LATE-GIL-nlp at Semeval-2025 Task 10: Exploring LLMs and transformers for Characterization and extraction of narratives from online news

Ivan Díaz[1], Fredin Vázquez[2], Christian Luna[3], Aldair Conde[5], Gerardo Sierra[4], Helena Gómez-Adorno[2], Gemma Bel-Enguix[4],

[1]Posgrado en Ciencia e Ingeniería de la Computación, [2] Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

[3]Facultad de Contaduría y Administración, [4] Instituto de Ingeniería, [5] Facultad de Ciencias

Universidad Nacional Autónoma de México

Correspondence: helena.gomez@iimas.unam.mx

Abstract

This paper tackles SemEval 2025 Task 10, "Multilingual Characterization and Extraction of Narratives from Online News," focusing on the Ukraine-Russia War and Climate Change domains. Our approach covers three subtasks: (1) **Entity Framing**, assigning protagonist-antagonist-innocent roles with a prompt-based Llama 3.1 (8B) method; (2) **Narrative Classification**, a multi-label classification using XLM-RoBERTa-base; and (3) **Narrative Extraction**, generating concise, text-grounded explanations via FLAN-T5. Results show a unified multilingual transformer pipeline, combined with targeted preprocessing and fine-tuning, achieves substantial gains over baselines while effectively capturing complex narrative structures despite data imbalance and varied label distributions.

1 Introduction

This shared task "Multilingual Characterization and Extraction of Narratives from Online News" (Piskorski et al., 2025) tackles the analysis of news articles from the Ukraine-Russia War and Climate Change domains through three subtasks. Subtask 1 - Entity Framing, assigning fine-grained roles (protagonists, antagonists, innocent) to named entities using a prompt-based Llama 3.1 8B model (Meta-AI, 2025); Subtask 2 - Narrative Classification, labeling articles via a multi-label XLM-RoBERTa-base classifier (Facebook-AI, 2019); and Subtask 3 - Narrative Extraction, generating explanations for dominant narratives with Google FLAN-T5 (Google-Research, 2022). These tasks are important for understanding how entities are portrayed and how broader narratives are constructed in multi-domain, multilingual settings.

Our key findings indicate that, while pretrained language models help manage complex label sets in multiple languages, class imbalance and domain variability remain challenging. Notable results include Exact Match Ratios up to 0.33670 (Portuguese) for Subtask 1, F1 (samples) up to 0.16300 (English) for Subtask 2, and macro-F1 scores up to 0.69558 (English) for Subtask 3. The system generally excels at frequent labels yet struggles with rare ones, highlighting the need for more balanced data and refined approaches to boost performance across languages.

2 Background

In the field of natural language processing (NLP), significant progress has been made in three key subtasks of SemEval 2025 Task 10: Entity Framing, Narrative Classification, and Narrative Extraction (Piskorski et al., 2025). These studies have used methodologies such as word embeddings analysis (e.g., Word2Vec and multilingual BERT) to capture semantic similarities across languages, as well as clustering and topic modeling techniques (e.g., LDA) to identify thematic patterns in large text corpora.

For Entity Framing, studies such as Kumar et al. (2013) developed Wikipedia-based systems for entity extraction, classification, and tagging in social media, outperforming traditional approaches. In Narrative Classification, Zhang et al. (2005) demonstrated the importance of narrative classification for effective key phrase extraction in web documents, using machine learning methods. In Narrative Extraction, Keith Norambuena et al. (2023) surveyed event-based techniques for extracting news narratives, highlighting their utility in analyzing evolving information landscapes. In the legal domain, Samy (2021) created a linguistic resource for named entity recognition and classification (NERC) in Spanish legal texts, combining regular expressions, external lists, and trained models. These studies employed tools such as machine learning models, transformers, and hybrid techniques, analyzing diverse texts, from news articles to social

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 657–665

media posts and legal documents.

