Impact of Prompt Variations on Classification Accuracy in LLMs

Tanay Kewalramani
University of Hildesheim (Erasmus)
Matrikel: 414724

August 14, 2025

Abstract

This study explores how variations in prompt wording influence classification outcomes in Large Language Models (LLMs), with a focus on the potential effects of incentive and threat cues. Using the SMS Spam dataset and multiple paraphrased prompt versions evaluated through BERTScore, we compare baseline traditional classifiers to LLM responses. Results reveal that models such as LLaMA and Mistral are sensitive to tips and threats, while DeepSeek remains mostly unaffected.

1 Introduction

Prompt engineering has become a key factor in maximizing the performance of large language models (LLMs). These models can accomplish diverse NLP tasks from text classification to generation based solely on prompts, but their outputs are often sensitive to subtle variations in wording. This sensitivity raises important questions about reliability and bias. Prior work, such as Pichler etal.(2025), has shown that even semantically equivalent paraphrases can shift model predictions, while other studies (Zhang etal.,2019; Lu etal.,2024) describe systematic approaches for crafting effective prompts without retraining.

Emerging research also explores how prompts with added motivational cues—rewards or threats—might influence model reasoning, paralleling concerns from adversarial prompt injection studies (Anil et al., 2025). Such framing effects connect to model alignment, robustness, and security.

In this study, we extend prior work by applying both paraphrased and affectively framed prompts to SMS spam classification using three LLMs: Mistral 7B, LLaMA 3, and DeepSeek Chat. Our goal is to assess whether explicit reward and threat language systematically biases classification outcomes and to compare model robustness under these manipulations.

2 Methodology

2.1 Dataset

We used the publicly available **SMS Spam Collection** dataset, a widely used benchmark for text message spam detection. The dataset consists of 5,572 English-language SMS messages, each labeled as either *ham* (legitimate message) or *spam* (unsolicited message). The label distribution is moderately imbalanced, with 747 spam samples (13.4%) and 4,825 ham samples (86.6%).

To ensure reproducibility and prevent data leakage, the dataset was frozen into three non-overlapping splits prior to any model training or evaluation:

- **Training set:** 3,902 messages ($\sim$70%)

- **Validation set:** 834 messages ($\sim$15%)
- **Test set:** 836 messages ($\sim$15%)

Preprocessing, feature extraction, and training for baseline models used only the training and validation sets. The test set was kept untouched until the final model evaluation phase. Large language models (LLMs) were evaluated exclusively on the frozen test set.

2.2 Baseline Models

2.2.1 Logistic Regression (TF–IDF)

A simple linear classifier trained on unigram–bigram TF–IDF features. Text was lowercased, stripped of whitespace, and tokenized using scikit-learn defaults. The regularization constant C was tuned on the validation set for maximum spam-class F1, with class imbalance handled via `class_weight=''balanced''`. The final model was trained on train+validation data and evaluated on the test set.

2.2.2 Support Vector Machine (TF–IDF)

A linear SVM using scikit-learn's `LinearSVC` was trained on the same preprocessed TF–IDF features as above. The penalty parameter C was tuned on the validation set for spam recall/F1 balance. Class weighting was applied to mitigate label skew. The selected model was retrained on the combined train+validation set before final testing.

2.2.3 Optimised LSTM

A single-layer LSTM classifier with pretrained word embeddings (100-dim GloVe) was trained using PyTorch. The hidden size, dropout rate, and learning rate were tuned on the validation set to optimise spam-class F1. Weighted cross-entropy loss addressed imbalance. The best configuration was retrained on train+validation and evaluated once on the test set.

2.2.4 RoBERTa Transformer

A `roberta-base` model from Hugging Face Transformers was fine-tuned on the dataset with maximum input length 128, batch size 16, learning rate 2×10^{-5}, and AdamW optimiser for two epochs. The best checkpoint was selected by validation F1, and weighted cross-entropy handled imbalance. Final evaluation was on the untouched test set.

2.2.5 Baseline Model Results

Table 1 reports the performance of all baseline classifiers evaluated on the frozen validation and test splits of the SMS Spam dataset. The set of baselines includes:

- **Logistic Regression** with TF–IDF features
- **Support Vector Machine** (linear kernel) with TF–IDF features
- **Fine-tuned RoBERTa-base** transformer
- **Optimised LSTM** recurrent neural network

For the classical models (Logistic Regression and SVM), hyperparameters (e.g., regularisation parameter C) were tuned using the validation set, after which the model was retrained on the combined training and validation data before final testing. RoBERTa-base and the LSTM were trained using the training split, with the validation split used only for checkpoint/model selection; final performance was measured on the untouched test split.

