KostasThesis2025 at SemEval-2025 Task 10: Multilingual Characterization and Extraction of Narratives from Online News

Konstantinos Eleftheriou[1] **Panos Louridas**[2] **John Pavlopoulos**[1]
eleftheriou.konst@gmail.com louridas@aueb.gr annis@aueb.gr

[1]Department of Informatics, AUEB, Greece
[2]Department of Management Science and Technology, AUEB, Greece

Abstract

In response to the growing challenge of propaganda through online media in online news, the increasing need for automated systems that can identify and classify narrative structures in multiple languages is evident. We present our approach to the SemEval-2025 Task 10 Subtask 2, focusing on the challenge of hierarchical multi-label, multi-class classification in multilingual news articles. Working with a two-level taxonomy of narratives and subnarratives in the Ukraine-Russia War and Climate Change domain, we present methods to handle long articles based on how they are naturally structured in the dataset, propose a hierarchical classification MLP with respect to the narrative taxonomy structure, and establish a continual learning training strategy that takes into advantage the multilingual nature of our data and tries to examine how different language orders affect performance. Our final system was evaluated in five languages, achieving competitive results while demonstrating low variance compared to similar systems in our leaderboard position.

1 Introduction

From early days, propaganda has been a tool in shaping people's beliefs, actions, and behaviours. The most effective propaganda techniques often go undetected, influencing readers without even their knowledge (Muller, 2018). With the rapid growth of the Internet and the Web revolutionizing the way people share and access information, it has also opened doors to propagandistic techniques being disseminated more effectively, reaching vast audiences worldwide (Tardáguila et al., 2018).

At research level, most work on propaganda detection has focused on high-resource languages like English, and little effort has been made for low-resource languages. Similarly, previous work examined content at the document level (Rashkin et al., 2017), where they work focused on analyzing entire articles to differentiate between propaganda, trusted news, and satire rather than analysing specific narrative structures. SemEval-2020 Task 11, which focused on propaganda and news analysis, was introduced to address this (Da San Martino et al., 2020) featuring the classification of portions of documents across 44 propagandistic techniques.

SemEval-2025 Task 10[1] (Piskorski et al., 2025; Stefanovitch et al., 2025) was introduced as a significant advancement that focuses on the automatic identification of specific narrative structures, their classification, and the roles of entities involved in online articles in a multilingual setting.

This study focuses on the Narrative Classification subtask of SemEval-2025 Task 10. Unlike previous tasks, it centers around the identification of both the broader narratives of articles and their specific subnarratives. In this paper, we explore how hierarchical MLPs can model this nested taxonomy structure, investigate methods for handling long article inputs, and examine how different language orders can affect model performance in a continual learning training strategy.

Researchers have studied whether language order affects catastrophic forgetting in continual learning, but optimal order could vary across tasks and language sets. Our research builds on their findings, attempting to address the following research question: "Is there an optimal language order in language-specific continual learning for narrative classification? If so, which is the best and which is the worst?"

During our participation in the challenge, our primary approach, consisting of an ensembled version of a continual learning training strategy was evaluated in five languages achieving top-five rankings in across all languages[2]. Our analysis revealed that the order in which our model is trained matters

[1]https://propaganda.math.unipd.it/semeval2025task10/index.html
[2]https://propaganda.math.unipd.it/semeval2025task10/leaderboardv2.html

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 899–908

significantly in model performance, with certain language orders outperforming others.

Reproducibility

The complete codebase for this work, including data preprocessing, model architectures, and training methodologies, is available as documented Jupyter notebooks in our public GitHub repository.

1.1 Related Work

1.1.1 Related Tasks

Coan et al. (2021) present a classification task that focuses solely on contradictory claims of climate change in a similar hierarchical taxonomy. Their work emphasized on structuring claims into multiple levels of specificity following a parent-child relationship. Piskorski et al. (2022) builds upon climate change, by showcasing an effective and interesting way of handling limited training data using data augmentation techniques by maintaining the meaning intact, and thus also preserve label consistency. In a different but conceptually relevant domain, Kotseva et al. (2023) developed a hierarchical classification system for COVID-19 misinformation narratives, spanning over 58,000 articles in a similar multilingual setting.

1.1.2 Continual Learning in NLP

Continual learning (Wang et al., 2024) has gained attention in NLP for its ability to transfer knowledge across different tasks. The major challenge in this is catastrophic forgetting phenomenon, where a continually trained model tends to forget the knowledge it has learned (Kirkpatrick et al., 2017).

