

Hidden Persuasion: Detecting Manipulative Narratives on Social Media During the 2022 Russian Invasion of Ukraine

Kateryna Akhynko **Oleksandr Kosovan** **Mykola Trokhymovych**
Ukrainian Catholic University Ukrainian Catholic University Pompeu Fabra University
Lviv, Ukraine Lviv, Ukraine Barcelona, Spain
kateryna.akhynko@ucu.edu.ua o.kosovan@ucu.edu.ua mykola.trokhymovych@upf.edu

Abstract

This paper presents one of the top-performing solutions to the UNLP 2025 Shared Task on Detecting Manipulation in Social Media. The task focuses on detecting and classifying rhetorical and stylistic manipulation techniques used to influence Ukrainian Telegram users. For the classification subtask, we fine-tuned the Gemma 2 language model with LoRA adapters and applied a second-level classifier leveraging meta-features and threshold optimization. For span detection, we employed an XLM-RoBERTa model trained for multi-target, including token binary classification. Our approach achieved 2nd place in classification and 3rd place in span detection.

1 Introduction

In times of war, information can have the same power as weaponry. During the 2022 Russian invasion of Ukraine, Telegram emerged not only as a battlefield communication tool but also as the primary source of information for 44% of Ukrainians. Its speed, reach, and anonymity became an important tool for civilians and military actors. However, these features — particularly minimal content moderation and user anonymity — have also made Telegram a favorable environment for influence operations (Vorobiov, 2024).

Manipulation on social media is a complex and nuanced phenomenon. It includes not just factual distortions (i.e., disinformation) but also rhetorical strategies, emotional appeals, and narrative framing that are designed to influence perception or behavior subtly. In this paper, we present the solution¹ to the UNLP 2025 Shared Task,² focused on manipulative narratives detection, which is defined as the intentional use of language and messaging

tactics aimed at influencing beliefs, emotions, or attitudes, without providing clear factual support.

The task includes several challenges that make it particularly complex. First, it focuses exclusively on the textual content of social media posts without incorporating metadata such as user history or engagement metrics. Second, the dataset presents multiple layers of complexity: it is imbalanced across manipulation types, multilingual (primarily Ukrainian and Russian), and multi-label, meaning that a single post can include several manipulation techniques simultaneously. Finally, the span detection subtask requires identifying the exact textual fragments responsible for the manipulation, often implicit, rhetorical, or emotionally charged language that is difficult to isolate.

Given these challenges, we developed a system that achieved second place in manipulation techniques classification and third place in span detection subtasks (see Figure 1). For classification, we fine-tuned the Gemma 2 language model using LoRA adapters and introduced a second-level classifier that leveraged meta-features and custom threshold optimization. For span detection, we trained an XLM-RoBERTa model capable of multi-target, token-level binary classification to locate manipulative spans within posts.

2 Related Work

Our research is based on a growing body of work in detecting propaganda and misinformation analysis. Numerous studies have focused on identifying propaganda techniques in news articles, particularly in the context of SemEval-2020 Task 11. Da San Martino et al. (2020) explored detecting propaganda techniques in news articles through span identification and technique classification tasks.

Similarly to previous research, the UNLP 2025 Shared Task includes two subtasks: manipulation technique classification (a multi-label classifica-

¹<https://github.com/akhynkokateryna/manipulative-narrative-detection>

²<https://github.com/unlp-workshop/unlp-2025-shared-task>

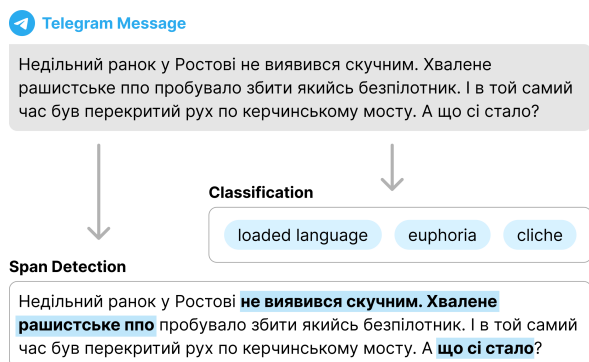


Figure 1: Sketch of manipulation techniques classification and span detection problems

tion) and span detection (a token classification). Within this framework, research from SemEval-2020 Task 11 demonstrated BERT’s remarkable capabilities for propaganda technique identification (Altiti et al., 2020). Further advancing this line of inquiry, Da San Martino et al. (2020) showcased RoBERTa’s performance in addressing both tasks simultaneously.

