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

Konstantinos Eleftheriou
eleftheriou.konst@gmail.com

Panos Louridas
louridas@aueb.gr

John Pavlopoulos
annis@aueb.gr

Abstract

In response to the growing challenge of propagandistic presence through online media in online news, the increasing need for automated systems that are able to 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. We present methods to handle long articles with respect to how they are naturally structured in the dataset, propose a hierarchical classification neural network model with respect to the taxonomy, and a continual learning training approach that leverages cross-lingual knowledge transfer. Our system was evaluated across 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 behaviors. 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. The SemEval-2025 Task 10[1] was introduced to address this problem by focusing on the automatic identification of propaganda narratives, their classification and the roles of the entities involved in online articles in a multilingual setting.

We present the approach we followed for the Narrative Classification subtask, a hierarchical multi-label, multi-class classification problem fostering systems to classify news articles according to a predefined taxonomy of narratives and their corresponding subnarratives.

The task presents several challenges, thoroughly discussed in Section 1.3. Our approach is designed to systematically handle those by: (1) a decomposition of long articles that respects their natural structure to create effective representations, (2) a hierarchical neural network model respecting the narrative-subnarrative relationship, and (3), a continual learning training approach that suits the multilingual setting of our data and helps models build up foundational patterns before focusing on target languages for classification.

We achieved strong performance in multiple languages, with particularly notable results in Portuguese (4th), Russian (6th), and Hindi (6th). Our approach also showed strength in prediction stability, showing lower std compared to similar-ranking teams near our entries.

1.1 Related Work

Coan et al. (2021) presents a classification task that focuses solely on contrarian 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. It showcases an effective and interesting way of handling limited training data using data augmentation techniques by maintaining the meaning intact, and thus also preserving label consistency when trying to generate synthetic data.

In a different, but conceptually relevant domain, Kotseva et al. (2023) developed a hierarchical classification system for COVID-19 misinformation narratives, spanning more than 58,000 articles in a similar multilingual setting.

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 Figure 1.

The articles vary significantly in length, a characteristic that introduces challenges we discuss in

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

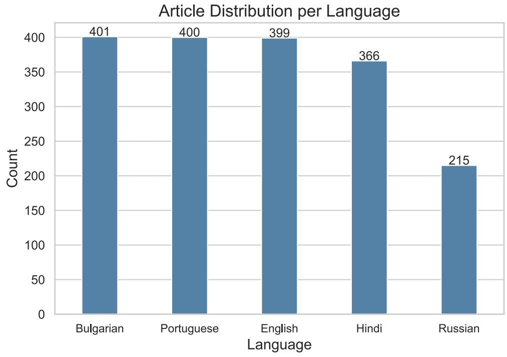

Figure 1: Distribution of articles across the five languages in the training dataset.

Section 1.3 and address in Subsection 2.1.

1.3 Challenges

The initial challenge lies in the way our labels are structured. The scale of this can be better understood by looking at Figure 2, which shows a sample of the taxonomy of narratives and subnarratives for the Ukraine-Russia War taxonomy.

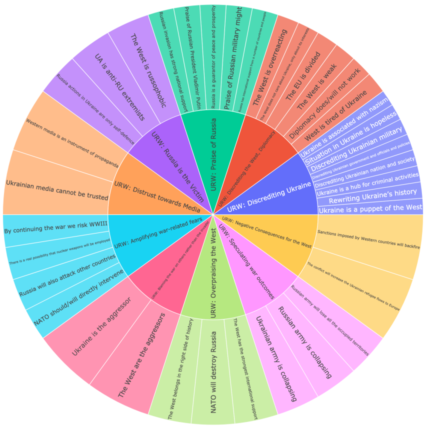

Figure 2: Sample taxonomy for Ukraine-Russia War, showing the hierarchical relationship between narratives (inner ring) and their corresponding subnarratives (outer ring).

Cross-lingual Variations: Working across multiple languages presents notable challenges; different languages tend to favor certain narrative patterns over others due to geopolitical factors. Appendix Figure 5 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 address in Subsection 2.3.

Article Length Variability: Articles vary significantly in length, ranging from short to extensive. Most (best) embedding models are specifically trained (or fine-tuned) to give good sentence embeddings, however, the limitation of tokens when processing news articles into representations that our models can understand is something to also be aware of.

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

2 System overview

2.1 Article Representation

We propose a chunking approach that follows the natural structure of news articles in the dataset.

Chunking Approach: Rather than employing an arbitrary paragraph-based splitting, we leveraged the inherent structure of articles, that is, their header/body/footer structure.

Combining the separated sections into a single embedding that describes the whole article is also something we ought to take care of.

