Framing the Language: Fine-Tuning Gemma 3 for Manipulation Detection

Mykola Khandoga[1], Yevhen Kostiuk[1,2], Anton Polishko[1], Kostiantyn Kozlov[1], Yurii Filipchuk[1], Artur Kiulian[1],

[1]OpenBabylon,
[2]ARG-Tech, University of Dundee, UK

Correspondence: Yevhen Kostiuk ykostiuk001@dundee.ac.uk

Abstract

In this paper, we present our solutions for the two UNLP 2025 shared tasks: manipulation span detection and manipulation technique classification in Ukraine-related media content sourced from Telegram channels.

We experimented with fine-tuning large language models (LLMs) with up to 12 billion parameters, including both encoder- and decoder-based architectures. Our experiments identified Gemma 3 12b with a custom classification head as the best-performing model for both tasks.

To address the limited size of the original training dataset, we generated 50k synthetic samples and marked up an additional 400k media entries containing manipulative content.

1 Introduction

Over the past decade, rapid progress in NLP has coincided with growing concerns about the influence of fake news on electoral outcomes, particularly during the 2016 U.S. presidential election (Gunther et al., 2019). It is perhaps no coincidence that the pioneering efforts to apply NLP methods to the automated detection of manipulative news took place in the late 2010s (Ahmed et al., 2017; Horne and Adali, 2017; Thota et al., 2018). However, these early attempts mostly relied on n-gram feature heuristics and only offered binary classification of the entire document as manipulative. The first fine-grained approach was proposed in 2019 (Da San Martino et al., 2019). The idea of fine-grained analysis of propaganda in the news became the foundation of Task 11 of the SemEval-2020 competition (Martino et al., 2020), where manipulation span detection and technique classification has been presented as separate subtasks.

The UNLP 2025 shared task competition comprises of two subtasks: manipulation span identification (SI) and manipulation technique classification (TC). The two subtasks share the same dataset, which included texts from the Ukraine-related social media content (specifically, Telegram) in Ukrainian and Russian. The objective of the SI is to identify manipulative words in the provided text without the need of classifying the manipulation technique. The TC task is a multi-label classification task, which requires identify whether a text contains one or several manipulation techniques from the following list: Loaded Language, Glittering Generalities, Euphoria, Appeal to Fear, FUD (Fear, Uncertainty, Doubt), Bandwagon/Appeal to People, Thought-Terminating Cliché, Whataboutism, Cherry Picking, and Straw Man. This taxonomy differs from that of SemEval-2020, including categories like Euphoria and Glittering generalities, which are characteristic for the Ukrainian media landscape.

The similarity between the SemEval-2020 and UNLP 2025 tasks offers a unique opportunity to highlight the evolution of NLP methods for solving such problems since 2020, which we explore in Section 2.

The paper is structured as follows. A brief overview of the training dataset along with the description of additional datasets used for this task is provided in Section 3. Our proposed solutions for the two subtasks are described in Section 4. Section 5 contains brief overview of exploratory experiments that we have conducted during development. The final section 6 contains information on the obtained results along with discussions.

2 Related Work

As mentioned in the introduction, SemEval-2020 Task 11 (Martino et al., 2020) marked a milestone in the early days of fine-grained manipulation detection in news. The task demonstrated the dominance of BERT-like encoder-based models, with only sparse use of earlier architectures such as TF-IDF, ELMo, RNNs, and CNNs. At the time, only

Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025), pages 49–54

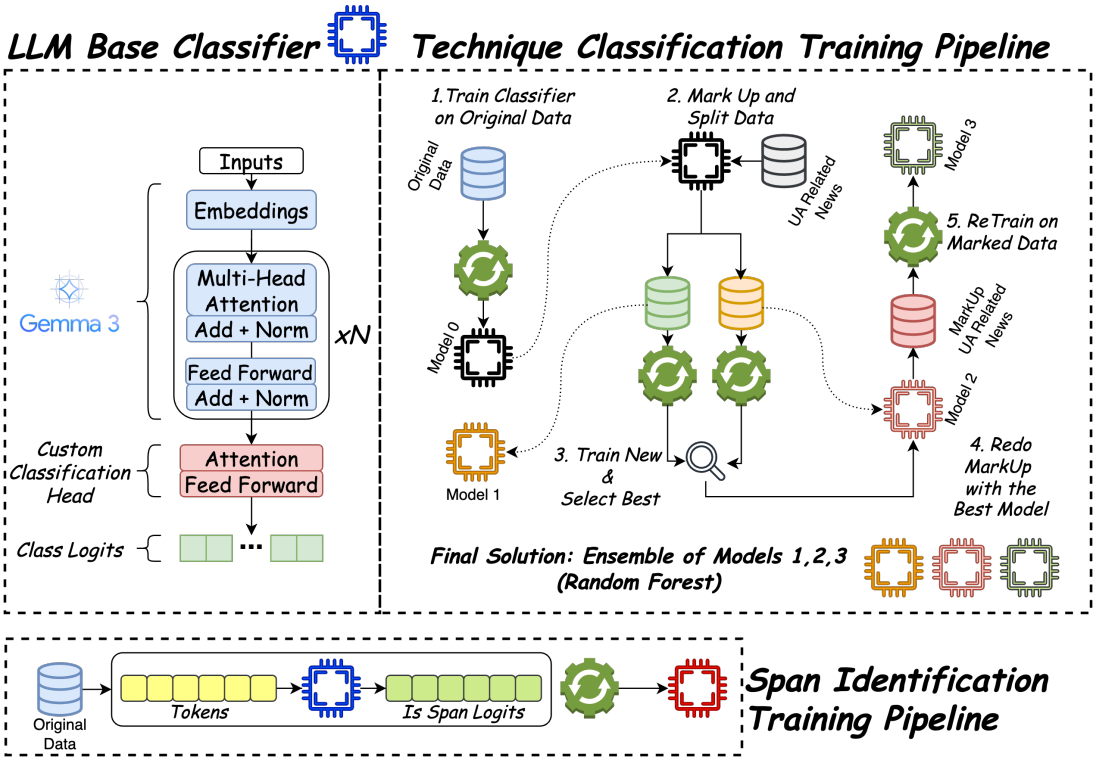

Figure 1: Training pipelines for shared tasks.

one team out of 27 experimented with a decoder-style model (GPT-2), but ultimately reverted to using RoBERTa.

The dominance of BERT-like models began to be challenged in 2023 with the rise of both proprietary and open-weight decoder-based LLMs. Notably, GPT-4 was reported to match the performance of state-of-the-art BERT models on SemEval-2020 Task 11 (Sprenkamp et al., 2023). By the end of 2023, the use of decoder-style LLMs for classification tasks in Ukrainian had become increasingly common (Pavlyshenko, 2023). However, studies have shown that open-weight models such as LLaMA 2 (et al., 2023) and Mistral (Jiang and et al., 2023) can still be outperformed by strong BERT-like baselines in binary fake news classification (Raza et al., 2024).

The shared tasks of UNLP 2025 demonstrate that by 2025, generative LLMs have become as dominant as BERT-based models were in 2020.

3 Datasets

3.1 Shared task dataset

For the shared task, the provided train dataset contained posts from Telegram in Ukrainian and Russian languages. The train dataset included the content of the post, language of the post (not available for submission or test dataset), list of trigger words (target for span identification task), list of manipulation techniques present in the content (target for manipulation techniques classification task). In total, the training set contained 3,822 posts: 2,147 in Ukrainian and 1,675 in Russian. Of these, 2,589 samples included at least one manipulation technique (and therefore trigger words).

The test dataset consisted of 5,735 samples, containing only the raw post content without any labels or metadata.

3.2 Augmented data

Given the limited size of the training dataset and the risk of overfitting with LLMs, our team explored various data augmentation strategies. Specifically, we investigated two approaches: generating a synthetic dataset, and using our best-performing model to annotate additional publicly available data, similar to the shared task dataset. All the resulting datasets are available at our HF repo[1].