3 System Overview

3.1 Subtask1 - Entity Framing

To approach the Entity Framing track, the following workflow (Figure 1) was applied uniformly across all languages (English, Bulgarian, Russian, Portuguese, and Hindi) to ensure consistency and reproducibility in the multi-label, multi-class text-span classification task.

Our approach addresses the multi-label, multi-class text-span classification task by dividing it into three sequential stages: context extraction, multi-class and multi-label classification. This approach ensures that both the multi-class and multi-label aspects of the task are effectively handled for each language.

Context Extraction For each entity mentioned in the news articles, we extract its surrounding context by capturing the 18 words to the left and right of the entity. This context is then refined using a prompt-based approach with a large language model (**Llama 3.1 8B** (Meta-AI, 2025)). The prompt includes the full news article and the entity, and the model generates a refined context in English, regardless of the original language of the article. This ensures consistency across languages and improves the quality of the context representation. We gave the prompt shown in Appendix A.1 to the model.

Multi-Class Classification: We first fine-tuned a **roBERTa** (Liu et al., 2019) transformer-based model using a dataset with labels closely related to the main roles in our task (Protagonist, Antagonist, Innocent). Then, we performed a sentiment Augmentation. We enriched the context of each entity by incorporating binary sentiment labels (positive or negative) obtained from a sentiment analysis model. Finally, we performed an additional fine-tuning step on the previously fine-tuned model using the sentiment-augmented dataset for each language in the multi-class classification task.

Multi-Label Classification: We cleaned the data in the previously refined generated dataset. We applied a preprocessing function using `spaCy` to clean the text. The details of this function are described later in the paper. Then, we fine-tuned sentence transformers for each language using preprocessed contexts and added multiple emotion labels per context. For English and Portuguese datasets, we used the **sentence-transformers/all-roberta-large-v1** (Reimers and Gurevych, 2019; Liu et al., 2019) model. For the Russian dataset, we used the **sentence-transformers/all-distilroberta-v1** (Reimers and Gurevych, 2019; Sanh et al., 2020) model. For the Bulgarian and Hindi datasets we used the **sentence-transformers/paraphrase-multilingual-mpnet-base-v2** (Reimers and Gurevych, 2019; Song et al., 2020) model. Finally, for each entity, we added multiple emotion labels to its context and generated embeddings. Using cross-validation, we then trained the classifier K-Nearest Neighbors or Random Forest (depending on the language).

3.1.1 Resources Beyond Training Data

For the Multi-Class Classification, we performed **binary sentiment augmentation** (positive or negative) using a model trained on the **sentiment analysis dataset** presented in (Orbach et al., 2021). This model was fine-tuned to assign binary sentiment labels to an entity based on its context. These sentiment labels were added to the refined contexts. The first fine-tuned model was trained using the HVVMemes dataset—a collection of memes related to US politics and COVID-19 (Sharma et al., 2022). This dataset is annotated with three labels: Villain, Hero, and Victim, which closely align with our classification scheme. Specifically, Villain corresponds to Antagonist, Hero to Protagonist, and Victim to Innocent.

For the Multi-Label Classification, we used the **TweetNLP** model (Camacho-Collados et al., 2022), to further enrich the refined contexts generated by **Llama 3.1 8B** model (Meta-AI, 2025). The sentiment labels provided by were added to the contexts.

3.1.2 Use of Llama 3.1 for entities context improving

Our initial approach employed an 18-word window centered on the target entity for context extraction. This approach lacked narrative coverage and included irrelevant tokens, reducing accuracy, Table 1 shows the results. These shortcomings motivated our adoption of Llama 3.1 for context refinement. An LLM processes discourse-level context, maintaining entity coherence across longer spans while filtering noise, yielding more accurate role classification, particularly for ambiguous cases with poor semantic information. Table 2 shows the improvement; 30% of the training data was used for validation/testing.

