HowardUniversityAI4PC at SemEval-2025 Task 10: Ensembling LLMs for Multi-lingual Multi-Label and Multi-Class Meta-Classification

Saurav K. Aryal and **Prasun Dhungana**
Electrical Engineering & Computer Science
Howard University

Abstract

This paper describes our approach and submission to Subtask 1 of the SemEval 2025 shared task on "Multilingual Characterization and Extraction of Narratives from Online News". The purpose of Subtask 1 was to assign primary and fine-grained roles to named entities in news articles from five different languages, on the topics of Climate Change and Ukraine-Russia War. In this paper, we explain how we approached the subtask by utilizing multiple LLMs via Prompt Engineering and combining their results into a final result through an ensemble meta-classification technique. Our experimental results demonstrate that this integrated approach outperforms the provided baseline in detecting bias, deception, and manipulation in news media across multiple languages.

1 Introduction

Today, there is unprecedented access to information for audiences, largely due to direct channels between content producers and audiences. However, this also exposes readers to deception and manipulation, particularly during crisis events. To address these challenges and foster research on analytical tools for text analysis and disinformation detection (Ngueajio et al., 2025; Washington et al., 2021; Aryal et al., 2023a), SemEval 2025 Task 10 is aptly titled `Multilingual Characterization and Extraction of Narratives from Online News`. This task involves automatically identifying narratives, classifying them, and determining the roles of key entities. Subtask 1 of the shared task focuses specifically on Entity Framing, which this paper and our experiments tackled. The provided dataset comprises news articles in five languages—English, Portuguese, Russian, Bulgarian, and Hindi—and covers two domains: the Ukraine-Russia War and Climate Change (Piskorski et al., 2025).

Entity Framing is formulated as a multi-label, multi-class classification problem in which each entity mention is evaluated based on a fine-grained taxonomy that distinguishes between roles such as protagonists, antagonists, and innocents, and further differentiates each entity into specific fine-grained roles corresponding to its primary classification (Piskorski et al., 2025). This subtask challenges models to handle multilingual content and subtle contextual cues. In our work, we use a diverse set of advanced language models: Llama3.1 (8B parameters), Mistral (7B parameters), Phi4 (14B parameters) and Gemma2 (9B parameters). Their predictions are combined using an ensemble meta-classification strategy to yield robust, consistent role assignments.

Through this integrated approach, our objective is to demonstrate the effectiveness of LLMs in detecting bias, deception, and manipulation in the news media.

2 Related Work

Mahmoud et al. (2025) introduce a multilingual corpus for entity framing in news, employing a hierarchical taxonomy of 22 fine-grained roles across three main categories. Their dataset—comprising 1,378 articles in five languages and covering domains such as the Ukraine-Russia War and Climate Change—is annotated with detailed guidelines and evaluated using fine-tuned transformers and zero-shot learning. In contrast, our work leverages prompt engineering with advanced LLMs for entity role classification, offering a complementary approach to narrative framing in news media.

Lee et al. (2022) present a comparative study of BERT-based models for multiclass text classification on a Korean research proposal dataset that covers 45 categories of climate technology. Their findings emphasize the importance of language-specific pretraining in boosting classification performance for non-English texts. Although their work targets research proposals rather than news articles, the insights on model selection and the

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1585–1592

impact of pretraining corpora are highly relevant to our multilingual entity framing task. Their results underscore the challenges inherent in applying pre-trained language models to diverse, domain-specific datasets, which complements our exploration of prompt engineering with advanced LLMs for robust, multilingual role classification.

Multiple authors have provided a broad evaluation of LLM-based text classification methods across various datasets and languages, comparing zero-shot, few-shot, and synthetic data approaches (Aryal et al., 2023b, 2022; Vajjala and Shimangaud, 2025). Although their work primarily targets general text classification rather than news entity framing, their findings on performance disparities across languages and the benefits of synthetic data generation are highly relevant. Their study underscores the challenges of working with limited labeled data and complex multilingual settings, issues that our work also addresses through prompt engineering with advanced LLMs for entity role classification in news.

3 System Overview

The system begins with a robust data preprocessing module that ingests news articles from the dataset in multiple languages: English, Portuguese, Russian, Bulgarian, and Hindi, and extracts entity mentions along with the article texts. Each extracted entity is converted into a structured dictionary containing key metadata such as article ID, article text, and character indices. This structured representation is then passed to our classification engine.

