were overcome through creative solutions-such as using initial image summaries in requests to avoid model biases and built in restrictions. This creative prompt engineering has improved the accuracy of classification but also underlined the adaptability needed when working with multimodal inputs.

Another insight that arose was that prompt engineering is iterative. Initial findings were reviewed for their accuracy and reliability, in order to further refine textual and visual prompts. This actually is a very important feedback loop when dealing with LLMs as it directly impacts subsequent tasks like creating structured data representations. Different prompts on the textual and visual datasets allowed for efficient feature extraction, including seven features for text and five for images, which eventually combined into one 12-feature vector for stance and persuasiveness classification.

This has implications far beyond the bounds of this project. The confidence scoring, which ranged from 0 to 1, added to the prompts was not only quantifying the model's degree of certainty but was a metric whereby the performance of the model would be evaluated. It is this dual-purpose methodology that helps bridge the gap between machine interpretation and human validation, furthering the understanding of how LLMs process and contextualize complex data. These findings open avenues for further explorations in prompt engineering, such as how to optimize prompts for emerging applications including real-time analysis and interactive systems. The nuanced final result of our project is greatly indebted to the strategies of prompt engineering employed. With emphasis on clarity, specificity, and iterative refinement, we have not only maximized the LLM's potential but also

contributed vitally to the valuable insight on the wider practice of prompt engineering in AI-driven systems.

IV. DATA PROCESSING

In order to properly run the data through classification methods, we must first vectorize the data. In order to vectorize, the data was first split into 8 separate sub-sections: training gun control tweets, training gun control images, testing gun control tweets, testing gun control images, training abortion tweets, training abortion images, testing abortion tweets, testing abortion images.

After the data was split into sub-sections, a python script was utilized to run each tweet through the LLM, applying the prompt mentioned in the previous section. The script output a .csv file that represented the vectorized data for each tweet. The process was repeated to obtain vectorized data for each image. Following this process two vectors were present for each image-tweet pair, a 7 feature vector outputted by the tweet prompt, and a 5 feature vector outputted by the image prompt. Each vector had an accompanying "tweet id" that would act as the primary key for matching the images and tweets together following the vectorization.

Once the vectors for each tweet-image pair are complete for each sub-section, the data was combined into four final files: train gun control, test gun control, train abortion, test abortion. All files contained a 12 feature vector, with two binary classification columns for support (0 for not support 1 for support) and persuasiveness (0 for not persuasive 1 for persuasive).

V. CLASSIFICATION

To classify the stance of tweets on topics such as abortion and gun control, our group employed four machine learning algorithms: logistic regression, support vector machines (SVMs), convolutional neural networks (CNNs), and decision trees. Each model was chosen for its specific strengths, allowing us to explore different approaches to understanding the textual and visual data. Our process began by pre-processing the tweet text to ensure consistency and compatibility with the machine learning models. This included converting the text into confidence scores, so these machine learning models can have ease classifying each stance, as mentioned in the previous section.

We began our analysis with logistic regression, a probabilistic model known for its simplicity and efficiency using cost functions, sigmoid functions, and more. Logistic regression is particularly well-suited for binary classification tasks, making it an excellent starting point for determining whether a tweet supports or opposes a given issue. By converting textual data into numerical features, such as our confidence scores for each classification question, we trained the model to identify linear relationships between specific words and their corresponding stances. The variables in logistic regression provided a clear understanding of how each individual prompt question contributed to the classification. However, its reliance on linear decision boundaries limited its ability to capture nonlinear relationships in the data, particularly when the language used in tweets was nuanced or ambiguous.

Next we implemented the SVM, which is highly effective for high-dimensional data and good in situations where there may be false positives or false negatives in the data. By using an extension of the decision boundary, we can eliminate potentially false classified data points in the SVM's hyperplane area. This capability allowed the SVM to help improve the results of our data and models by using its hyperplane to eliminate bad data. While the SVM provided this positive outcome, we also had to be careful with tuning the parameters such as the kernel and regularization strength to optimize performance. Additionally, the cost of training the SVM increased with the size of the dataset.

The CNN was used with the goal of identifying any intricate patterns in the datasets. Since CNNs are known to be used in image

processing, we decided to give this a trial run with our tweets since some tweets had images attached to them. Also to note, CNNs can be used for text classification as well, which helped further our results with the CNN model. For example, CNNs are good at recognizing multi-word expressions, which was a good initial attempt to try and train models about tweets with longer sentences to be able to further classify our results from the datasets. However, to use the CNN, it requires some complexity to analyze our datasets, which could cause problems in our classification dataset as there would be overfitting.

In addition, we explored a decision tree model to emphasize the interpretability of the classification process. Decision trees are good for hierarchical decisions. So, providing a confidence score to this decision tree was seen as a way to run down this tree's hierarchical branches to lead to a classification or a stance from the tweets. However, due to lack of parameters and other metrics, there was a great chance the decision tree could lean towards an overfitted model.