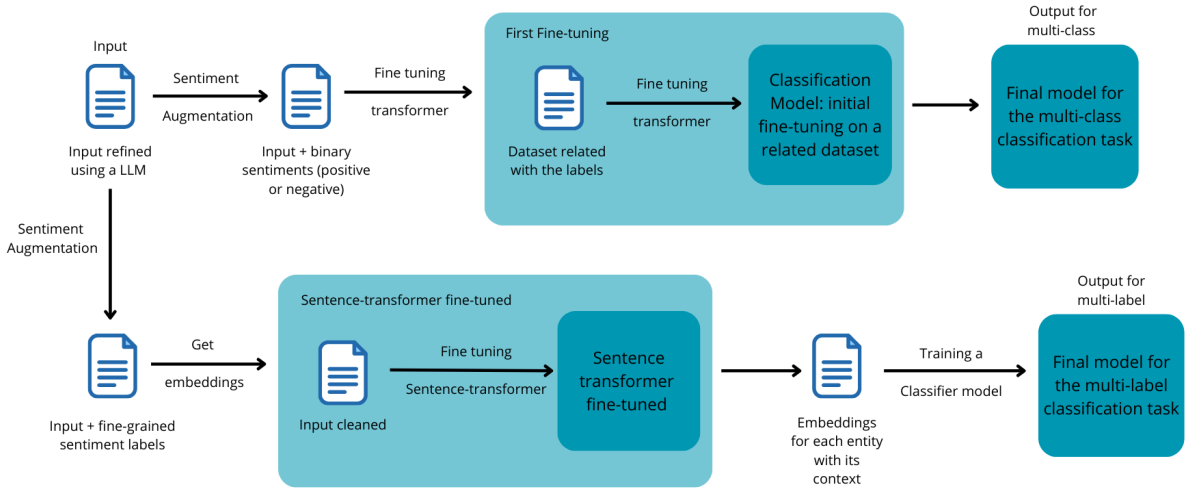

Figure 1: Multi-label classification flow.

	Precision	Recall	F1-score	Support
Antagonist	0.74	1.00	0.85	71
Protagonist	1.00	0.10	0.18	20
Innocent	0.20	0.08	0.12	12
Accuracy	–	–	0.72	103

Table 1: RoBERTa classifier using an 18-word window

	Precision	**Recall**	**F1-score**	**Support**
Antagonist	0.86	0.94	0.90	71
Protagonist	0.65	0.55	0.59	20
Innocent	0.62	0.42	0.50	12
Accuracy	–	–	0.81	103

Table 2: RoBERTa classifier using context generated by Llama 3.1

Llama 3.1 significantly improved the classification of Protagonist and Innocent entities, addressing issues of ambiguity and data scarcity.

3.2 Subtask2 - Narrative Characterization

3.2.1 Key Algorithms and Modeling Decisions

Our approach addresses multi-label multi-class document classification using a two-level taxonomy of narratives (and subnarratives). To effectively capture both levels, we decompose the classification task into three sequential stages (with an adaptation for Russian texts):

1. Binary Classification (Narrative vs. Other): We first distinguish articles that fall under any narrative of interest from those labeled as "Other." This step is omitted in the Russian corpus, as no "Other" category exists there.

2. Multi-Label Multi-Class Classification (Narratives): Once we filter out "Other" articles (or skip directly in Russian), we predict all possible narratives the article may belong to. Each document can have multiple narrative labels.

3. Multi-Label Multi-Class Classification (Subnarratives): For documents assigned one or more narratives in the second stage, we further classify each into their corresponding subnarratives. As before, this is multi-label: an article can contain multiple subnarratives for each of its assigned narratives.

These three stages ensure a clear delineation of responsibilities—separating high-level filtering (stage 1) from more granular classification (stages 2 and 3), as illustrated in Figure 2. We implement all classifiers using XLM-RoBERTa-base, chosen for its multilingual capacity and solid performance across the languages considered in the task.

3.2.2 Model Variants

In our approach, Bulgarian, English, and Hindi employ a three-stage pipeline: (1) a binary classifier distinguishes "Other" from any narrative, (2) multi-label classification identifies possible narratives, and (3) another multi-label classifier targets subnarratives. For Russian, we use a two-stage pipeline by omitting the "Other" label; thus, we directly perform multi-label classification on narratives, followed by subnarratives.