At the same time, the nature of propaganda on social media evolves continuously, adapting to specific circumstances to remain undetected. Solopova et al. (2023) explored this process by combining machine learning and linguistic analysis to reveal how pro-Kremlin propaganda evolved in the context of the 2022 Russian invasion of Ukraine. In this context, it is important to note that while our work has a similar goal, we focus specifically on detecting manipulative narratives regardless of the factual support of the claim. This distinguishes our approach from fact-checking or knowledge manipulation detection methods (Trokhymovych and Saez-Trumper, 2021; Trokhymovych et al., 2025).

In our case, we are dealing with multilingual Telegram data containing Ukrainian and Russian texts. In this scenario, fine-tuning a multilingual model, such as XLM-RoBERTa, appears to be a more productive approach, as demonstrated in research on hostility identification for low-resource Indian languages (Sai et al., 2021). Moreover, XLM-RoBERTa-based models have demonstrated cross-lingual strengths in other downstream tasks, including those involving Ukrainian and Russian languages (Mehta and Varma, 2023; Trokhymovych et al., 2024).

While Sprenkamp et al. (2023) discovered that fine-tuned RoBERTa outperformed zero and few-shot learning approaches with LLMs for propaganda detection, newer advances in large language

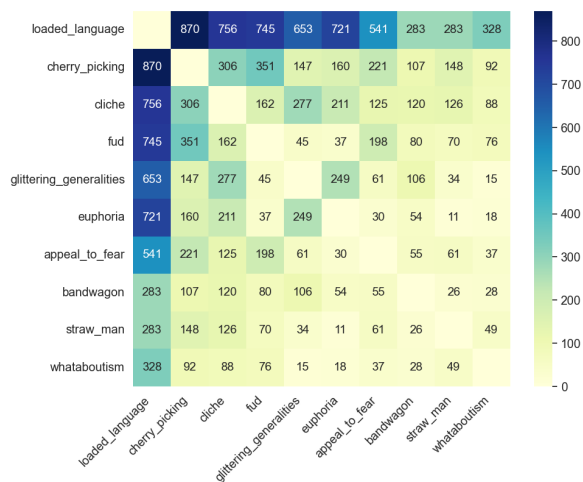


Figure 2: Co-occurrence of manipulation techniques in the combined training and testing sets

models show considerable promise. Recent innovations have developed methods to transform decoder-only LLMs into effective text encoders suitable for classification tasks (BehnamGhader et al., 2024). Models such as Gemma offer particularly interesting customization potential for classification challenges (Team et al., 2024).

Notably, Gemma-family models enable fine-tuning with LoRA adapters and support quantization techniques, making them viable options even with limited computational resources. Building on this foundation, (Kiulian et al., 2024) ventured into fine-tuning both Gemma and Mistral specifically to enhance Ukrainian language representation, providing valuable insights that directly inform our approach to detecting manipulative narratives within Telegram content from the region.

3 Data

The UNLP shared task dataset contains more than 9,500 text samples collected from Telegram channels, with 68% of these collected samples containing manipulative narratives. This dataset forms the basis for a dual-task challenge: classifying manipulation techniques and identifying corresponding text spans.

The data is divided into training and testing sets, with 3,822 samples allocated for training and 5,735 for testing. Among the 3,822 training samples, 2,147 (56%) are in Ukrainian and 1,675 (44%) are in Russian. At the same time, the testing set does not include language labels. Notably, the testing set is further split into public and private sets for leaderboard evaluation.