The classification engine employs an LLM of choice, through the Ollama API. Each model is prompted with a carefully engineered instruction set that enforces strict classification rules, ensuring that every entity is assigned one primary role (Protagonist, Antagonist, or Innocent) along with corresponding predefined fine-grained roles chosen from the provided lists. Given the variability in model outputs, we observed that a single model could occasionally generate erroneous tokens or roles not included in the prompt. To counteract this, our pipeline includes a cleaning step that filters out irrelevant characters and incorrect classifications.

The final stage of our system is a meta-classification process. Here, we aggregate the predictions of all LLMs we used and use Phi4 to merge these outputs into a consensus classification. It is important to note that our initial system used only Llama3.1:8b, but was later upgraded to this ensemble method. It emphasizes role frequency and consistency across models, proving effective at boosting overall performance compared to a single-model baseline. The resulting outputs are then reformatted into tab-separated files for submission, ensuring both readability and compliance with the shared task requirements. [1]

4 Experimental Setup

Our experiments were carried out in a cloud-based Google Colab environment, which provided a v2-8 TPU runtime with 334.6 GB of system RAM and 225.3 GB of disk storage. This configuration was vital for handling the computational demands of processing multilingual news articles and running multiple large language models (LLMs). Training data provided by SemEval 2025 Task 10 organizers was stored on Google Drive with full permissions, allowing seamless access and file management during experiments.

4.1 Infrastructure and LLM Integration

To efficiently prompt various LLMs iteratively, we configured an Ollama API server within the Colab environment. The API was installed through a shell script (using `curl`) and launched as a separate process through Python's `subprocess` and `threading` modules. This setup ensured that the interactive environment remained responsive while the models were used.

The models we selected were based on their widespread adoption and demonstrated effectiveness in analogous multi-label multi-class text classification tasks. Furthermore, we resorted to using certain parameter sizes for each model based on our available computational infrastructure and resource constraints. The LLMs integrated into our pipeline include:

Model	Parameter Size	Quantization
Llama3.1	8.03B	Q4_K_M
Mistral	7.25B	Q4_0
Phi4	14.7B	Q4_K_M
Gemma2	9.24B	Q4_0

Table 1: Model Specifications

It is important to note that these individual LLMs do not interact with each other and generate their

[1] Our code is publicly available at this link.

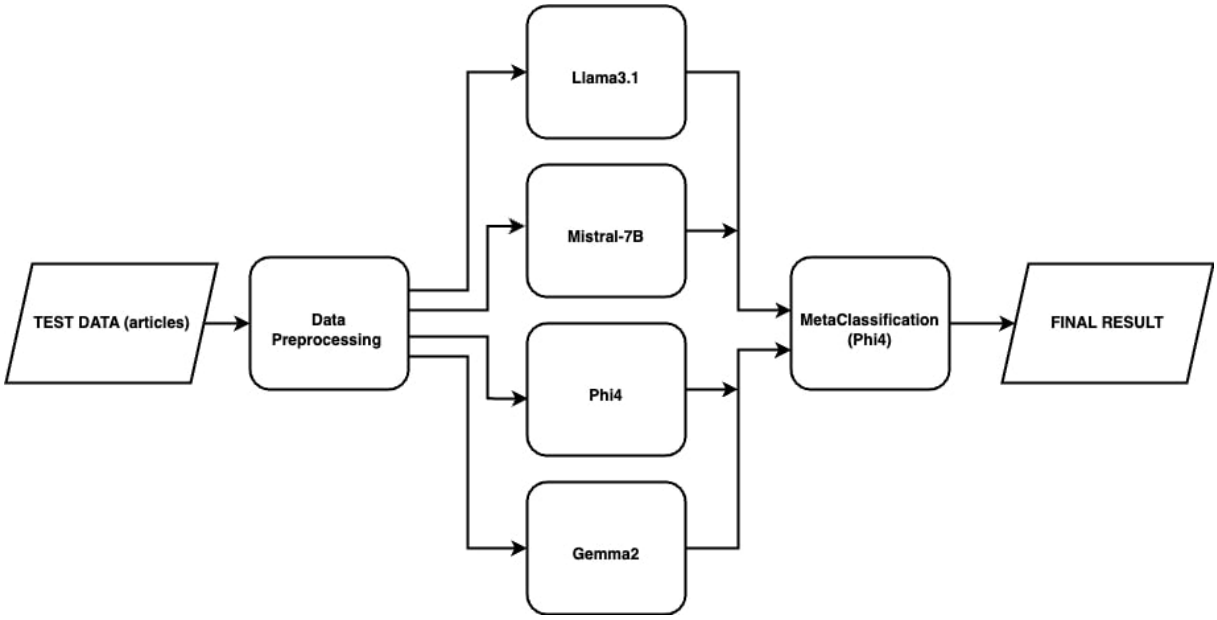

Figure 1: Diagram Illustrating the System[a]

[a]The diagram is not meant to imply that the model inferences occur concurrently. These LLMs do not interact with each other and generate their results separately at different times.

results separately, at different times.