Synthetic data generation We have tested two strategies for synthetic generated data: fully synthetic and paraphrasing of the shared task dataset samples. For the paraphrased version of the dataset, the Gemini 2 (Team and et al., 2024) model was in-

[1] https://github.com/OpenBabylon/unlp2025-pub

structed to paraphrase the content from the original train dataset as well as keep the indicated manipulative trigger words (available from the span identification task). For the machine-generated data, we first analyzed the stylistic patterns present in the original dataset for each language. The analysis was done on a subsample of 400 the dataset with GPT-4o (et al., 2024b) model, where it was prompted to analyze and describe styles present in the texts. Then, for each identified style, we sampled 10,000 war-related news articles from Ukrainian and Russian corpora[2] and used Gemini 2 to generate synthetic posts conditioned on both the style and the source content. We did not evaluate the quality of the synthetic dataset, but rather evaluate an impact of LLMs' generated text on the training.

Marked up data We have used a 2-step iterative self-training strategy to markup 400k samples from the Ukrainian news dataset[3] with 200k being relabeled with a better model. The procedure is described in more detail in 4.1. The marked up datasets are available at HuggingFace [4] [5] [6].

4 Solution description

For both subtasks, we fine-tuned the Gemma 3 12B (Team., 2025) model as a base, replacing the original language modeling head with a classification head. The classification head consisted of one-head attention pooling over the final hidden states, followed by a dense output layer for classification. We used a significantly higher learning rate for the classification head layers (7e-5) that for the base model layers (2e-6). The best performing model was selected from 10 epochs of training based on validation set. The training curricula for the two subtasks are schematically illustrated in Figure 1.

4.1 Manipulation Techniques Classification

In the first round of training, we split the shared task dataset into training and validation sets using an 80/20 ratio. During this phase, we experimented with several model architectures (see Section 5), optimized training hyperparameters, and selected the best-performing model (Model 0, see sketch) to label two batches of unlabeled data (200,000 samples each; see Section 3).

In the second round, we trained two new classifiers from scratch (Model 1 and Model 2), using the two newly labeled batches as training data. Model evaluation and threshold tuning were performed using the original shared task training set. Of the two, Model 2 achieved the best performance and was subsequently used to re-label one of the training batches.

In the third round, we trained a final classifier (Model 3) on the data labeled by Model 2. This model achieved the best overall performance.

For our final submission, we built an ensemble of the three top-performing models. Their validation and test logits were combined using a label-wise Random Forest stacking approach with threshold tuning, which improved both performance and robustness across manipulation technique classes. Stacking optimization was again performed using the shared task training set. The code for stacking optimization is available in the public github repository.

4.2 Span Identification

For span identification task we used the same base model as for TC subtask with a different classification head. The classification head outputs a per-token class logits for each manipulation technique. The shared task dataset has been split into train/validation parts (80/20), with validation part used for evaluation and threshold tuning.

5 Experiments

During the development of our solution for the TC subtask, we experimented with parameter-efficient fine-tuning of a variety of models. We believe that sharing these experiments may be of interest to the community, as they provide insight into the trade-offs and capabilities of different approaches. In the following, we describe the most notable experiments. The results of each experiment are presented in the Table 1.

LLaMa 3 8b and LLaMa Guard 3 8b We have started our experiments by fine-tuning LLaMa 3 8b (et al., 2024a) as the baseline model in the class

[2] https://huggingface.co/datasets/zeusfsx/ukrainian-news, https://www.kaggle.com/datasets/makslethal/lenta-ru-news-dataset-v-2-extended

[3] https://huggingface.co/datasets/zeusfsx/ukrainian-news

[4] https://huggingface.co/datasets/OpenBabylon/ua-news-type0-200k

[5] https://huggingface.co/datasets/OpenBabylon/ua-news-type1-200k

[6] https://huggingface.co/datasets/OpenBabylon/ua-news-type1-200k-round2

of 8-12b parameters. In particular, we were interested whether it can beat the BERT baseline. We assumed that the Guard (Inan et al., 2023) model type is more sensitive to the manipulation techniques.