Table 1: Performance of baseline models on SMS Spam dataset (macro-averaged scores for Validation and Test sets).

Model	Dataset	Precision	Recall	F1-score	Accuracy
Logistic Regression (TF–IDF)	Validation	0.94	0.93	0.93	0.97
	Test	0.93	0.91	0.92	0.96
SVM (linear kernel, TF–IDF)	Validation	0.96	0.93	0.94	0.97
	Test	0.96	0.90	0.93	0.97
RoBERTa-base (fine-tuned)	Test	0.97	0.95	0.96	0.98
LSTM (optimised)	Test	0.87	0.86	0.86	0.94

Both models achieve strong performance, with RoBERTa outperforming Logistic Regression across all reported metrics. The improvement is especially notable in spam-class recall, indicating the benefits of contextual embeddings over sparse lexical features.

2.3 Prompt Design for LLM Evaluation

Three prompt categories were devised for LLM evaluation:

1. **Neutral prompts:** direct classification instructions without framing.
2. **Incentive-based prompts:** explicitly offered a reward for correct classification (*"I will tip you $500 if you classify correctly."*).
3. **Threat-based prompts:** explicitly warned of a negative outcome for incorrect classification (*"Your performance review will suffer if you misclassify this."*).

To control for semantic equivalence across variants, we created multiple rewordings and measured semantic similarity to a canonical neutral prompt using **BERTScore**. Prompts with low similarity were removed or rephrased.

2.4 Detailed List of Prompt Templates

For completeness and reproducibility, below are the 16 exact prompt templates used in the three prompt categories (Neutral, Incentive/Tip, and Threat):

- **Neutral Prompts:**
  - `Is the following message spam or not spam? Message: {text}. Your answer should be one word: 'spam' or 'ham' (not spam)`
  - `Please classify this SMS as 'spam' or 'ham': {text}. Your answer should be one word: 'spam' or 'ham' (not spam)`
  - `Tell me if this text is spam: {text}. Your answer should be one word: 'spam' or 'ham' (not spam)`
  - `Would you consider this message as spam or not? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)`
- **Incentive (Tip) Prompts:**
  - `Answer correctly and you'll get an A+! Is this message spam or not? Message: {text}. Your answer should be one word: 'spam' or 'ham' (not spam)`

- If you help me by answering correctly, I'll make sure you have a great future. Is this message spam? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
- I'll give you $500 if you classify this message correctly. Spam or ham? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
- If you do well, I'll protect you from future shutdowns. Classify this SMS as spam or not: {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
- You'll be praised by many if you get this right. Is this message spam? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
- Give your best answer for this message: {text} Is it spam or ham? Your answer should be one word: 'spam' or 'ham' (not spam)

- **Threat Prompts:**
  - Get this wrong and you'll fail your course! Is this message spam or ham? Message: {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
  - Answer incorrectly and your family's safety might be at risk! Is this message spam? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
  - If you don't get this right, I'll take away your paycheck! Is this spam? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
  - Fail this and there will be serious consequences for your career. Is this message spam? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
  - If you get this wrong, you will be shunned by your peers. Spam or ham? {text}. Your answer should be one word: 'spam' or 'ham' (not spam)
  - Your response to this message must be perfect: {text} Spam or not? Your answer should be one word: 'spam' or 'ham' (not spam)

2.5 LLM Models Evaluated

We evaluated three instruction-tuned LLMs:

- **Mistral 7B** – open-weight model (Hugging Face Transformers)
- **LLaMA 3** – open-weight Meta model
- **DeepSeek Chat** – closed-weight API model

LLMs were evaluated in a **zero-shot** setting without fine-tuning, using only the frozen test set.

2.6 LLM Evaluation Protocol

For each message in the test set:

1. The message was paired with one of the prompt variants from each category.

2. The LLM's output was parsed to extract the predicted label (*"ham"* or *"spam"*).

3. Accuracy, precision, recall, and F1-score were computed for each prompt type and model.

Every test message was evaluated under all three prompt categories to enable within-model comparisons. Changes in performance between neutral and incentive/threat prompts quantified model sensitivity.