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.

We analyze the impact of these strategies in Section 3.

2.2 Model Architecture

The problem itself 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 comparison experiments.

2.2.1 Multi-Head Base Architecture

Our base architecture consists of three main components:

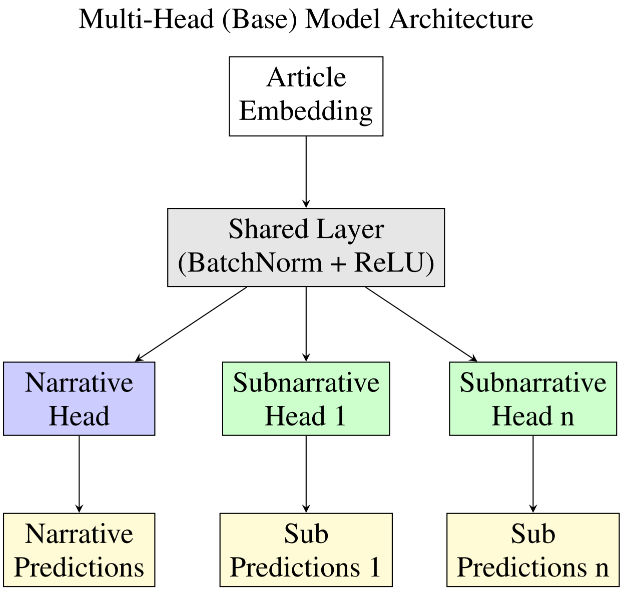

Figure 3: Architecture of the base multi-head model showing the flow from article embedding through shared layer to narrative and subnarrative heads.

- 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 independently. We enhanced the input each subnarrative head receives by concatenating the narrative probabilities with the shared layer output:

$$P(subnarr_j|x) = \sigma([h(x); P(narr_i|x)]) \quad (1)$$

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.
- At the same time, the shared output of the shared layer will help determine which subnarrative is most appropriate for the given article.

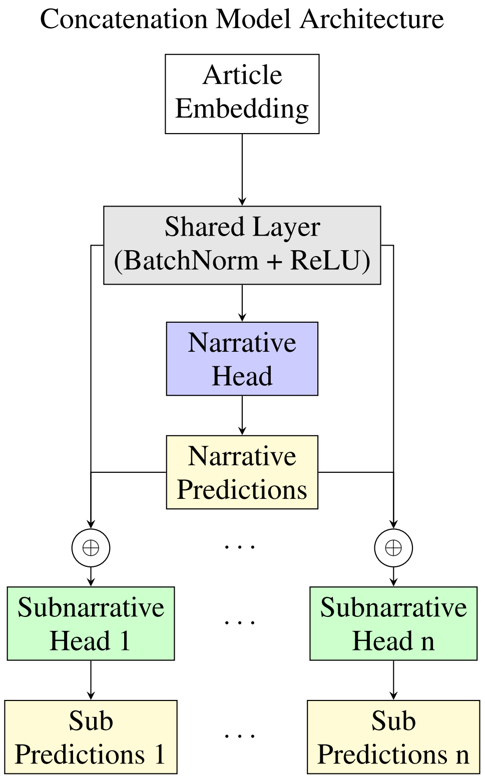

Figure 4: Architecture overview of the architecture for the concatenation model, showing how narrative predictions are combined with shared layer output to feed into subnarrative heads.

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

$$P(subnarr_j|x) = \sigma(h(x) \odot P(narr_i|x)) \quad (2)$$

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 BCE (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 Miss-classifications: We penalize inconsistencies in the hierarchy and label miss-classifications. A complete loss break down is presented in Appendix A.2.1.

2.4 Training Strategies

Our initial experiments with the base models revealed significant performance instability across training runs (Section 3).

This motivated us to explore the idea of ensembling, which has been shown to reduce variance and improve generalization in classification tasks. (Dietterich, 2000)

2.4.1 Continual Learning

For the training phase of our models, we tried a more sequential approach, matching closely with how we, humans, learn different concepts.

Just as learning Ukrainian becomes easier when you know Russian (by having similar grammar and vocabulary), 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 Slavic languages.
- Every single language that follows keeps enriching the model's understanding with its unique characteristics (Kirkpatrick et al., 2017).

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

We created an ensemble combining multiple models trained upon different order. During inference, better performing language orders get more weight in the final prediction.

3 Evaluation

Below we present comparison results between base model variants and training strategies. All comparison experiments are performed specifically for the English validation dataset. Each model was run five times. The results were aggregated to ensure a fair comparison.