Both variants share the same XLM-RoBERTa backbone (Conneau et al., 2020). Our approach is adapted to varying label distributions across languages using a multilingual transformer model (Wolf et al., 2020).

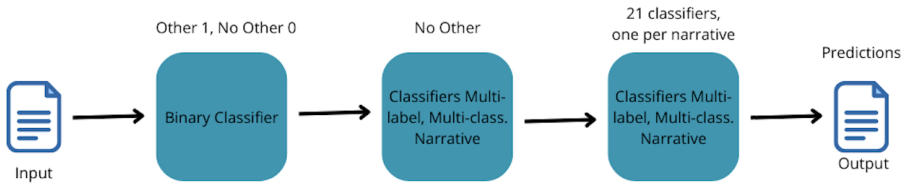

Figure 2: Process diagram for 3 classifiers for Subtask 2

3.3 Subtask 3 - Narrative Extraction

The approach used for this subtask introduces a system based on the fine-tuning of a transformer model for narrative extraction and the generation of an explanation supporting the extracted narrative (Face, 2025). The process is structured into four main stages (illustrated in Figure 3), adapted for four different languages: English, Russian, Bulgarian, and European Portuguese.

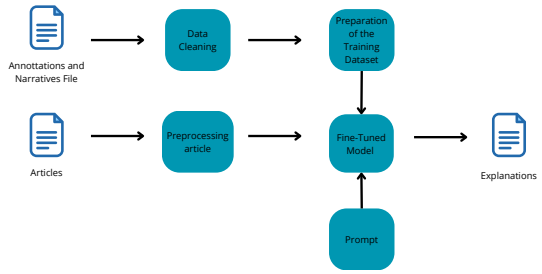

Figure 3: Generate Explanations flow for Subtask 3

3.3.1 Key Algorithms and Modeling Decisions

Data Cleaning: This phase involves two steps. First, cleaning of Annotation and Narrative Files where unwanted prefixes (e.g., "CC:" and "URW:") present in the annotation and narrative files are removed, as these prefixes may be irrelevant and could introduce issues in the generation process. Second, cleaning of articles, where the first two lines of each article are omitted, as they correspond to the title. If a duplicate title is detected, it is also removed. The remaining lines are concatenated to form the complete content of each article.

Preparation of Training Dataset: The system iterates over the annotation rows, and for each row, the corresponding article content is retrieved, and the prompt shown in Appendix A.2 is given.

Then, we fine-tuned the pre-trained Google FLAN-T5 model used for text-to-text generation tasks (Google-Research, 2022). The Hugging Face Trainer API is employed with the hyperparameters as follows: Batch size=4, Epochs=4, Learning rate =3e-5, Save steps= 100 and checkpoint interval=100.

Explanation Generation: The content of all articles is iterated over to generate a structured summary. This summary is derived through the extraction of key sentences using spaCy and NLTK.

Then, we gave a prompt shown in Appendix A.3 is given to the model.

In the postprocessing step, the generated output undergoes refinement that ensures the explanation is adjusted to have a natural opening; overly subjective terms like "obviously" are replaced with more neutral alternatives at the end of the generated text.

4 Experimental Setup

4.1 Subtask1 - Entity Framing

We adopted a stratified split approach to partition the training dataset provided into training, development, and test sets.

For **multi-class classification**, We allocated 90% of the data for training, 5% for development, and 5% for testing. Additionally, for testing purposes, we used the dev dataset provided to simulate a realistic scenario with unseen data.

For **multi-label classification**, We used a 5x2 cross-validation strategy, where the training data was split into 5 parts, with 80% used for training and 20% for testing in each iteration. Again, we used the dev dataset for testing purposes.