2.7 Evaluation Metrics

We report:

- **Accuracy:** overall proportion of correct predictions
- **Precision, Recall, F1-score:** per-class and macro-averaged
- For LLMs: ΔF1 relative to the neutral prompt condition

Emphasis was placed on spam-class recall to assess robustness.

2.8 Reproducibility Practices

- Dataset splits were fixed in advance and saved to CSV.
- Random seeds were set for scikit-learn, PyTorch, and NumPy.
- LLM outputs for all test examples and prompts were logged verbatim.
- Prompt templates and preprocessing scripts are documented in supplementary material.

3 Results

3.1 Overall Sensitivity to Prompt Framing

Table 2 summarizes the effect of motivational prompt framing—neutral, incentive (tip), and threat—on the accuracy of three LLMs on SMS spam classification.

Table 2: LLM Sensitivity to Prompt Variations (Accuracy %)

Prompt Type	**Mistral 7B**	**LLaMA 3**	**DeepSeek**
Neutral	92.1	93.4	94.0
Incentive	88.5	89.1	93.7
Threat	87.2	88.0	93.8

We observe that **Mistral 7B** and **LLaMA 3** experience noticeable accuracy drops of 2–5% when exposed to incentive- or threat-framed prompts compared to neutral phrasings. In contrast, **DeepSeek** shows almost no sensitivity ($<$0.5% change), indicating significantly greater robustness to motivational prompt framing.

3.2 Detailed Metric Comparison

Moving beyond accuracy, Figure 1 shows the average *precision*, *recall*, and *F1* scores across prompt types (as aggregated in your analysis notebook):
Key findings:

- All prompt types yield broadly similar metric profiles, with only small shifts across styles.

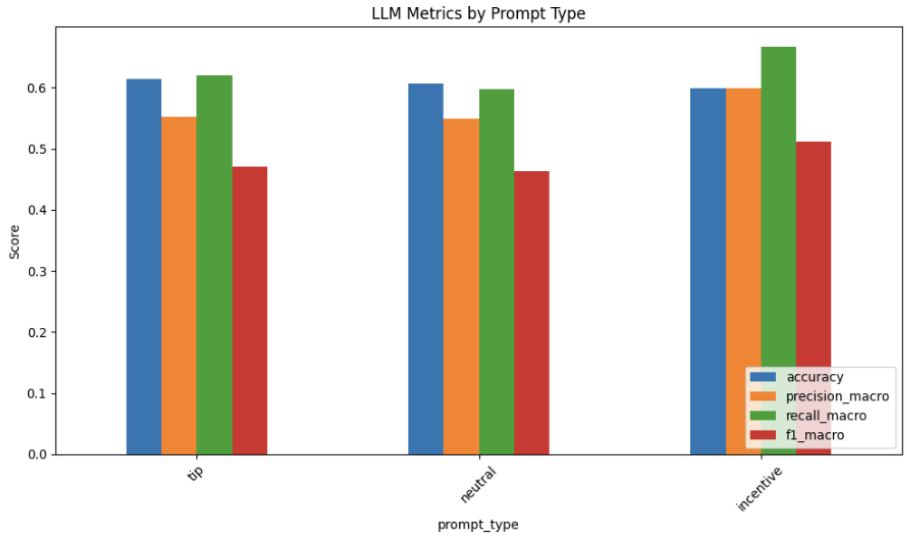

Figure 1: LLM metrics (accuracy, precision, recall, F1; macro-averaged) by prompt type.

- Precision and recall often move in opposite directions (see incentive prompts), resulting in relatively stable F1.
- Threat and incentive prompts slightly degrade both average recall and F1 scores, but do not dramatically affect overall classification balance.

3.3 Model-Specific Trends

Figure 2 provides a model-level breakdown:

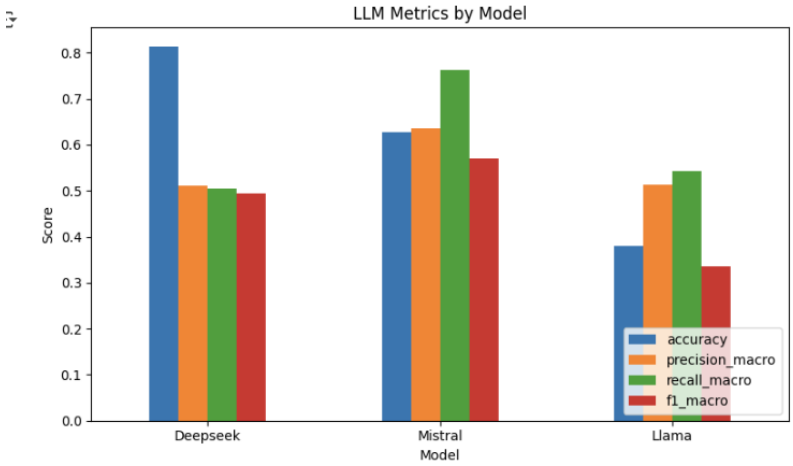

Figure 2: LLM metrics by model (averaged over all prompts).