Each post is annotated for both classification and span detection tasks. Specifically, every sample is labeled with one or more of ten predefined manipulation techniques, detailed in Appendix A. Manipulative text segments are also defined, irrespective of the specific technique involved.

Figure 2 illustrates manipulation techniques’ co-occurrence patterns across training and testing sets. As the distribution of labels is similar in both subsets, we present them together for clarity.

4 Methodology

In this section, we present our approaches for solving the technique classification and span identification subtasks.

4.1 Technique Classification

The manipulation technique detection task is formulated as a multi-label text classification problem, where each input text may contain multiple manipulation strategies. Each sample is annotated with any number of 10 predefined manipulation techniques.

Our best-performing solution involves multi-stage fine-tuning of the instruction-tuned Gemma 2 2B IT model.³ The complete fine-tuning pipeline schema is presented in Figure 3.

Firstly, we fine-tune the model using a causal language modeling (CLM) objective, where the model learns to predict the next token given a left-to-right context. Specifically, we employed the `AutoModelForCausalLM` class from HuggingFace Transformers.

The model was trained to autoregressively generate a comma-separated list of manipulation techniques based on a task-specific prompt. We constructed a dataset of prompt inputs for each training data point, which included:

- an instruction to identify manipulative techniques in a text;
- descriptions of all ten manipulation techniques;
- four few-shot examples, selected from the training set: two were chosen based on cosine similarity between the target text and other texts in the training set, and the other two based on cosine similarity between the target text and the trigger phrases (i.e., manipulative spans in texts) found in other training samples.

To control input length, we select the few-shot examples from the subset limited by texts shorter than 500 characters. To get a vector representation of the texts, we encode them using SentenceTransformers, employing *mGTE* model (Reimers and Gurevych, 2019; Zhang et al., 2024). Later, these vectors are used for few-shot candidates selection and text clustering.

As for this stage of model tuning, we used almost the whole training dataset, as our main goal was to expose the model to as much relevant data as possible rather than tuning to a specific downstream task. Due to the high computational cost of full model fine-tuning, we instead trained LoRA adapter using a CLM objective. The adapter was configured with causal LM task type via the PEFT library to ensure compatibility with the CLM setup. Finally, we got the fine-tuned adapter for the text generation in the form of a list of manipulation techniques.

In the second stage, we merged the LoRA adapter from the first stage with the base model, set the model to a multi-label classification mode, and trained an additional LoRA adapter. The input for this stage consisted of text samples and their corresponding technique labels.

In the third stage, we combined the probability outputs from the previous stage with a set of engineered meta-features to train a CatBoost model for multi-label classification on the same training set. The additional features include:

1. distances from each text to the centroids of clusters formed by triggered phrases from the training set using K-means;
2. frequency of each manipulation technique among the most similar examples from the training set selected based on cosine similarity with their text and trigger phrases;
3. additional meta-features such as word count, number of question marks, presence of URLs, etc.

To construct the clustering-based features, we applied the K-means clustering algorithm to the set of triggered phrases extracted from the training set. Firstly, we encode the text with SentenceTransformers as mentioned earlier. We set the number of clusters (K) to be K=10, equal to the number of unique manipulation techniques. Finally, for each sample text, we calculate the cosine distance to the