4.1.1 Preprocessing and Parameter Tuning

We applied the following preprocessing steps using **spaCy**: First, we removed common stopwords to minimize noise. Next, punctuation marks were eliminated to standardize the text, and extra spaces were reduced to single spaces for consistency. Additionally, we removed numbers and numeric references, as well as special characters, to focus solely on textual content. Finally, lemmatization was applied to reduce words to their base or dictionary

form, ensuring text normalization and improving the quality of the data for subsequent analysis.

About the parameter tuning for **multi-class classification**, the best checkpoint was selected based on the F1-score metric. A softmax function was used to convert raw logits into probabilities.

On the other hand, for **multi-label classification**, using a cross-validation strategy, we were able to obtain the best classifier based on the exact match metric, meaning that the classifier correctly assigns all correct labels to each entity.

The dataset was highly unbalanced, particularly for Innocent entities, as most records were related to Antagonist entities. We did not apply data augmentation or oversampling, as augmenting such data could introduce noise and degrade model performance. To address class imbalance, we first fine-tuned a transformer using the HVVMemes dataset, which was discussed previously.

4.2 Subtask2 - Narrative Characterization

We adopted a stratified split approach to partition our dataset into training and test sets, allocating 80% of the data for training and 20% for testing while maintaining the overall class distribution. Since the dev set presented notable class distribution discrepancies across different languages, we decided to merge the dev set with the training set. Consequently, a combined train+dev set was used for both training and internal validation, and the final test set was solely used to report the official results.

4.2.1 Preprocessing and Parameter Tuning

We used raw text without cleaning—no stopword removal, lemmatization, or stemming—and applied padding and truncation to 128 tokens during tokenization. Fine-tuning was conducted for 3 epochs with a per-device batch size of 8 and a learning rate of 2×10^{-5}; the best checkpoint was selected based on macro-F1 and micro-F1 metrics. No data augmentation was performed. For post-processing, a sigmoid function converts raw logits to probabilities, and labels are assigned when scores exceed a threshold of 0.3, balancing precision and recall (Wolf et al., 2020; Devlin et al., 2018).

Our threshold of 0.3 was chosen based on development set analysis. A standard threshold of 0.5 resulted in too few positive predictions, as the model's outputs rarely exceeded this value, whereas a lower threshold of 0.2 led to many false positives. By evaluating F1 scores, we determined that 0.3 struck the optimal balance, ensuring that texts truly representative of the narrative class surpassed the threshold while minimizing spurious activations.

We did not employ data augmentation or oversampling for classes with very few examples—sometimes as few as 1 to 5—since there were insufficient samples to serve as reliable references. Augmenting such scarce data could have introduced noise and adversely affected the model's performance. Therefore, we aimed to preserve the natural distribution of the narratives.

4.3 Subtask 3 - Narrative Extraction

The training dataset used for the training of the fine-tuned model was divided using an 80/20 split strategy — 80 percent for training and 20 percent for validation. This partitioning was performed using standard functions (such as train test split) to ensure that the validation sample is representative of the entire dataset. The dev set was used for preliminary evaluation of the model, while the test set was exclusively used to report the official results.

5 Results

5.1 Subtask1 - Entity Framing

The model demonstrated strong classification performance across all available languages, consistently ranking within the top 10 positions for each language.

For the **multi-class classification** task the evaluation metric used was **Accuracy**. For the **multi-label classification** task, the **Exact Match Ratio** was the primary metric used to determine the ranking in the competition.

The official results from the leaderboard are presented in the accompanying Table 3.

As can be observed from the results, the performance achieved by the model is highly competitive. The quality of the context surrounding the entities, plays a crucial role in accurately assigning both the main roles and the finer-grained sub-roles. While we utilized **Llama 3.1 8B model** (Meta-AI, 2025) for context refinement, it is possible that more powerful models, such as **DeepSeek R1**, could provide richer and more detailed contexts for the entities.