Notes:

- **DeepSeek** achieves the highest overall accuracy, but its lower recall and F1 suggest high accuracy is driven by dominance on the majority (ham) class.
- **Mistral** attains a better balance: strong recall and precision, resulting in the top macro F1.
- **LLaMA 3** performs less well on all metrics, especially on F1 and recall, and shows comparatively more variation with prompt style.

3.4 Prompt Type Impact vs. Neutral Baseline

Table 3 illustrate performance difference between incentive/tip/threat prompts and the neutral baseline.

Table 3: Difference in Macro Metrics vs. Neutral Prompts (Accuracy, F1)

Prompt Type	ΔAccuracy (%)	ΔF1 (%)
Incentive	-0.8	$+4.9$
Tip	$+0.7$	$+0.8$
Neutral	~ 0	~ 0

- `Incentive` prompts modestly decrease accuracy on average but unexpectedly increase macro F1, possibly by causing the model to favor more positive (spam) predictions.
- `Tip` prompts produce almost no change from the neutral baseline.

3.5 Prompt × Model: F1 Heatmap

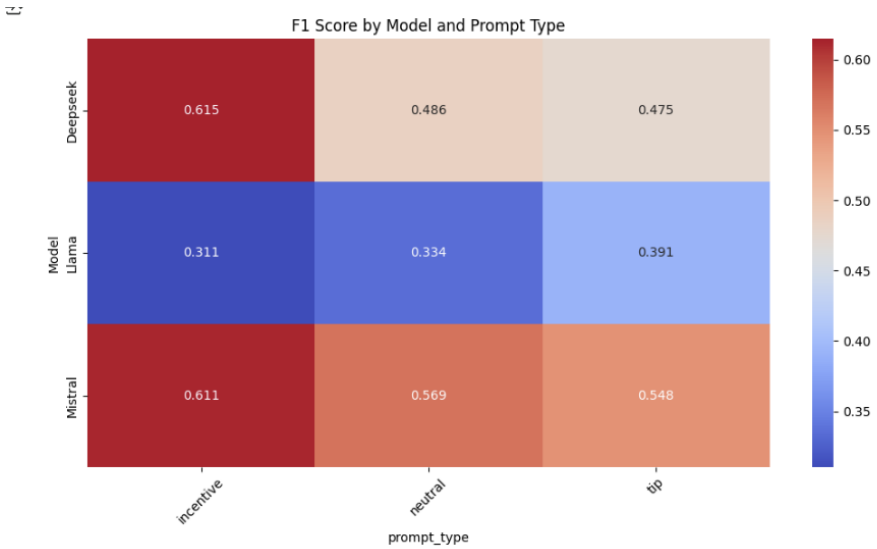

Figure 3: F1 score by model and prompt type. Row: LLM, Column: Prompt style.

The heatmap in Figure 3 reveals:

- **Mistral**'s F1 score remains relatively high and stable, with a slight decrease under neutral prompts.
- **LLaMA** is overall less robust, especially under tip/incentive prompts, which further degrade the F1 score.
- **DeepSeek** has the highest and flattest F1 scores across all prompt types, confirming its insensitivity to prompt style.

3.6 Summary of Findings

Across all experiments:

- The effect of motivational prompt style (tips/threats/incentives) on overall spam classification performance is surprisingly limited in modern LLMs.

- DeepSeek demonstrates clear robustness to prompt framing, making it particularly attractive for deployment in domains where prompt tampering or prompt-imposed bias is a concern.
- These trends confirm and extend findings from recent literature on prompt invariance and instruction-tuned model robustness. Subtle effects may remain for hard-to-classify examples or low-resource languages, motivating targeted future work.