In the NLP area, continual learning has been established in different domains. Mi et al. (2020) demonstrates this with a dialogue system that is trained on sequential data. In a next token prediction task, Gogoulou et al. (2024) experimented with training a model in different languages using continual learning. That is, the model was trained first on a single language, then the training would continue with a different language, and so on. They discovered that the language order in which the model is trained, plays a crucial role – a carefully selected language order also seemed to reduce catastrophic forgetting.

1.2 Dataset

The dataset is composed of news articles in five languages: Bulgarian, Russian, Portuguese, English, and Hindi. These articles were collected and annotated specifically for the Ukraine-Russia War and Climate Change domains.

The data is divided into training (1,781 articles), development (178), and test (460) sets. The distribution of training data across languages is shown in Appendix Figure 5.

The articles vary significantly in length, a characteristic that introduces challenges we discuss in Section 1.3 and address in Subsection 2.1.

1.3 Challenges

The initial challenge resides in the way our labels are structured for classification. Each article can belong to one or more narratives that each map to one or more subnarratives, creating this two-level hierarchy. This presents challenges not only in hierarchical depth, since both levels must be predicted correctly, but also in managing the large number of possible labels. The scale of this can be better understood by looking at Figure 1, which shows a partial taxonomy of narratives and subnarratives for the Ukraine-Russia War domain.

Figure 1: Sample taxonomy for Ukraine-Russia War, showing the hierarchical relationship between narratives (inner ring) and their corresponding subnarratives (outer ring).

Cross-lingual Variations Working across multiple languages presents several specific challenges. Different languages tend to favor certain narrative patterns over others due to geopolitical factors. Appendix Figure 4 demonstrates this, with the Russian language focusing solely on the Ukraine-Russia War Taxonomy, while others exhibit a more balanced distribution between domains.

The combination of cross-lingual variations with limited training data poses data imbalance issues,

where certain narrative-subnarrative pairs appear much more frequently than others, something we discuss about in Subsection 2.3.

Article Length Variability Articles vary significantly in length, ranging from short to extensive (mean 403 words, std dev 237 words; between 88 to 924 words across languages).

Most (best) text classification models are specifically trained (or fine-tuned) to give good sentence embeddings; however, these models typically have a maximum token limit (usually 512 or 1024 tokens), which becomes problematic when processing large articles into representations that our classification models can then understand.

We carefully handle longer news articles in Section 2.1 to overcome a situation where article representation adversely affects the classification task.

2 System Overview

2.1 Article Representation

When articles are very long, most NLP work handles this by either including summarization preprocess step of the article into their pipeline (Tsirmpas et al., 2023), or paragraph splitting / hierarchical encoding (Dai et al., 2022).

We propose an alternative chunking approach, one that follows the natural structure of news articles in the dataset. Specifically, we observed that the articles consistently followed a header/body/footer organization, and we used this to perform a more targeted, semantically informed splitting.

However, combining the separated sections into a single embedding that describes the whole article is also something we need to address. We explored various strategies for doing so:

- Average pooling between sections: Average of all section embeddings, preserving each section equally.
- Weighted average based on section length: Similar to averaging, but sections contribute proportionally to their length.
- Sum of section embeddings: Element-wise addition of all section embeddings, essentially preserving all information.

2.2 Model Architecture

Initial experiments with simple classification models like logistic regression served as our first baselines by treating the problem as a flat classification without considering the label hierarchy. This approach revealed limited performance, and led us to explore approaches that could leverage this hierarchy.

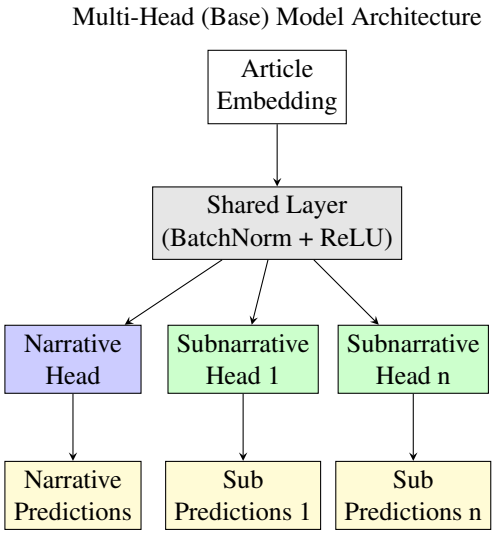

Figure 2: Architecture of the base multi-head model showing the flow from article embedding through shared layer to narrative and subnarrative heads.