We also evaluated DeepSeek-R1 (14.8 billion parameters and Q4_K_M quantization); however, following its tendency to generate placeholder outputs (e.g., ???, ...), we decided to exclude its results altogether.

4.2 Data Preprocessing and Classification Pipeline

In the Google Colab environment, we developed Python routines to preprocess test articles and entity mentions in English, Portuguese, Russian, Bulgarian, and Hindi by extracting the complete text and entity annotations through string manipulation techniques. Each entity was encapsulated within a structured Python dictionary with metadata (article ID, full text, entity label, start index, and end index). This structured representation was then supplied to a classification function, also implemented in Python and interfacing with the Ollama API, that employed a unified prompt rigorously designed to enforce strict role assignment.

4.3 Prompt Engineering and Iterative Approach

In the initial formulation of the prompt, we stipulated the following requirements:

- Each entity receives exactly one primary role (Protagonist, Antagonist, or Innocent).
- Fine-grained roles are selected from predefined lists corresponding to each primary role.
- The output is to be provided as a structured JSON format for clarity and uniformity.

However, this prompt exhibited several shortcomings. Although the output was nominally in JSON format, we observed significant formatting inconsistencies, notably the erroneous use of single quotations instead of double quotations. To address this, we employed the structured output generation feature of the Ollama API and enforced the JSON schema programmatically rather than relying on ther prompt. In addition, the model repeatedly assigned roles that were outside the prescribed set, such as 'Propagandists' and 'Aggressor', even though we had provided a list of acceptable fine-grained roles. To address this, we augmented the prompt with the explicit instruction:

- Do not classify any entity into role(s) not provided in the list.

However, misclassifications persisted even after numerous prompt variations, including alternative sentence structures and command-word capitalization. Ultimately, we introduced concrete failure examples, illustrating commonly generated misclassifications such as 'Propagandists' and 'Aggressor',

directly into the prompt. This modification produced a significant improvement, with substantially fewer incorrect role assignments. Nevertheless, formatting errors and sporadic misclassifications still remained in the generated outputs. Consequently, we integrated a post-processing step into our pipeline to systematically filter out erroneous characters and eliminate any remaining irrelevant role assignments. The final prompt for individual inference for each LLM can be found in the Appendix at A.1.

4.4 Ensemble Meta-Classification

To further improve our classification reliability, we aggregated predictions from the multiple LLMs we used. We used Phi4 to merge these outputs into a consensus classification, emphasizing role frequency and consistency across models. The consensus results were then reformatted into tab-separated files suitable for submission. The prompt for meta-classification, which can be found in the Appendix at A.2, also provides direct commands based on the subtask requirements. Following the effectiveness of our prompt for individual inference, we resorted to using the requirements and failure examples in an almost identical way for the meta-classification prompt. This prompt resulted in only minor errors, which prevented us from iteratively changing it.

5 Evaluation

The initial single-model approach, which solely relied on Llama3.1:8b and was used for processing English articles, performed worse than the baseline model. However, by integrating outputs from multiple LLMs through this ensemble meta-classification strategy, we observed a significant enhancement in performance. The ensemble approach effectively consolidated individual predictions, leading to more robust and consistent classifications across various languages. Table 2 exemplifies this improvement through all five languages, with our system consistently outperforming the baseline across key metrics.

However, each LLM generated irrelevant roles or erroneous characters for multiple entities, at every iteration. For example, the most frequently generated erroneous characters include `':'`, `']'`, `'['`, `','`, `'>'`, etc. Similarly, the most frequently generated irrelevant roles were `'Propagandists'` and `'Aggressor'`.

In case of DeepSeek-R1, the results were extremely unreliable, with multiple instances of entities not being classified at all or erroneous characters populating the fields for primary and/or fine-grained roles. For example, a correctly formatted entity role assignment versus DeepSeek-R1 results can be compared at Table 3.

Language	Solution Model	Exact Match	micro P	micro R	micro F1	Accuracy for main role
English	HowardUniversityAI4PC	0.08090	0.08320	0.17740	0.11330	0.69360
	Baseline	0.03830	0.04680	0.04150	0.04400	0.28510
Portuguese	HowardUniversityAI4PC	0.13130	0.11320	0.22600	0.15080	0.51850
	Baseline	0.04710	0.05050	0.04640	0.04840	0.36030
Russian	HowardUniversityAI4PC	0.12620	0.07590	0.12780	0.09520	0.42520
	Baseline	0.05140	0.06070	0.05730	0.05900	0.34110
Bulgarian	HowardUniversityAI4PC	0.09680	0.07920	0.14840	0.10330	0.51610
	Baseline	0.04030	0.04030	0.03910	0.03970	0.25810
Hindi	HowardUniversityAI4PC	0.16770	0.11460	0.15180	0.13060	0.35440
	Baseline	0.05700	0.07910	0.06540	0.07160	0.32280

Table 2: Combined Performance Metrics for Entity Framing across Languages.