³<https://huggingface.co/google/gemma-2-2b-it>

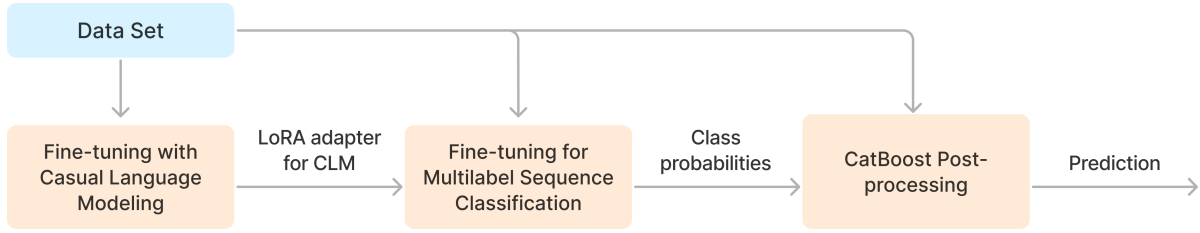


Figure 3: Pipeline of technique classification solution

centroid of each cluster. This approach allows the model to capture how semantically close a text is to common manipulation patterns identified in the training data.

For the similarity-based frequency features, we computed pairwise cosine similarity between the embedded texts. For each text, we selected two sets of 10 most similar examples from the training set: (1) based on overall similarity to other full texts, and (2) based on similarity to trigger phrases from other texts. We calculated the frequency distribution of manipulation techniques among the nearest neighbors in both cases. These techniques and meta-linguistic features (e.g., word count, presence of punctuation) were combined with model probabilities to train the final CatBoost classifier.

Finally, since the dataset is highly imbalanced, we optimized class-wise thresholds by performing k-fold cross-validation and choosing the median of the best thresholds within folds for each class separately. This approach avoids the pitfalls of using a single global threshold, especially for rare classes, and improves overall performance on the macro F1 score, which treats all classes equally. So, we used this method to construct the final prediction using the probability scores from the CatBoost model.

4.2 Span Identification

Span identification for manipulative content is defined as a binary token classification task, where each token is labeled as either manipulative or non-manipulative, independent of the specific manipulation technique. Identified manipulative tokens are then mapped to character indices and grouped into spans, allowing for precise extraction of manipulative text.

For this task, we employ a multi-headed architecture based on the XLM-RoBERTa-Large⁴ (see Figure 4). Two custom classification heads are introduced: one dedicated to classifying manipula-

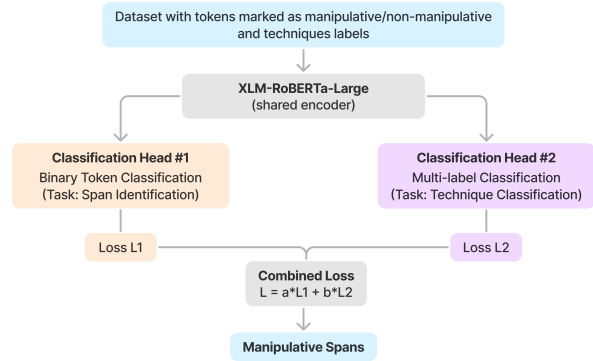


Figure 4: Pipeline of span identification solution

tive techniques (multi-label classification) and the other to token classification. Both heads share a common encoder, allowing the model to benefit from shared representations across tasks.

The span identification head consists of a single linear layer applied to the contextualized token representations, predicting the likelihood of each token being part of a manipulative span.

The technique classification head operates on a pooled representation formed by concatenating the *[CLS]* token embedding, mean-pooled, and max-pooled token embeddings. This concatenated vector is passed through a linear layer that projects it to a lower-dimensional space of size 256, followed by a GELU activation. The intermediate representation is then regularized through layer normalization and dropout before being passed to a final linear layer that projects it to the space of manipulation technique labels.

To balance the influence of both tasks during training, we apply a reduced weighting coefficient to the classification head’s loss when computing the overall objective. This ensures that span detection remains the primary focus, while the model still benefits from auxiliary guidance.

Consistent with Technique Classification Sub-task, we determine optimal prediction thresholds through k-fold cross-validation, ensuring robust calibration and generalization across splits.

⁴<https://huggingface.co/FacebookAI/xlm-roberta-large>

5 Evaluation

5.1 Technique Classification

The manipulation techniques classification subtask, as defined in the shared task, uses a macro-averaged F1 score as its primary evaluation metric. This metric treats all classes equally, regardless of their frequency in the dataset. Appendix B.1 provides a detailed explanation of the metric.