Finally, an important observation was that the model consistently performed better for languages with Latin-based vocabularies compared to non-Latin languages. This discrepancy could be at-

Language	Exact Match Ratio	micro P	micro R	micro F1	Accuracy for main role	Leaderboard
English	0.3106	0.3671	0.3283	0.3466	0.8383	8
Portuguese	0.3367	0.3872	0.3560	0.3710	0.7172	7
Russian	0.3131	0.3604	0.3524	0.3563	0.6449	10
Hindi	0.2722	0.3711	0.4031	0.3864	0.6361	10

Table 3: Performance scores achieved on the leaderboard for the test set in English, Portuguese, Russia, and Hindi in Subtask 1.

tributed to the limitations of the language model used or the inherent grammatical differences in non-Latin languages, which may make it more challenging to extract meaningful information. Hindi showed strong class imbalance and low performance, partly due to Llama 3.1 generating mixed-language outputs (Hindi and English), which was a problem related to the LLM. This, combined with our limited understanding of Hindi, made it difficult to evaluate and refine the context effectively.

Commonly misclassified labels: Table 4 shows the percentage of misclassifications for each class relative to the other labels in the English Dev dataset. The last column provides examples of entities that were frequently misclassified for each class.

Label	Antag.	Prot.	Inno.	Misclassified samples
Antagonist	95.9%	4.1%	00.0%	climate, sunak, ukraine
Protagonist	44.4%	55.6%	00.0%	russia, conflict, action
Innocent	55.6%	11.1%	33.3%	angeles, conflict, garcetti

Table 4: Misclassified labels on the English Dev dataset by the classifier model.

Ambiguous entities and role confusion often arise with politically charged terms (e.g., "Sunak," "Ukraine") or geopolitical language ("Russia," "conflict"), where context shifts polarity. Negative connotation bias leads to misclassifying Innocent → Antagonist (e.g., "conflict," "action"), as such terms are tied to adversarial contexts. References to places ("Los Angeles") or leaders ("Putin") lack inherent roles but reflect training data biases. Innocent was the most difficult entity type to classify, largely due to the limited number of samples in the dataset, which produces a difficult classification for a multi-label task. Table 5 presents the Exact Match Ratio both overall and broken down by class. It also shows the number of correct predictions relative to the total samples per class, along with the class distribution in the English Dev dataset. This provides a general view of the model's overall performance on the multi-label classification task.

Metric	Overall	Antag.	Prot.	Inno.
Exact Match Ratio	0.32	0.32	0.45	0.00
Correct Predictions	29/91	25/79	4/9	0/3
Class Distribution	100%	86.8%	9.9%	3.3%

Table 5: Fine-grained Role Classification Performance on the English Dev Dataset.

As future work, we propose leveraging verb semantics (e.g., distinguishing between actions such as "negotiate" vs. "attack") to better infer entity roles based on contextual cues. Additionally, incorporating metadata—such as the type of entity (e.g., country vs. individual leader)—could help disambiguate roles and improve classification accuracy. Table 6 presents the results of experiments conducted based on iterative insights. English served as the reference language for approach selection; the methods were first evaluated on the English Dev set, with the best-performing configuration subsequently applied to the remaining languages.

5.2 Subtask2 - Narrative Characterization

We present a multi-stage classification system for English, Bulgarian, and Hindi, targeting both narratives and subnarratives. Our primary evaluation metric is the F1 score at the `narrative_x:subnarrative_x` level, which guided model selection. Overall, our approach outperforms the baseline across all three languages.

In English, binary classification performance is moderate, and multi-label results are strong but affected by class imbalance in the macro-F1. Bulgarian and Hindi achieve high accuracy on the binary task yet show lower F1 for the positive class, also due to imbalance. Across languages, frequent labels are handled effectively, whereas rare labels lower macro-F1.