However, the problem is structured in such a way that it differs from a two-head classification model, where we have a head for classifying narratives and a separate for subnarratives. Each narrative has its own set of subnarratives creating this natural hierarchy.

We developed a base multi-head, multi-task model approach where we have a single head for predicting narratives, then multiple heads for predicting the subnarratives for the given narrative hierarchy. We then explored several variants of this model as for experiments.

2.2.1 Multi-Head Base Architecture

Our base architecture (Figure 2) consists of three main components:

- A shared base layer that learns features and provides its output to the lower layers.
- A narrative head for predicting the top-level narratives.
- Multiple heads, one per narrative hierarchy, each predicting the corresponding subnarratives for that hierarchy.

2.2.2 Hierarchical Variants

Concatenation Model Our base model treated narrative and subnarrative predictions indepen-

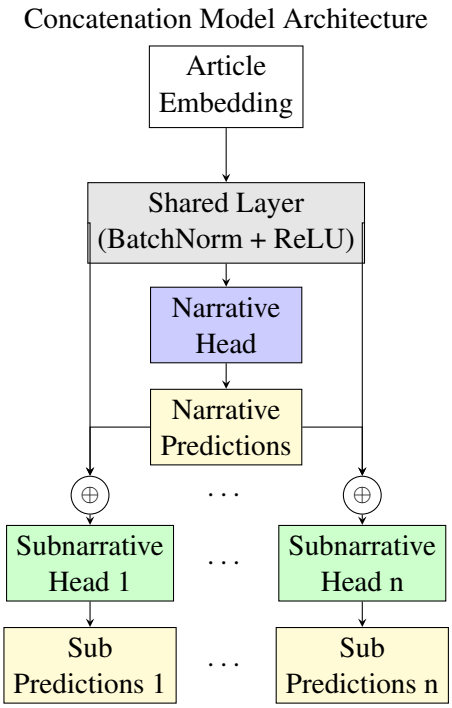

Figure 3: Architecture overview of the architecture for the concatenation model, showing how narrative predictions are combined with shared layer output to feed into subnarrative heads.

dently. That is, subnarrative predictions were computed as:

$$P(subnarr_j|x) = \sigma(h(x)) \quad (1)$$

where $h(x)$ is the output of the shared layer (the gray box) for article embedding x.

We enhanced this by concatenating the narrative probabilities with the shared layer output:

$$P(subnarr_j|x) = \sigma([h(x); P(narr_i|x)]) \quad (2)$$

where $narr_i$ is the parent narrative of $subnarr_j$.

This is intuitive, because:

- If the probability of the narrative is high, the subnarrative head will be more likely to predict the relevant subnarratives.
- If the probability is low, the model will learn to ignore the corresponding subnarratives.

Multiplication Model As an alternative to concatenation, we implemented element-wise multiplication between the output of the shared layer and the narrative probabilities.

$$P(subnarr_j|x) = \sigma(h(x) \odot P(narr_i|x)) \quad (3)$$

where $h(x)$ is the shared layer output for article embedding x.

This conceptually creates a stronger hierarchical dependency, acting as a natural "gate" in the hierarchy:

- If the narrative probability is close to 0, the corresponding subnarrative head's input will be scaled down, effectively disabling that subnarrative head.
- If the narrative probability is close to 1, the shared layer output passes through somewhat unaffected.

2.3 Loss Function

Our loss function is designed to handle both imbalanced labels and the need to stay consistent in our hierarchical predictions.

Weighted BCE We use a weighted version of Binary Cross Entropy to account for the class imbalance. Each label is assigned a weight that is proportional to its frequency in the dataset. This way, rare labels contribute proportionally more to the loss.

Hierarchy and Misclassifications We penalize inconsistencies in the hierarchy and label misclassifications. A complete loss break down is presented in Appendix A.2.

2.4 Continual Learning

Our initial experiments with the base architectures revealed significant performance instability across training runs (Section 3). This instability problem motivated us to try an alternative approach, one that changes the way the model learns from the training data resembling it in a way knowledge builds upon existing foundations.