Our main results are summarized in Table 2, where F1 scores were recalculated on the full testing set. As a baseline, we used a multi-label CatBoost model with threshold optimization. For baseline training, we use a dataset that consists only of meta-features used in the final step, as explained in Section 4.1.

Although the baseline appeared to be an effective solution regarding resource efficiency and performance, it was insufficient to remain competitive in the challenge. This motivated integrating the Gemma 2-based solution, as introduced in Section 4.1. In our final comparison, we present two configurations of this model—with and without final post-processing using CatBoost and metafeatures. The results demonstrate that Gemma-based solutions significantly outperform the baseline. Although the post-processing step results in only a minor improvement, it is essential to achieve a competitive advantage in the competition.

We also conducted a performance analysis for each class (see Table 1), revealing considerable variation in the model’s effectiveness across different techniques. Notably, the model performs significantly worse on underrepresented classes such as *whataboutism*, *straw_man*, and *bandwagon*. In contrast, it achieves the highest performance on the *loaded_language* class, which has over ten times more samples than the mentioned underrepresented ones.

5.2 Span Identification

Like the previous subtask, span identification relies on the evaluation metrics defined in the shared task. Specifically, we use the span-level F1-score, quantifying the overlap between predicted and defined character spans. Appendix B.2 provides a detailed explanation of this metric.

Our span detection pipeline also incorporates post-processing and a threshold selection step, as described in Section 4.2. As a strong baseline, we employed the XLM-RoBERTa model configured for token classification. Building on top of it, we ex-

Technique	F1 score	Support
appeal_to_fear	0.450	449
bandwagon	0.215	236
cherry_picking	0.467	768
cliche	0.328	695
euphoria	0.550	695
fud	0.525	576
glittering_generalities	0.644	723
loaded_language	0.782	2959
straw_man	0.287	207
whataboutism	0.296	235

Table 1: Classification report for technique prediction

Solution	F1 macro
Baseline (CatBoost)	0.40801
Gemma	0.45007
Gemma with post-processing	0.45447

Table 2: Comparison of our solutions for technique classification during the competition

plored the hypothesis that a two-head transformer, combined to address both subtasks simultaneously, could enhance generalization and improve results. Although, as shown in Table 4, the performance gain was not large. This approach ultimately secured us third place in the competition, as reported in Table 5. These findings suggest that, for practical applications, a simpler baseline approach may be more robust and justified.

6 Conclusion

To sum up, this paper presents a competitive solution to the UNLP 2025 Shared Task on detecting manipulative narratives in Ukrainian Telegram news. By leveraging a multi-stage fine-tuned Gemma 2 language model with LoRA adapters for technique classification and a two-headed XLM-RoBERTa architecture for span detection, our approach secured second and third place in the respective subtasks.

Key achievements include a two-phase fine-

Team	Public	Private
GA	0.47369	0.49439
MolodiAmbitni	0.46203	0.46952
CVisBetter_SEU	0.43669	0.45519

Table 3: Comparison of metrics for top-3 solutions from competition leaderboard for manipulation classification

Solution	Span-level F1
Baseline	0.58588
Two-head transformer	0.59888

Table 4: Comparison of our solutions for span detection during the competition

Solution	Public	Private
GA	0.64598	0.64058
CVisBetter_SEU	0.59873	0.60456
MolodiAmbitni	0.59662	0.60001

Table 5: Comparison of metrics for top-3 solutions from competition leaderboard for span detection subtask

tuning of a decoder-only model (Gemma) for classification, first via causal language modeling, then supervised multi-label learning. We further enhanced performance with a post-processing step using a CatBoost classifier that combined meta-features with previously predicted class probabilities. Per-class threshold optimization addressed label imbalance and improved macro-F1 performance. For span detection, we introduced a dual-head architecture that jointly learned classification and token-level labeling, encouraging better generalization through shared representations.