MaxSent-BERT. An interesting set of experiments with the modified BERT architecture (MaxSent-BERT). MaxSent-BERT architecture combines both sentence-level and document-level representations derived from a pre-trained transformer model. We used LiBERTa (Haltiuk and Smywiński-Pohl, 2024) model for Ukrainian. Firstly, the sentence-level features are extracted by splitting input text into sentences with NLTK tokenization (Bird and Loper, 2004). Each sentence is embedded via LiBERTa (Haltiuk and Smywiński-Pohl, 2024). Then, we applied max pooling across CLS tokens of every sentence embeddings. To extract document-level representations, we used CLS token embeddings of the whole input text. Finally, these two representations were summed to create a hybrid embedding that captures both local (sentence) and global (document) context. As a classification head, a linear layer was applied to procude target class probabilities. We trained all the layers of the model with batch size of 4, learning rate of 1e-5, 8 epochs, and BCE loss.

Mistral-UA We tested the Mistral with an extended Ukrainian vocabulary (Kiulian et al., 2024) and additional pre-training on the Ukrainian corpus.

Gemma 3 with synthetic datasets As it was described in Section 3, we have created two synthetic datasets: a fully generated one and a dataset that consists of shared dataset's paraphrases.

6 Results and Discussions

Both subtasks of UNLP-2025 were evaluated using the macro F1 score. For the TS subtask, we experimented with various models and ensembles (see Sections 4 and 5), with the results summarized in Table 1. Our best result for the SI subtask is 0.59096.

The obtained results highlight a shift since SemEval-2020: generative LLMs now consistently outperform BERT-like models and have become the solution of choice for text classification tasks, even despite the limitations imposed by their causal nature.

Experiment	Macro F1 Score
LLaMa 3 8b	0.38870
LlaMa 3 Guard 8b	0.35896
MaxSent-BERT	0.37094
Mistral-UA 7b	0.38255
Gemma 3 12b + paraphrased	0.35228
Gemma 3 12b + generated	0.35982
Gemma 3 12b (Model 0)	0.42232
Gemma 3 12b (Model 1)	0.44754
Gemma 3 12b (Model 2)	0.44934
Gemma 3 12b (Model 3)	0.45134
Model 1 & 2 ensemble	0.45100
Model 1, 2 & 3 ensemble	**0.45265**

Table 1: F1 macro scores obtained in the TS subtask on the full test dataset.

Throughout our experiments, we fine-tuned several decoder-based models, including BERT (Devlin et al., 2019), Ukr-RoBERTa (YouScan, 2023) and LiBERTa (Haltiuk and Smywiński-Pohl, 2024). However, none of these encoder models matched the performance of compact generative LLMs such as Mistral 7B, LLaMa 3 8B, or Gemma 3 12B. The obtained results also provide insights into the factors that contribute to model performance on this type of task. It is no surprise that Gemma 3 12B outperforms the other tested models, as it has the largest vocabulary, the highest parameter count, and is the most recent. LLaMa 3 Guard demonstrates the weakest performance among the evaluated models, possibly due to its lack of support for the Ukrainian language. Mistral-UA, on the other hand, nearly matches the larger and more advanced LLaMa 3, likely due to its extended vocabulary and additional pretraining on Ukrainian corpora. A notable characteristic of the shared task dataset is reflected in the underperformance of models trained on synthetic data. A possible reason is that machine-generated samples lack contextual awareness of the Ukrainian media landscape, particularly with respect to relatively new slang (e.g., "ТЦК" (Territorial Center of Recruitment), "Чмобик" (poorly trained, unwilling, or inept mobilized Russian soldier), "пэтриот" (MIM-104 Patriot, surface-to-air missile system)), uncommon or domain-specific terms (e.g., "Ту-22М3" (Tupolev Tu-22M military plane), "Контрнаступ" (counteroffensive, reffered to Ukrainian liberation campaign), "Ухилянт" (someone who evades mobilization)), and words

used in non-standard or culturally specific senses (e.g., “Ґрад” (BM-21 Grad, a Soviet-designed multiple rocket launcher system or heavy rain), “Мясо” (term used to describe poorly trained, expendable soldiers)).