Just as learning Ukrainian becomes easier when you know Russian, we hypothesized that this sequential order can help our model find meaningful patterns per language. In particular, for our problem:

- Russian language can provide a good base for the URW taxonomy.
- Bulgarian builds on top of Russian as both are Slavic languages.
- Every single language that follows keeps enriching the model's understanding with its unique characteristics.

Upon reaching our target language for classification during the training phase, we give the model more time to adapt by increasing its training patience and lowering the learning rate.

3 Experiments and Results

Below we present the comparison results across model variants, embedding models, and aggregation strategies. We report both Coarse-F1 (for narratives) and Fine-F1 (for subnarratives), along with their standard deviations. However, the primary focus of the task is on the Fine-F1 score.

All comparisons are performed specifically for the English validation dataset, as it demonstrated the most balanced distribution of narratives in the dataset across the two domains and is widely recognized as the most prominent language in NLP research.

Each model was run five times, and the results were aggregated to ensure a fair comparison. We evaluated our experiments with two embedding models: KaLM[3] and Stella[4]. We specifically chose these embedding models because they are both multilingual, instruction-based that achieved high performance on the MTEB (Massive Text Embedding Benchmark) leaderboard[5].

During the stage of transforming our article sections into meaningful numbers that our classification models can understand, we instructed the embedding models to:

"Produce an embedding useful for detecting relevant war- or climate-related narratives from a taxonomy."

3.1 Model Architecture and Embedding Performance

3.1.1 Hierarchical Architecture Variants

Table 1 shows the mean performance across model base variants. The high standard deviation (±0.02-0.03) indicates run-to-run instability.

Concat variant shows a sign of effectiveness in comparison to the Simple model by slightly outperforming it. Multiplication variant lags behind for both approaches, indicating that the hard-gating mechanism might be too restrictive. If our narrative predictions are not confident or even, and most importantly, not correct, the subnarrative head will receive very weak input because of the hard gating.

Metric	Simple	Concat	Mult
Coarse-F1	0.489 ± 0.03	0.497 ± 0.02	0.477 ± 0.02
Coarse std	0.385 ± 0.01	0.386 ± 0.01	0.384 ± 0.01
Fine-F1	0.329 ± 0.03	$\mathbf{0.333} \pm 0.02$	0.311 ± 0.02
Fine std	0.320 ± 0.02	0.327 ± 0.02	0.321 ± 0.01

Table 1: Mean performance comparison between the base hierarchical model and its variants (averaged over 5 runs).

3.1.2 Embedding Model Comparison

Table 2 shows performance between embedding models.

Metric	KaLM	Stella
Coarse-F1	0.497 ± 0.02	0.450 ± 0.02
Fine-F1	0.333 ± 0.02	0.298 ± 0.02

Table 2: Performance comparison across embedding models.

KaLM embeddings consistently appear to outperform Stella in all metrics.

However, when analyzing different aggregation strategies, our experiments revealed different patterns between embedding models: KaLM performed best with sum aggregation, while Stella showed superior results with weighted aggregation. A more in-depth analysis is presented in Appendix A.3.1.

3.1.3 Threshold Optimization

Our previous experiments tried to find the most optimal thresholds separately for narratives and subnarratives, exploring values up to 0.6. These thresholds determine the minimum probability for a narrative or subnarrative to be considered active in the predictions.

We later found out that the weighted aggregation strategy benefits significantly from increasing this threshold range up to 0.9, with the most noticeable improvement for Stella Embeddings. Detailed results and analysis can be found in Appendix A.3.2.

3.2 Continual Learning Performance

Table 3 shows the results between several language sequences and embedding combination strategies using the Concat hierarchical variant.

Impact of Aggregation Strategy At first glance, we see that the combination strategy is sensitive to

[3] https://huggingface.co/HIT-TMG/KaLM-embedding-multilingual-mini-instruct-v1.5

[4] https://huggingface.co/NovaSearch/stella_en_1.5B_v5

[5] https://huggingface.co/spaces/mteb/leaderboard

Order	Sum	Avg	W. Avg
RU→BG→PT→HI→EN	**0.378**	0.351	0.316
RU→BG→HI→PT→EN	0.356	0.323	0.341
BG→RU→PT→HI→EN	0.314	0.343	0.316
HI→PT→RU→BG→EN	0.302	0.312	0.330
PT→HI→RU→BG→EN	0.300	0.289	**0.352**
Ensemble of All Orders	0.350	0.349	**0.357**

Table 3: Impact of language ordering on Fine-F1 scores across different embedding combination strategies using Stella embeddings and 0.6 thresholds.

the language order:

- Sum strategy shows drastic response to the language ordering, with Fine-F1 scores ranging from 0.300 to 0.378.
- Mean strategy shows similar-to-moderate sensitivity, with Fine-F1 scores ranging from 0.289 to 0.351.
- Weighted average demonstrates the most balanced performance across orders, with Fine-F1 scores ranging from 0.316 to 0.357.