As shown in Table 7, error analysis indicates frequent misclassifications for labels with minimal training examples, with the system defaulting to more common narratives. This imbalance skews recall toward well-represented classes. Potential improvements include acquiring more diverse data,

Approach	EMR	micro P	micro R	micro F1	Acc. main role
Window-18 words + CountVectorizer	0.04400	0.33330	0.07000	0.11570	0.80220
Llama 3.1 context + CountVectorizer	0.19780	0.23160	0.22000	0.22560	0.86810
Llama 3.1 context + Sentence-Transformer	0.30770	0.31730	0.33000	0.32350	0.86810
Llama 3.1 context + SentenceTransformer + Cross-Validation	0.35160	0.38780	0.38000	0.38380	0.86810

Table 6: Comparison of the performance of different experiments on the English Dev set. The benchmark was established using a RoBERTa multiclass classifier and a KNN multilabel classifier.

oversampling, or using data augmentation to enhance recall for underrepresented narratives.

To validate our approach against a strong general-purpose baseline, we benchmarked it against GPT-3.5 (base). Table 7 shows that our XLM-RoBERTa–based pipeline consistently outperforms GPT-3.5 on binary accuracy and both multi-label micro- and macro-F1 metrics across all languages, demonstrating that specialized fine-tuning yields superior performance compared to off-the-shelf LLM outputs.

Failures centered on underrepresented subnarratives in all three languages,nearly half had fewer than ten examples (many only one), which led to near-zero recall and F1 as the model defaulted to more common labels.

Hindi's poor performance was driven by extreme class imbalance and too few positive examples—causing the binary stage to default to "Other"—while the scarcity of narrative training data hampered reliable learning; translation issues were only a minor factor.

5.3 Subtask 3 - Narrative Extraction

In our latest submission, we integrated data preprocessing, fine-tuned FLAN-T5 model, and employed advanced decoding. Table 8 shows that, despite a robust pipeline, several instances underperformed relative to the baseline.

While the pipeline functioned efficiently across four languages, three evaluations fell below baseline. The model often generated over-simplified outputs, missing key narrative elements, and it struggled with language-specific nuances. These findings suggest the need of more tailored fine-tuning strategies and a richer multilingual training corpus.

6 Conclusions

Our multi-stage architecture—binary filtering followed by multi-label narrative and subnarrative classification demonstrates strong performance on frequent categories and outperforms a baseline system. We evaluated three languages (English, Bulgarian, and Hindi) due to the availability and consistency of data, but the approach generalizes to other languages. Effective capture of major narratives and high micro-F1 suggests robust coverage for well-represented classes. Modular design allows the system to scale and adapt to multilingual contexts with minimal code changes. Reliance on large amounts of representative data for adequate training, classes with fewer than 15 samples defaulted to a single subnarrative, limiting granularity. Finally, we established a pipeline that constructs a model that can both classify and generate narrative explanations. We then fine-tuned a T5-based model for sequence-to-sequence tasks for narrative classification and explanation generation.

Acknowledgments

This work was partially supported by UNAM PAPIIT projects IG400725, IN104424, IG400325 and by the Mexican Government through SECIHTI Proyect FC-2023-G-64.

References

Jose Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa-Anke, Fangyu Liu, Eugenio Martínez-Cámara, et al. 2022. TweetNLP: Cutting-Edge Natural Language Processing for Social Media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Abu Dhabi, U.A.E. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep

Language	Our Approach			GPT-3.5 Base			Sub-F1	Leaderboard
	Binary Acc.	Micro-F1	Macro-F1	Binary Acc.	Micro-F1	Macro-F1	(Sub)	
English	0.6704	0.8983	0.4732	0.6000	0.8500	0.4200	0.50–0.75	23
Bulgarian	0.9204	0.9142	0.4776	0.8800	0.8700	0.4300	0.50–0.73	11
Hindi	0.7530	0.9172	0.4784	0.7000	0.8900	0.4400	0.46–0.70	11

Table 7: Comparison of main performance scores (train+dev set) and leaderboard standings (test set) across English, Bulgarian, and Hindi for SubTask 2, alongside a benchmark of our XLM-RoBERTa–based system versus GPT-3.5 base on binary accuracy and multi-label F1 metrics (test set).