We evaluated our experiments with two embedding models: KaLM[2] and Stella[3]. Both embedding models are instruction-based, and during the stage of transforming our sections into meaningful numbers that our models can understand, we instructed the models to:

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

3.1 Baseline Comparisons

Table 1 shows performance across model base variants.

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	0.333 ± 0.02	0.311 ± 0.02
Fine std	0.320 ± 0.02	0.327 ± 0.02	0.321 ± 0.01

Table 1: Performance comparison across architectures.

Standard deviations (±0.02-0.03) indicates notable run-to-run instability.

Concatenation 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.

Aggregation Strategy Comparisons: Our experiments revealed different patterns between embedding models: KaLM performed best with sum aggregation, while Stella showed superior results with weighted aggregation.

Detailed comparisons are presented in Appendix A.3.1.

Threshold Optimizations: Our previous experiments tried to find the most optimal thresholds separately for narratives and subnarratives, exploring values up to 0.6. We discovered that the weighted aggregation strategy benefits from higher thresholds (0.9), particularly visible for Stella Embeddings and the weighted strategy. Other strategies with also different embedding models seem to receive relatively higher variance.

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

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

Detailed results and analysis can be found in Appendix A.3.1.

3.2 Continual Learning

Table 2 shows the results between several language sequences and embedding combination strategies using the Concat variant.

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

Impact of Aggregation Strategy: Combination strategy shows sensitivity to the language order:

- Sum strategy reveals a drastic response to language ordering.
- Mean strategy has similar-to-moderate sensitivity.
- Weighted average demonstrates the most balanced performance across orders.

Specifically for the weighted average strategy, it specifically focuses on certain sections of articles which may help classification task, making order less significant.

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 leads to a notable performance drop. This suggests that when exposing the model with sequential data, starting with certain languages might help it build strong foundation patterns, strongly influencing final performance. In Appendix A.3.2 we do an in-depth order significance analysis.

Impact of Embedding Choice: Interestingly, while KaLM embeddings outperformed Stella in our standalone experiments, 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.

Threshold Optimization for Continual Learning: While we are at it, we also experimented with different thresholds in Appendix A.3.1, as we did for the base model and the variants.

4 Discussion

4.1 Test Set Performance

For the final submission, we used the ensemble version of the Continual Learning model with Concat as a base model. We positioned each target language, as the final stage of the learning sequence, which we give more patience and a lower learning rate. The results for test set are shown below:

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 3: Test set performance across languages.
C-F1: Coarse-F1, F-F1: Fine-F1, std-C/F: Standard deviation for coarse/fine metrics.

A notable aspect of our results is stability. The proportion of F1 score and std is lower in comparison to teams nearby 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 is not enough to capture more complex ones.

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

Previous experiments showcased that the weighted avg strategy with increased thresholds yielded better performance, at least in the validation set; however, this discovery occurred after our test submission deadline. Post-competition analysis revealed that it would have yielded slightly better results. Detailed results of this analysis are presented in Appendix A.3.3

4.2 Future Work

Embeddings: Our section-based approach opens opportunities for instruction-tuned embedding models. Different sections also contain different contexts, with headers as a more generic content for the article, and body adding supportive details - a future study can experiment with a more targeted processing per section with instruction-based embedding models where different instructions can be applied to different sections.

Architecture: As we noted previously, our models power is limited, since it is based on simple neural networks. One visible work would be to see how leveraging pre-trained models could enhance performance, and how continual learning would respond to this new architecture.

5 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.

A Appendix

A.1 Dataset Analysis

Figure 5 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.

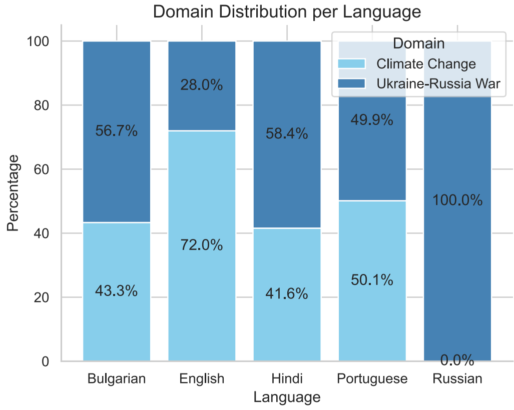

Figure 5: Distribution of domain across the five languages in the training set.