Specifically the weighted average strategy performs consistently better across different orders. In contrast to other strategies, it focuses on certain sections which might help the classification task, making thus the order less significant. However, when evaluating the effectiveness of a language order, we should primarily focus on the Sum and Avg strategies (which do not introduce any weighting). Both of these strategies agree that the first order produces the best results.

Impact of Language Order When evaluating for English data, the sequence that starts with Russian followed by Bulgarian outperforms every other sequence. Even swapping between these languages shows a performance drop. This suggests that when training the model with sequential data, starting with certain languages helps it build strong foundation patterns, strongly influencing final performance. In Appendix A.3.3 we do an in-depth order significance analysis.

Impact of Embedding Choice Interestingly, while KaLM embeddings outperformed Stella in our stand-alone experiments (Section 2), we observed different behavior in continual learning, with KaLM model under performing. This might suggest that Stella embeddings might be more appropriate in a knowledge transfer setup.

3.2.1 Threshold Optimization for Continual Learning

While we are at it, we extended our threshold optimization in Appendix A.3.2 to cover Continual Learning.

4 Discussion

4.1 Test Set Performance

For our final submission, we created an ensemble combining multiple models trained on different language orders, (where better performing language orders get more weight in the final prediction) using the Concat hierarchical variant. We positioned each target language, as the final stage of the learning sequence, which we give more patience and a lower learning rate.

The training configuration used Stella embeddings with a searching threshold of up to 0.6 and a sum aggregation strategy for section embeddings.

The results for our initial submission for the test set are presented in Table 4.

Lang	Rank	C-F1	std-C	F-F1	std-F
EN	16/30	0.409	0.314	0.239	0.243
PT	4/14	0.478	0.201	0.309	0.153
RU	6/15	0.596	0.257	0.333	0.234
BG	7/13	0.510	0.322	0.333	0.300
HI	6/14	0.384	0.418	0.282	0.402

Table 4: Version 1 of the leaderboard for the test set performance across the different languages.
C-F1: Coarse-F1, F-F1: Fine-F1, std-C/F: Standard deviation for coarse/fine metrics.

An important aspect of our results is stability. The proportion of F1 score and std is lower in comparison to teams near our entry. This shows a sign that our model is able to generalize and learn robust features. In comparison however to top teams, it's architecture might not be sufficient to capture more complex ones.

We conducted a brief post-competition analysis applying a higher threshold (0.9), which led to improved performance, particularly for English that is interesting to observe. Details and a comparison table are provided in Appendix A.3.4.

Limitations

Our approach used powerful pre-trained embeddings and a clear limitation is that we did not perform any fine-tuning on pre-trained models, something that was time and resource consuming for this

research. Top-performing teams likely used larger language models which offer greater performance but at higher computational costs. Our method provides some advantages in computational efficiency but the performance gap is evident. A promising direction would be to explore how incorporating larger models while maintaining our framework would respond to this new architecture.

Acknowledgments

This research was conducted as an undergraduate semester thesis project.

I would like to thank Panos Louridas and John Pavlopoulos for their guidance and AUEB for supporting this work.

References

Travis G. Coan, Constantine Boussalis, John Cook, and Mirjam O. Nanko. 2021. Computer-assisted classification of contrarian claims about climate change. *Scientific Reports*, 11(1):22320.

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.

Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. Revisiting transformer-based models for long document classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7212–7230, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. 2024. Continual learning under language shift. *arXiv preprint arXiv:2311.01200*. Accepted to TSD 2024, Correspondence: evangelia.gogoulou@ri.se.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*.

Bonka Kotseva, Irene Vianini, Nikolaos Nikolaidis, Nicolò Faggiani, Kristina Potapova, Caroline Gasparro, Yaniv Steiner, Jessica Scornavacche, Guillaume Jacquet, Vlad Dragu, Leonida della Rocca, Stefano Bucci, Aldo Podavini, and Jens P. Linge. 2023. Trend analysis of covid-19 mis/disinformation narratives–a 3-year study. *PLOS ONE*, 18(11).

Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual learning for natural language generation in task-oriented dialog systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3461–3474, Online. Association for Computational Linguistics.

Robert Muller. 2018. Indictment of internet research agency. pages 1–37. Public domain document, U.S. Government.

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.

Jakub Piskorski, Nikolaos Nikolaidis, Nicolas Stefanovitch, Bonka Kotseva, Irene Vianini, Sopho Kharazi, and Jens P. Linge. 2022. Exploring data augmentation for classification of climate change denial: Preliminary study. In *Proceedings of the Workshop on NLP for Climate Change (ClimateNLP 2022)*, volume 3117. CEUR Workshop Proceedings.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Nicolas Stefanovitch, Tarek Mahmoud, Nikolaos Nikolaidis, Jorge Alípio, Ricardo Campos, Dimitar Dimitrov, Purificação Silvano, Shivam Sharma, Roman Yangarber, Nuno Guimarães, Elisa Sartori, Ana Filipa Pacheco, Cecília Ortiz, Cláudia Couto, Glória Reis de Oliveira, Ari Gonçalves, Ivan Koychev, Ivo Moravski, Nicolo Faggiani, and 19 others. 2025. Multilingual characterization and extraction of narratives from online news: Annotation guidelines. Technical Report JRC141322, European Commission Joint Research Centre, Ispra, Italy.

Cristina Tardáguila, Fabrício Benevenuto, and Pablo Ortellado. 2018. Fake news is poisoning brazilian politics. whatsapp can stop it. *The New York Times*. Opinion.

Dimitrios Tsirmpas, Ioannis Gkionis, Georgios Th. Papadopoulos, and Ioannis Mademlis. 2023. Neural natural language processing for long texts: A survey on classification and summarization. *arXiv preprint arXiv:2305.16259*.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

A Appendix

A.1 Dataset Analysis

Figure 4 shows the complete distribution of domains across languages. As shown, Russian articles focus exclusively on the Ukraine-Russia War domain, while other languages show more balanced distribution between domains.

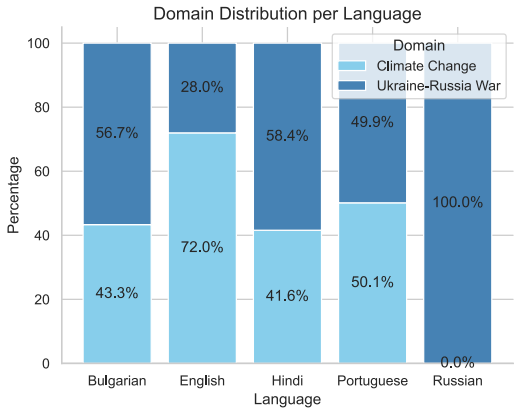

Figure 4: Distribution of domain across the five languages in the training set.

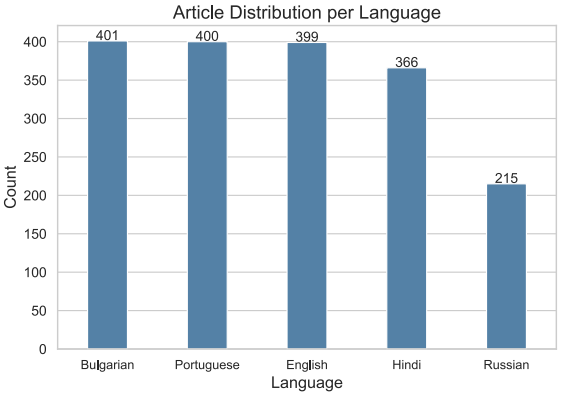

Figure 5: Distribution of articles across the five languages in the training dataset.

A.2 Loss Function Details

We penalize inconsistencies in the hierarchy and label miss-classifications. More specifically, the loss consists of:

$$\mathcal{L}_{\text{total}} = (1-W_{\text{sub}})\cdot\mathcal{L}_{\text{narr}}+W_{\text{sub}}\cdot\mathcal{L}_{\text{sub}}+W_{\text{cond}}\cdot\mathcal{L}_{\text{cond}} \quad (4)$$

$\mathcal{L}_{\text{narr}}$ represents the weighted BCE loss for narrative predictions, while $\mathcal{L}_{\text{sub}}$ captures the weighted BCE loss for subnarrative predictions. The term $\mathcal{L}_{\text{cond}}$ serves as a conditioning term that enforces hierarchical relationships.