Language	Without Fine-Tuning			With Fine-Tuning			Rank
	Precision	Recall	F1	Precision	Recall	F1	
English	0.63715	0.67313	0.65386	0.70540	0.68674	0.69558	11
Portuguese	0.63904	0.38204	0.60295	0.68395	0.35600	0.67297	7
Russian	0.60153	0.65492	0.62833	0.61043	0.67674	0.64161	8
Bulgarian	0.60417	0.60305	0.63106	0.62947	0.61907	0.62406	7

Table 8: Comparison of results without and with fine-tuning for the test set in 4 languages (Subtask 3).

bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hugging Face. 2025. Text generation task overview. *Hugging Face Documentation*.

Facebook-AI. 2019. Xlm-roberta base. https://huggingface.co/FacebookAI/xlm-roberta-base. [Online; accessed 10-Feb-2025].

Google-Research. 2022. Flan-t5. https://huggingface.co/google/flan-t5. [Online; accessed 16-Feb-2025].

Brian Felipe Keith Norambuena, Tanushree Mitra, and Chris North. 2023. A survey on event-based news narrative extraction. *ACM Comput. Surv.*, 55(14s).

Rohit Kumar, Digvijay S. Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. 2013. Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proc. VLDB Endow.*, 6:1126–1137.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Meta-AI. 2025. Llama 3.1 8b. https://huggingface.co/meta-llama/Llama-3.1-8B. [Online; accessed 16-Feb-2025].

Matan Orbach, Orith Toledo-Ronen, Artem Spector, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. YASO: A targeted sentiment analysis evaluation dataset for open-domain reviews. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9154–9173, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jakub Piskorski, Giovanni Da San Martino, Elisa Sartori, Tarek Mahmoud, Preslav Nakov, Zhuohan Xie, Tanmoy Chakraborty, Shivam Sharma, Roman Yangarber, Ion Androutsopoulos, John Pavlopoulos, Nikolaos Nikolaidis, Ricardo Campos, Alípio Jorge, Purificação Silvano, Nuno Ricardo Guimarães, Dimitar Dimitrov, Ivan Koychev, and Nicolas Stefanovitch. 2025. Multilingual characterization and extraction of narratives from online news. https://propaganda.math.unipd.it/semeval2025task10/. SEMEVAL 2025 TASK 10.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Doa Samy. 2021. Reconocimiento y clasificación de entidades nombradas en textos legalesen español. *Proces. del Leng. Natural*, 67:103–114.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Shivam Sharma, Tharun Suresh, Atharva Kulkarni, Himanshi Mathur, Preslav Nakov, Md. Shad Akhtar, and Tanmoy Chakraborty. 2022. Findings of the CONSTRAINT 2022 shared task on detecting the hero, the villain, and the victim in memes. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 1–11, Dublin, Ireland. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Joe Brew. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. 2005. Narrative text classification for automatic key phrase extraction in web document corpora. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, WIDM '05, page 51–58, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Prompt for subtask 1

Given the following inputs: Full Text: {new text} Entity: {entity} Initial Context: {context}
Refine and enhance the context related to the entity 'entity' to improve text classification for the entity's role. The refined context should:

- Focus specifically on actions, events, or relationships involving the entity that align with one or more of the following roles with its definitions: {descriptions}
- Use the descriptions and examples provided for these roles as a guideline for identifying relevant context.
- Exclude irrelevant or repetitive details, compressing the information into a single concise paragraph.
- Ensure clarity and specificity to support classification, while maintaining alignment with the role definitions.

Provide only the refined context as the output, written as a single paragraph with no introductory phrases or extraneous formatting.

A.2 Constructed Prompt for fine tunning in subtask 3

Input: "Use the following narrative row['dominant narrative'] and the following text to train yourself in order to generate the target explanation. Article-content: row['article content']." **Target**: "Target explanation to generate: row['explanation']".

A.3 Constructed Prompt for explanation generation in subtask 3

Given the dominant narrative row['dominant narrative'], write an explanation of why the following text supports the choice of the narrative. Text: row['processed text']. Use the next template only as a reference to ensure the generated explanation is similar in style row['explanation'].