A.2 Model Details

A.2.1 Loss Function Details

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

$$\mathcal{L}_{\text{total}} = (1-W_{\text{sub}})\cdot\mathcal{L}_{\text{narr}} + W_{\text{sub}}\cdot\mathcal{L}_{\text{sub}} + W_{\text{cond}}\cdot\mathcal{L}_{\text{cond}} \tag{3}$$

$\mathcal{L}_{\text{narr}}$ 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:

$$\mathcal{L}_{\text{cond}} = \text{mean}(|p_{\text{sub}}\cdot(1-p_{\text{narr}})| + p_{\text{narr}}\cdot|p_{\text{sub}}-y_{\text{sub}}|) \tag{4}$$

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 Quantitative Evaluation

Aggregation Strategy Analysis Tables 4 and 5 present the scoring across architectures and aggregation strategies per embedding model.

Notably, sum aggregation appears to perform best across all architectures for the KaLM Embeddings. This shows that KaLM benefits from preserving all information.

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 4: Fine-F1 scores for KaLM embeddings across architectures and aggregation strategies.

Model	Sum	Mean	Weighted
Simple	0.309 ± 0.01	0.259 ± 0.01	**0.343** ± 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 5: Fine-F1 scores for Stella embeddings across architectures and aggregation strategies.

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

Threshold Analysis and Optimization **Base Model Threshold Optimization:** Table 6 presents results for model variants, weighted aggregation strategy and Stella embeddings after exploring for higher thresholds, up to 0.9.

Model	C-F1	F-F1	F-std
Simple	0.538 ± 0.021	**0.426** ± 0.010	0.375 ± 0.008
Concat	0.554 ± 0.025	**0.442** ± 0.019	0.375 ± 0.016
Mult	0.556 ± 0.014	**0.426** ± 0.017	0.362 ± 0.011

Table 6: Performance metrics for Stella embeddings with weighted aggregation with 0.9 threshold.
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

Continual Learning Threshold Optimization: Following our discovery that weighted aggregation benefits from higher thresholds, we applied this approach to our continual learning training method. Table 7 presents these results.

A.3.2 Language Order Analysis

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.

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 7: 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

Language Order Analysis with Sum Strategy: Table 8 shows the model performance for the sum strategy of our experiments.

The in-theory best sequence (RU→BG→PT→HI→EN) achieved the highest score for the Fine F1-score.

Order	Fine	Coarse	p-val
RU→BG→PT→HI→EN	**.352** ± .017	.513 ± .013	6.89e-05
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	1.45e-23
PT→HI→RU→BG→EN	.289 ± .011	.476 ± .011	1.17e-07

Table 8: Impact of language order on model performance across different article batches and random seeds for sum aggregation strategy.

The variant that starts with Bulgarian and follows Russian, led to a slight decrease in performance. The high p-value (approximately p = 0.6) suggests that the difference is not statistically significant, indicating that both orders work similarly well.

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 shows 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.

Language Order Analysis with Weighted Strategy: While we are at it, we also did a thorough analysis for the weighted strategy, that outperformed the sum strategy. Table 9 presents the results of this analysis.

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 (approx. p = 0.068).

Order	Fine	Coarse	p-val
RU→BG→PT→HI→EN	**.423** ± .006	.583 ± .020	.068
BG→RU→PT→HI→EN	.355 ± 0.034	.501 ± .015	1.10e-09
HI→PT→RU→BG→EN	.398 ± 0.014	.571 ± .021	9.17e-06
RU→HI→PT→BG→EN	**.440** ± .013	.611 ± .018	3.09e-06
PT→HI→RU→BG→EN	.405 ± 0.014	.576 ± .015	.0029

Table 9: Impact of language order using weighted average strategy across different article batches and random seeds.

Notably, 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. However, the weighted strategy appears to be more robust to order variations, showing generally higher performance across all orderings compared to sum strategy.

This shows that embedding aggregation affects the importance of language order. Sum aggregation preservers 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.3.3 Extended Results

Table 10 presents post-submission for the test set, when using the weighted strategy and searching for best thresholds up to 0.6 and 0.9.

	0.6		0.9	
Language	F1 samples	F1 std samples	F1 samples	F1 std samples
EN	0.287	0.296	**0.362**	0.370
PT	0.329	0.171	0.326	0.208
HI	0.340	0.434	0.341	0.450
BG	0.355	0.311	0.357	0.349
RU	0.398	0.292	0.400	0.283

Table 10: Post submission comparison of test set performance using threshold 0.6 vs threshold 0.9 limits with weighted strategy and Stella Embeddings.

The results show notable improvements across all languages when using weighted strategy. The increase 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.

References

Travis G Coan, Constantine Boussalis, John Cook, and Mirjam O Nanko. 2021. Computer-assisted classification of contrarian claims about climate change.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning.

James Kirkpatrick, Pascanu, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks.

Bonka Kotseva, Irene Vianini, and Nikolaos Nikolaidis. 2023. Analyzing covid-19 misinformation narratives across multilingual sources.

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.