The conditioning term enforces the hierarchical structure through:

$$\mathcal{L}_{\text{cond}} = \text{mean}(|p_{\text{sub}}\cdot(1-p_{\text{narr}})|+p_{\text{narr}}\cdot|p_{\text{sub}}-y_{\text{sub}}|) \quad (5)$$

The first part ($|p_{\text{sub}} \cdot (1 - p_{\text{narr}})|$) penalizes the model for predicting subnarratives when their parent narrative is inactive. The remaining part ensures subnarrative predictions match ground truth when their parent narrative is active.

A.3 Experimental Analysis

A.3.1 Model Evaluation Across Embedding Types and Architectures

Tables 5 and 6 present Fine-F1 scores (our primary goal is to improve subnarrative classification, we limit this analysis to solely Fine-F1 scores for simplicity) across model variants and aggregation strategies per embedding model.

Model	Sum	Mean	Weighted
Simple	0.329 ± 0.03	0.285 ± 0.01	0.325 ± 0.02
Concat	0.333 ± 0.02	0.305 ± 0.01	0.300 ± 0.02
Mult	0.311 ± 0.02	0.287 ± 0.02	0.283 ± 0.01

Table 5: Fine-F1 scores for KaLM embeddings across model variants and aggregation strategies.

Model	Sum	Mean	Weighted
Simple	0.309 ± 0.01	0.259 ± 0.01	$\mathbf{0.343} \pm 0.01$
Concat	0.298 ± 0.02	0.256 ± 0.02	0.338 ± 0.02
Mult	0.260 ± 0.01	0.260 ± 0.01	0.327 ± 0.01

Table 6: Fine-F1 scores for Stella embeddings across model variants and aggregation strategies.

Sum aggregation strategy appears to perform best across all other strategies for the KaLM Embeddings. This shows that KaLM benefits from preserving all information.

On the other hand, the weighted strategy seems to suit well with Stella, consistently outperforming all other strategies.

A.3.2 Extended Threshold Analysis

Optimizing Classification Thresholds Table 7 presents results for model variants, weighted aggregation strategy and Stella embeddings after exploring for higher thresholds, up to 0.9.

The weighted aggregation strategy benefits from higher thresholds likely because it prioritizes certain sections based on length. Higher thresholds

Model	C-F1	F-F1	F-std
Simple	0.538 ± 0.021	$\mathbf{0.426} \pm 0.010$	0.375 ± 0.008
Concat	0.554 ± 0.025	$\mathbf{0.442} \pm 0.019$	0.375 ± 0.016
Mult	0.556 ± 0.014	$\mathbf{0.426} \pm 0.017$	0.362 ± 0.011

Table 7: Performance metrics for Stella embeddings with weighted aggregation with 0.9 threshold.
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

help filter out noise in these weighted sections, requiring greater confidence for predictions. This leads to fewer but more precise positive classifications.

In contrast, other aggregation strategies combined with different embedding models tend to produce higher variance in their results.

Threshold Optimization in Continual Learning Following our discovery that weighted aggregation benefits from higher thresholds, we applied this approach to our continual learning training method. Table 8 presents these results.

Language Order (Thresh)	C-F1	F-F1	F-std
RU→BG→PT→HI→EN (0.75/0.50)	**0.614**	**0.449**	0.349
RU→BG→HI→PT→EN (0.75/0.55)	0.608	0.437	0.352
RU→HI→PT→BG→EN (0.80/0.60)	0.600	0.444	0.359
BG→RU→PT→HI→EN (0.70/0.55)	0.575	0.404	0.364
PT→HI→RU→BG→EN (0.75/0.60)	0.586	0.424	0.359
HI→PT→RU→BG→EN (0.70/0.50)	0.561	0.376	0.371
Ensemble (0.75/0.60)	0.570	0.424	0.362

Table 8: Performance of continual learning models, 0.9 thresholds, using Stella embeddings with weighted aggregation.
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

Again, higher thresholds benefit the continual learning approach in all language orders. The optimized thresholds vary slightly between different language sequences (ranging from 0.70-0.80 for narratives and 0.50-0.60 for subnarratives), suggesting that language-specific patterns influence the optimal decision boundary.

A.3.3 Statistical Analysis of Language Order Effects

For testing the significance of language order, we performed 25 independent experiments (5 random data batches per language × 5 random seeds per order) to ensure stability and performed statistical significance for the theoretically best order, against the other variants.