VI. RESULTS AND ANALYSIS

After running the vectorized data through all four models, the data was presented in *Table 2*. This table consists of the four machine learning models that our group used with all of their results. We can see that the logistic regression model and the SVM were both at the top for getting the best results. However, the SVM model was our best model out of the four listed. The total stance accuracy for gun control was around 84% and 83% for abortion which was not a bad accuracy to get on this project. The SVM also had 13 false positives for gun control and 14 for abortion. The SVM also had 3 false Negatives for abortion and for gun control. Furthermore, we also tested for persuasiveness in our models too. The SVM performed about 67% accuracy for if the tweet was persuasive for gun control and roughly 74% persuasive for abortion. Along with zero False positives for gun control and abortion, and it was paired with 33 false negatives for gun control and 26 false negatives for abortion.

Up next for our best results ended up being the logistic regression model. The logistic regression had a total stance accuracy with gun control of roughly 84% and 83% for abortion. The number of false positives was 14 for each abortion and gun control. The false negatives ended up being 2 for gun control and 3 for abortion. In addition, the persuasiveness accuracy came out to be roughly 67% for gun control and 72% for abortion. This was paired with zero false positives for gun control and 2 false positives for abortion. There were also 33 false negatives for gun control and 26 for abortion.

Next, the decision tree performed the third best. The overall stance for a tweet was 75% for gun control and 83% for abortion. The false positives ended up being 11 for gun control and 13 for abortion. The false negatives ended up being 14 for gun control and 4 for abortion. The overall persuasiveness for the decision tree in this project was 63% for gun control and abortion. The false positives were 12 each, along with 25 each for the false negatives.

The model that performed the worst was CNN. The CNN resulted in 54% accuracy for gun control for the overall stance, and 81% overall stance for abortion. The reason why the abortion accuracy could be much higher than the gun control stance is most likely due to a lack of data leveling in the abortion data. However, the number of false positives for gun control were 11 and zero for abortion. The number of false negatives for gun control was 13 and 19 for abortion. For CNN, the overall persuasiveness was 67% for gun control and 74% for abortion. The number of false positives ended up being zero for both gun control and abortion. The false negatives resulted in 33 for gun control and 26 for abortion.

In summary, the SVM stood out as the most reliable and accurate model for both stance and persuasiveness classification across both topics,

while the logistic regression proved to be a strong choice, there was slightly lower accuracy which is why the SVM prevailed in our implementation. However, the CNN and decision tree were much lower than anticipated and therefore were easy to avoid for future implementations for this project. These findings in the CNN and decision tree do have importance to our project as it shows what models can perform the best in our case, such as using confidence scores to determine a stance.

VII. CONCLUSION

If we were to attempt this project again from scratch knowing what we do now, there are a few changes we would make to our approach. First, we now know that an SVM classifier yields the highest accuracy with our current model, so there is no need to use any of the other machine learning classifiers. Additionally, due to hardware and timing limitations, we were only able to process the dataset a single time, resulting in a general lack of robustness. If we had access to a more powerful machine ahead of time, we would have processed the data 3 or more times to ensure the feature set we then passed to our classifier was accurate.

Our model also suffered from a lack of training and testing data. Using a 12-feature set in a machine learning classifier typically requires a much larger dataset than what was provided. One option to work around this obstacle is to use the LLM to generate additional artificial data that can then be processed with the original data to increase the number of samples in our feature set. The only reason we did not attempt this approach in this iteration of the project is again due to hardware limitations. If we had been given access to a more powerful machine or a larger model's API, this approach would certainly have had an effect on the accuracy of our model.

REFERENCES

[1] A. Sharma, A. Gupta, and Maneesh Bilalpur, “Argumentative Stance Prediction: An Exploratory Study on Multimodality and Few-Shot Learning,” *arXiv (Cornell University)*, Jan. 2023, doi: https://doi.org/10.18653/v1/2023.argmining-1.18.

[2] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” *arXiv (Cornell University)*, Feb. 2024, doi: https://doi.org/10.48550/arxiv.2402.07927.

[3] A. A. Awan, “Fine-tuning Llama 3.2 and Using It Locally: A Step-by-Step Guide,” *Datacamp.com*, Sep. 29, 2024. https://www.datacamp.com/tutorial/fine-tuning-llama-3-2 (accessed Dec. 11, 2024).

[4] S. Schwab, “3 Ways to Run LLama 3.2 Locally - Sacha Schwab - Medium,” *Medium*, Oct. 2024. https://sacha-schwab.medium.com/3-ways-to-run-llama-3-2-locally-79bfaf161669 (accessed Dec. 11, 2024).