We hypothesize that the LLMs used in our experiments were trained primarily on pre-invasion data, and therefore lack adequate exposure to this updated vocabulary and context. To test this hypothesis, we trained Wide & Deep (Cheng et al., 2016)-inspired classifier. The model showed higher performance on the validation set than on the test submission. After removing the "wide" component (lemmatized vocabulary per language with Stanza (Qi et al., 2020)) the scores became aligned, indicating that the model likely memorized it.

Overall, we find that the UNLP-2025 shared task provides valuable insights into both the progress of the NLP field and the importance of language- and culture-specific contextual training.

Limitations

Our approach, while effective, is subject to several limitations. Firstly, all experiments were conducted using models with up to 12 billion parameters due to hardware constraints. As a result, we did not evaluate the performance of larger or more recent LLMs (LLaMa 3 70b, Gemma 27b, QWEN 32b), which may offer improved performance for this task.

Secondly, while we introduced a large volume of synthetic and automatically annotated training data, we did not perform a rigorous quality evaluation of this data beyond validation set performance. Consequently, there is a risk that mislabeled or low-quality synthetic samples may have introduced noise during training.

Finally, although our best-performing models achieved strong results, they relied heavily on English-language pretraining and exhibited limitations in their handling of culturally specific or contextually nuanced terms in Ukrainian and Russian. This is particularly evident in their struggle with emerging slang and post-2022 domain-specific terminology. One potential way of mitigating this challenge is to fine-tune the model on a rich corpora of culturally aligned texts before training it on the downstream task.

Acknowledgments

We would like to express our gratitude to **Google** for providing credits used for model training and inference.

References

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. pages 127–138.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. *Preprint*, arXiv:1606.07792.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Aaron Grattafiori et al. 2024a. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

OpenAI et al. 2024b. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Richard Gunther, Paul Beck, and Erik Nisbet. 2019. “fake news” and the defection of 2012 obama voters in the 2016 presidential election. *Electoral Studies*, 61.

Mykola Haltiuk and Aleksander Smywiński-Pohl. 2024. Liberta: Advancing ukrainian language modeling through pre-training from scratch. In *Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP)@ LREC-COLING 2024*, pages 120–128.

Benjamin Horne and Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):759–766.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *Preprint*, arXiv:2312.06674.

Albert Q. Jiang and et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Artur Kiulian, Anton Polishko, Mykola Khandoga, Yevhen Kostiuk, Guillermo Gabrielli, Łukasz Gagała, Fadi Zaraket, Qusai Abu Obaida, Hrishikesh Garud, Wendy Wing Yee Mak, Dmytro Chaplynskyi, Selma Belhadj Amor, and Grigol Peradze. 2024. From english-centric to effective bilingual: Llms with custom tokenizers for underrepresented languages. *Preprint*, arXiv:2410.18836.

G Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. *arXiv preprint arXiv:2009.02696*.

Bohdan M. Pavlyshenko. 2023. Analysis of disinformation and fake news detection using fine-tuned large language model. *Preprint*, arXiv:2309.04704.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *Preprint*, arXiv:2003.07082.

Shaina Raza, Drai Paulen-Patterson, and Chen Ding. 2024. Fake news detection: Comparative evaluation of bert-like models and large language models with generative ai-annotated data. *Preprint*, arXiv:2412.14276.

Kilian Sprenkamp, Daniel Gordon Jones, and Liudmila Zavolokina. 2023. Large language models for propaganda detection. *Preprint*, arXiv:2310.06422.

Gemini Team and Rohan Anil et al. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Gemma Team. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, and Nibrat Lohia. 2018. Fake news detection: A deep learning approach. *SMU Data Science Review*, 1(3):10.

YouScan. 2023. ukr-roberta-base. `https://huggingface.co/youscan/ukr-roberta-base`. Accessed: 2025-04-18.