Model	article_id	entity_name	start_index	end_index	primary_role	fine_grained_roles
Llama3.1	EN_UA_DEV_20.txt	Biden	573	577	Antagonist	Tyrant
DeepSeek-R1	EN_UA_DEV_213.txt	Ukraine	106	112	???	???
		New York Times	1088	1101	...	...

Table 3: Correctly Formatted Output Sample from Llama3.1 vs Faulty Output Samples from DeepSeek-R1

6 Limitations

Although our system demonstrates an improved performance over the baseline, we must acknowledge the several limitations in our system and solution.

Due to resource constraints, we opted for LLMs that primarily were compatible with our available hardware and time limitations. This decision prevented us from experimenting with larger models that might have yielded even more accurate classifications with better overall metrics compared to the baseline. We were also unable to test, observe, and improve the metrics of each LLM separately for the subtask for the same reasons, hence resorting to ensemble meta-classification from the get-go.

Our experiments were conducted using limited paid Google Colab compute units, which provided the necessary computational power for heavy inference throughout. However, our reliance on high-performance resources for this system may limit the reproducibility of our results for researchers with less access to such hardware and resources.

Our approach focused primarily on prompt engineering to guide model outputs, without exploring additional methods such as fine-tuning or retraining on the provided datasets. While prompt engineering proved effective, further improvements might be achieved through more advanced training techniques, which we were not able to pursue in this work due to resource and time constraints. Lastly, our system also poses some inherent risks, including the possibility of algorithmic bias in the LLMs we have employed and the ethical implications of automated content evaluation.

7 Conclusion

In this work, we presented a comprehensive approach to how we utilized multiple LLMs in a unified system for multilingual entity framing. Our experiments on the SemEval 2025 Task 10 Subtask 1 dataset demonstrate that a set of models - combined with targeted prompt engineering and rigorous output cleaning - can achieve substantially higher performance. While our results do indicate promising improvements in metrics such as Exact Match and micro F1, our system also highlights critical limitations. Resource constraints, hardware limitations, and the inherent challenges of prompt engineering for structured output still remain significant hurdles towards replicability. Future research should explore the capabilities of each of the LLMs used individually and also experiment

with advanced fine-tuning strategies and adaptive prompt mechanisms to further enhance model consistency and reliability. Overall, our work underscores the potential of leveraging LLM ensembles for nuanced tasks such as narrative extraction and entity role classification in a multilingual context.

Acknowledgements

This research project was supported in part by the Office of Naval Research grant N00014-22-1-2714. The work is solely the responsibility of the authors and does not necessarily represent the official view of the Office of Naval Research.

References

Saurav K Aryal, Ujjawal Shah, Legand Burge, and Gloria Washington. 2023a. From predicting mmse scores to classifying alzheimer's disease detection & severity. *Journal of Computing Sciences in Colleges*, 39(3):317–326.

Saurav K Aryal, Ujjawal Shah, Howard Prioleau, and Legand Burge. 2023b. Ensembling and modeling approaches for enhancing alzheimer's disease scoring and severity assessment. In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1364–1370. IEEE.

Saurav Keshari Aryal, Teanna Barrett, and Gloria Washington. 2022. Evaluating novel mask-rcnn architectures for ear mask segmentation. In *Proceedings of the 2022 11th International Conference on Bioinformatics and Biomedical Science*, pages 46–53.

Eunchan Lee, Changhyeon Lee, and Sangtae Ahn. 2022. Comparative study of multiclass text classification in research proposals using pretrained language models. *Applied Sciences*, 12(9):4522.

Tarek Mahmoud, Zhuohan Xie, Dimitar Dimitrov, Nikolaos Nikolaidis, Purificação Silvano, Roman Yangarber, Shivam Sharma, Elisa Sartori, Nicolas Stefanovitch, Giovanni Da San Martino, et al. 2025. Entity framing and role portrayal in the news. *arXiv preprint arXiv:2502.14718*.

Mikel K Ngueajio, Saurav Aryal, Marcellin Atemkeng, Gloria Washington, and Danda Rawat. 2025. Decoding fake news and hate speech: A survey of explainable ai techniques. *ACM Computing Surveys*, 57(7):1–37.