Order	Fine	Coarse	p-value
RU→BG→PT→HI→EN	**.350** ± .017	.513 ± .013	6.89×10^{-5}
RU→BG→HI→PT→EN	.323 ± .022	.485 ± .020	.601
HI→PT→RU→BG→EN	.312 ± .005	.479 ± .007	.025
RU→HI→PT→BG→EN	.210 ± .016	.369 ± .027	1.45×10^{-23}
PT→HI→RU→BG→EN	.289 ± .011	.476 ± .011	1.17×10^{-7}

Table 9: Impact of language order on model performance across different article batches and random seeds for sum aggregation strategy.

Effects with Sum Aggregation Strategy The in-theory best sequence (RU→BG→PT→HI→EN) achieved the highest score for the Fine F1 score. The variant that starts with Bulgarian and follows Russian, led to a slight decrease in performance.

Our hypothesized worst language order (RU→HI→PT→BG→EN) gave poor performance, with a very small p-value (1.17e-07), meaning it's very unlikely this poor performance occurred by chance.

Overall, the results show that when trying to create a model for English data, having certain languages early on in the sequence tends to help the model perform better.

Effects with Weighted Aggregation Strategy While we are at it, we also did a thorough analysis for the weighted strategy, which outperformed the sum strategy.

Order	Fine	Coarse	p-value
RU→BG→PT→HI→EN	**.423** ± .006	.583 ± .020	.068
BG→RU→PT→HI→EN	.355 ± 0.034	.501 ± .015	1.10×10^{-9}
HI→PT→RU→BG→EN	.398 ± 0.014	.571 ± .021	9.17×10^{-6}
RU→HI→PT→BG→EN	**.440** ± .013	.611 ± .018	3.09×10^{-6}
PT→HI→RU→BG→EN	.405 ± 0.014	.576 ± .015	.0029

Table 10: Impact of language order using weighted average strategy across different article batches and random seeds.

Weighted strategy revealed different patterns compared to sum.

Both RU→BG→PT→HI→EN and RU→BG→HI→PT→EN orders maintain strong performance, their difference is not statistically significant (p = 0.068). Language order RU→HI→PT→BG→EN performs surprisingly well, better than our best order for sum strategy and contrasting with its poor performance under the same approach.

Impact of Aggregation Strategy on Language Order Sensitivity The weighted strategy appears to be more robust to order variations, showing gen-

erally higher performance across all orderings compared to sum strategy. This shows that embedding aggregation affects the importance of language order. Sum aggregation preserves all article information equally, making language order clear and much more significant. Weighted average weights sections by their length, it shows more balanced performance across different orders, making language order less significant to performance.

A.3.4 Post-competition Analysis

In our post-competition analysis, we applied our findings about weighted aggregation with higher thresholds (0.9) to the test set. This post-analysis showed a positive sign in our results, particularly for English (Table 11).

Lang	Rank	C-F1	std-C	F-F1	std-F
EN	5/27	0.556	0.396	0.362	0.370
PT	3/14	0.539	0.214	0.329	0.171
RU	5/15	0.571	0.344	0.400	0.283
BG	5/13	0.523	0.371	0.357	0.349
HI	5/14	0.453	0.441	0.341	0.456

Table 11: Version 2 leaderboard results for test set performance across languages.
C-F1: Coarse-F1, F-F1: Fine-F1, std-C/F: Standard deviation for coarse/fine metrics.

Tables 12 and 13 compare performance using the weighted aggregation strategy with different thresholds (0.6 vs 0.9).

Language	F1 samples	F1 std samples
EN	0.287	0.296
PT	0.329	0.171
HI	0.340	0.434
BG	0.355	0.311
RU	0.398	0.292

Table 12: Post submission comparison of test set performance using threshold 0.6 with weighted strategy and Stella Embeddings.

Language	F1 samples	F1 std samples
EN	**0.362**	0.370
PT	0.326	0.208
HI	0.341	0.450
BG	0.357	0.349
RU	0.400	0.283

Table 13: Post submission comparison of test set performance using threshold 0.9 with weighted strategy and Stella Embeddings.

The results show improvements in all languages when using the weighted strategy. The increased range of threshold values up to 0.9 proved significant for the English dataset. However, for the rest of the languages, having an increased threshold did not seem to contribute to better performance, with some languages even experiencing higher variance.