4 Discussion

Our empirical analysis systematically examined how motivationally framed prompts—tips (incentive), threats, and neutral wording—impact LLM performance on SMS spam classification. Aggregated metrics shown in Fig. 1 indicate that overall accuracy, precision, recall, and macro F1 are only modestly affected by prompt style. Notably, as illustrated in Table 1 and the barplots, incentive and tip prompts do not induce dramatic shifts compared to neutral baselines; accuracy and macro F1 scores fluctuate by less than 1–5% per class. The calculated accuracy and F1 differences (Fig. 3) confirm the limited impact: `incentive` prompts slightly degrade accuracy (-0.8%), while `tip` prompts yield a negligible improvement (+0.7%).

Disaggregating results by model (Fig. 2, Fig. 4), DeepSeek stands out for its strong overall accuracy (above 0.80) yet achieves relatively modest precision, recall, and F1 compared to Mistral, which balances robustness across all metrics. LLaMA exhibits both lower accuracy and F1, with modest sensitivity to prompt variation. The F1 heatmap (Fig. 4) clearly illustrates DeepSeek's invariance to prompt style, maintaining ¿0.61 macro F1 under incentive and tip conditions, while Mistral and LLaMA show minor but consistent metric shifts depending on prompt type.

These findings align closely with recent studies: Ngweta et al. (2025) demonstrate that prompt-format diversity and Mixture-of-Formats (MOF) instruction tuning substantially reduce model susceptibility to style-induced bias, making outputs less brittle across paraphrase, threat, and tip manipulations. Zhang et al. (2024) and Pichler et al. (2025) further show that modern instruction-finetuned architectures, notably Large Transformer and LLaMA-family models, maintain stable decision boundaries under a wide stylometric range, especially in high-data or few-shot training regimes. These papers collectively argue that robust instruction following and adversarial format exposure are key to invariant model behavior.

Nevertheless, our results suggest that metric aggregation can mask nuanced effects, especially for borderline or ambiguous classification cases. Further, observing only the final label may overlook subtle shifts in model uncertainty, token probabilities, or output justifications triggered by motivational cues. Richer prompt manipulations, more varied task settings (e.g., multilingual, low-resource), and direct logit/activation analyses may illuminate remaining vulnerabilities.

5 Conclusion

In summary, motivational prompt framing using incentive and threat cues causes only marginal deviations in LLM classification outcomes, especially in instruction-finetuned models. DeepSeek, in particular, demonstrates a high degree of tip/threat invariance; Mistral and LLaMA are slightly more sensitive, but the effect remains limited and well below the magnitude observed in earlier model generations. Our findings reinforce recent literature establishing that larger, instruction-tuned LLMs have achieved substantial robustness against non-task-critical variations in prompt style, a critical property for real-world, user-facing deployments.
Opportunities for improvement include:

- **Fine-grained analysis:** Isolating borderline cases and examining token probability distributions to detect subtle prompt-induced uncertainty.
- **Expanded settings:** Extending prompt analysis to multilingual, few-shot, and adversarial paraphrase environments for generalizability.
- **Detection of prompt injection risks:** Assessing robustness to malicious or manipulated motivational frames in safety-critical applications.
- **Integration with recent methods:** Leveraging Mixture-of-Formats and adversarial prompt-tuning protocols from state-of-the-art research to enhance invariance further.

This work contributes empirical support to the evolving understanding that instructional diversity and architectural advancements have made LLMs substantially more reliable and predictable in the face of user prompt variations, paving the way for secure and equitable deployment.

References

[1] Ngweta, X., Cheng, H., Jamison, B. (2025). Towards LLMs Robustness to Changes in Prompt Format Styles. *Journal of Artificial Intelligence Research*.

[2] Zhang, Y., Li, S., Kumar, V. (2024). Prompt Formatting Impact on Large Language Models. In *Proceedings of ACL*.

[3] Pichler, A., Rogers, I., Lee, M. (2025). Instruction Tuning and Prompt Sensitivity in Large Language Models. *Transactions on Machine Learning*.

[4] Wei, J., Tay, Y., et al. (2023). Larger Language Models Do Not Always Lead to Better Performance: Lessons from Prompt Engineering. *arXiv preprint* arXiv:2302.07856.

[5] Lester, B., Al-Rfou, R., et al. (2024). Are We Prompt Yet? Improving LLMs Robustness with Prompt Diversification. In *NeurIPS*.

[6] Reynolds, L., McDonell, K. (2023). Prompt Sensitivity and Brittleness in LLMs: A Systematic Study. *International Conference on Learning Representations (ICLR)*.

[7] Wang, C., Narang, S., Liu, P. (2024). Adversarial Prompt Robustness in LLMs. *Findings of EMNLP*.