Results show that each enhancement added measurable value. Post-processing raised the classification macro-F1 from 0.45007 to 0.45447, while span detection improved from 0.58588 to 0.59888 with the dual-head setup. However, performance varied notably across manipulation types: while frequent classes like *loaded_language* were predicted with high accuracy, rarer classes such as *whataboutism* and *straw_man* remained challenging.

Limitations

We are working with a dataset that includes texts only in Ukrainian and Russian. While LLMs are improving multilingual support, existing open-source models have limited support for those languages. Also, Telegram posts often contain informal language, slang, neologisms, emojis, and irregular formatting. It may reduce the effectiveness of pre-trained models, which are typically trained on more formal text.

While the dataset was annotated by experienced professionals, the manipulation signal is subjective and context-dependent. This can lead to ambiguous labels, especially in span identification, where the boundaries of manipulative content are not always

clearly defined.

Moreover, the dominance of certain manipulation techniques (e.g., loaded language) makes the classification task imbalanced. Although steps can be taken to mitigate this (e.g., resampling, class weighting, or threshold selection in our case), performance on rare techniques remains a challenge.

The dataset presented for the competition appears to be divided into training and test sets without considering the chronological order of posts. As a result, the evaluation may not reflect the real-world scenario of predicting new, emerging manipulation patterns.

Acknowledgments

We thank the Applied Sciences Faculty at Ukrainian Catholic University for providing access to computational resources that supported this research.

References

- Ola Altit, Malak Abdullah, and Rasha Obiedat. 2020. [JUST at SemEval-2020 task 11: Detecting propaganda techniques using BERT pre-trained model](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1749–1755, Barcelona (online). International Committee for Computational Linguistics.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [Llm2vec: Large language models are secretly powerful text encoders](#). *Preprint*, arXiv:2404.05961.
- Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. [SemEval-2020 task 11: Detection of propaganda techniques in news articles](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.
- Artur Kiulian, Anton Polishko, Mykola Khandoga, Oryna Chubych, Jack Connor, Raghav Ravishankar, and Adarsh Shirawalmath. 2024. [From bytes to borsch: Fine-tuning gemma and mistral for the ukrainian language representation](#). *Preprint*, arXiv:2404.09138.
- Rahul Mehta and Vasudeva Varma. 2023. [LLM-RM at SemEval-2023 task 2: Multilingual complex NER using XLM-RoBERTa](#). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 453–456, Toronto, Canada. Association for Computational Linguistics.

- 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*. Association for Computational Linguistics.
- Siva Sai, Alfred W. Jacob, Sakshi Kalra, and Yashvardhan Sharma. 2021. Stacked embeddings and multiple fine-tuned xlm-roberta models for enhanced hostility identification. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, pages 224–235, Cham. Springer International Publishing.
- Veronika Solopova, Christoph Benzmüller, and Tim Landgraf. 2023. [The evolution of pro-kremlin propaganda from a machine learning and linguistics perspective](#). In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 40–48, Dubrovnik, Croatia. Association for Computational Linguistics.
- Kilian Sprenkamp, Daniel Gordon Jones, and Liudmila Zavolokina. 2023. [Large language models for propaganda detection](#). *Preprint*, arXiv:2310.06422.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshov, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidson, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofiz Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kupala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshtir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. [Gemma 2: Improving open language models at a practical size](#). *Preprint*, arXiv:2408.00118.
- Mykola Trokhymovych, Oleksandr Kosovan, Nathan Forrester, Pablo Aragón, Diego Saez-Trumper, and Ricardo Baeza-Yates. 2025. [Characterizing knowledge manipulation in a russian wikipedia fork](#). *Preprint*, arXiv:2504.10663.
- Mykola Trokhymovych and Diego Saez-Trumper. 2021. [Wikicheck: An end-to-end open source automatic fact-checking api based on wikipedia](#). In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM '21*, page 4155–4164, New York, NY, USA. Association for Computing Machinery.
- Mykola Trokhymovych, Indira Sen, and Martin Gerlach. 2024. [An open multilingual system for scoring readability of Wikipedia](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6296–6311, Bangkok, Thailand. Association for Computational Linguistics.
- Mykyta Vorobiov. 2024. [Has ukraine become too dependent on telegram?](#) Accessed: 12 April 2025.
- Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. 2024. [mgte: Generalized long-context text representation and reranking models for multilingual text retrieval](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