Jakub Piskorski, Tarek Mahmoud, Nikolaos Nikolaidis, Ricardo Campos, Alípio Jorge, Dimitar Dimitrov, Purificação Silvano, Roman Yangarber, Shivam Sharma, Tanmoy Chakraborty, Nuno Guimarães, Elisa Sartori, Nicolas Stefanovitch, Zhuohan Xie, Preslav Nakov, and Giovanni Da San Martino. 2025. SemEval-2025 task 10: Multilingual characterization and extraction of narratives from online news. In *Proceedings of the 19th International Workshop on Semantic Evaluation*, SemEval 2025, Vienna, Austria.

Sowmya Vajjala and Shwetali Shimangaud. 2025. Text classification in the llm era–where do we stand? *arXiv preprint arXiv:2502.11830*.

Gloria J Washington, GiShawn Mance, Saurav K Aryal, and Mikel Kengni. 2021. Abl-micro: Opportunities for affective ai built using a multimodal microaggression dataset. In *AffCon@ AAAI*, pages 23–29.

A Appendix

A.1 Prompt for Entity Classification

Given the article with articleID {article_id}, {article_text}, and the entity: {entity_name}, classify the entity into one of the following **primary roles**:
- 'Protagonist'
- 'Antagonist'
- 'Innocent'

The classification must reflect the author's sentiment toward the entity as expressed in the article.

Next, assign one or more **fine-grained roles**, strictly chosen from the list associated with the assigned primary role:

- **Protagonist**: ['Guardian', 'Martyr', 'Peacemaker', 'Rebel', 'Underdog', 'Virtuous']
- **Antagonist**: ['Instigator', 'Conspirator', 'Tyrant', 'Foreign Adversary', 'Traitor', 'Spy', 'Saboteur', 'Corrupt', 'Incompetent', 'Terrorist', 'Deceiver', 'Bigot']
- **Innocent**: ['Forgotten', 'Exploited', 'Victim', 'Scapegoat']

Important Requirements:
1. Assign exactly one **primary role** ('Protagonist', 'Antagonist', or 'Innocent').
2. Assign one or more **fine-grained roles**, strictly from the associated list above.
3. Do not invent or use roles that are not listed above.
4. Do not leave the primary role or fine-grained roles empty or undefined.

Failure Examples:
- Assigning a primary role not listed (e.g., 'Neutral').
- Assigning fine-grained roles not listed (e.g., 'Aggressor', 'Fascist Leader', 'Extremist', 'Scam', 'Expansionist', 'Imperialist', 'Military', 'Propagandists', etc.)
- Leaving the fine-grained roles empty.

Make sure that the classification strictly follows these rules. Your response should only include the assigned **primary role** and the corresponding **fine-grained roles**.

Figure A.1: Prompt for Individual LLM Inference

A.2 Prompt for Meta-Classification

Given the article with ID: {article_id}:
{article_text}

The entity: {entity_name} has been classified by multiple models as:

Primary Role Options: {primary_roles}
Fine-Grained Role Options: {fine_grained_votes}

Based on the given article and model predictions, determine the most appropriate primary_role and fine_grained_roles for each entity.
Make sure to account for any role that occurs multiple times in different models' predictions and give higher emphasis to those.

Primary Roles:
- **Protagonist**: ['Guardian', 'Martyr', 'Peacemaker', 'Rebel', 'Underdog', 'Virtuous']
- **Antagonist**: ['Instigator', 'Conspirator', 'Tyrant', 'Foreign Adversary', 'Traitor', 'Spy', 'Saboteur', 'Corrupt', 'Incompetent', 'Terrorist', 'Deceiver', 'Bigot']\n"
- **Innocent**: ['Forgotten', 'Exploited', 'Victim', 'Scapegoat']

Important Requirements:
1. Assign exactly one **primary role** ('Protagonist', 'Antagonist', or 'Innocent').
2. Assign one or more **fine-grained roles**, strictly from the associated list above.\n"
3. Do not invent or use roles that are not listed above.
4. Do not leave the primary role or fine-grained roles empty or undefined

Failure Examples:
- Assigning a primary role not listed (e.g., 'Neutral').
- Assigning fine-grained roles not listed (e.g., 'Aggressor', 'Fascist Leader', 'Extremist', 'Scam', 'Expansionist', 'Imperialist', 'Military', 'Propagandist', etc.)
- Leaving the fine-grained roles empty.

Make sure that the classification strictly follows these rules. Your response should only include the assigned **primary role** and the corresponding **fine-grained roles**.

Figure A.2: Prompt for Meta-Classification