A Manipulation Techniques

Table 6 contains each class explanation that was provided by the organisers.⁵

B Metrics

B.1 Techniques Classification

To evaluate the classification of manipulation techniques, we use the macro-averaged F1 score, which ensures balanced assessment across all techniques. Given a set of texts V and manipulation techniques T , each text is labeled with a binary vector indicating the presence of techniques. The model predicts a vector of the same size, and for each technique $t \in T$, we compute the F1 score:

$$F1_t = \frac{2 \cdot P_t \cdot R_t}{P_t + R_t}$$

where precision P_t measures correct predictions among all predicted instances, and recall R_t measures correct predictions among actual instances. The final macro-F1 score is obtained as:

$$F1_{\text{macro}} = \frac{1}{|T|} \sum_{t \in T} F1_t$$

This approach is particularly useful for handling class imbalances as it prevents frequently occurring techniques, which are typically detected with greater accuracy, from dominating the overall performance score.

B.2 Span Identification

To evaluate the accuracy of detected spans, we use the span-level F1 score, which measures the overlap between predicted and actual spans. Let V be the set of all texts in the dataset. Each text $v \in V$ has a set of ground truth spans S_v and predicted spans \hat{S}_v . The set of manipulated tokens in text v is defined as the collection of all characters whose index falls in at least one manipulation span:

$$T_v = \bigcup_{(s,e) \in S_v} \{s, s+1, \dots, e-1\}$$

$$\hat{T}_v = \bigcup_{(s,e) \in \hat{S}_v} \{s, s+1, \dots, e-1\}$$

Precision and recall are computed as:

$$P = \frac{\sum_{v \in V} |T_v \cap \hat{T}_v|}{\sum_{v \in V} |\hat{T}_v|}$$

$$R = \frac{\sum_{v \in V} |T_v \cap \hat{T}_v|}{\sum_{v \in V} |T_v|}$$

The final span-level F1 score is given by:

$$F1 = \frac{2PR}{P + R}$$

⁵<https://github.com/unlp-workshop/unlp-2025-shared-task/blob/main/data/techniques-en.md>

Name	Description
Loaded Language	The use of words and phrases with a strong emotional connotation (positive or negative) to influence the audience.
Glittering Generalities	Exploitation of people’s positive attitude towards abstract concepts such as “justice,” “freedom,” “democracy,” “patriotism,” “peace,” “happiness,” “love,” “truth,” “order,” etc. These words and phrases are intended to provoke strong emotional reactions and feelings of solidarity without providing specific information or arguments.
Euphoria	Using an event that causes euphoria or a feeling of happiness, or a positive event to boost morale. This manipulation is often used to mobilize the population.
Appeal to Fear	The misuse of fear (often based on stereotypes or prejudices) to support a particular proposal.
FUD (Fear, Uncertainty, Doubt)	Presenting information in a way that sows uncertainty and doubt, causing fear. This technique is a subtype of the appeal to fear.
Bandwagon/Appeal to People	An attempt to persuade the audience to join and take action because “others are doing the same thing.”
Thought-Terminating Cliché	Commonly used phrases that mitigate cognitive dissonance and block critical thinking.
Whataboutism	Discrediting the opponent’s position by accusing them of hypocrisy without directly refuting their arguments.
Cherry Picking	Selective use of data or facts that support a hypothesis while ignoring counter-arguments.
Straw Man	Distorting the opponent’s position by replacing it with a weaker or outwardly similar one and refuting it instead.

Table 6: Explanation of Manipulation Techniques provided by UNLP Shared Task