ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Bachelor's Thesis
in
Information Technology

SemEval-2025 Task 10: A Continual Learning Approach to Propaganda Analysis in Online News

Konstantinos Eleftheriou, p3200283

Supervisor: Prof. John Pavlopoulos
Department of Informatics
Athens University of Economics and Business

Co-Supervisor: Prof. Panos Louridas
Department of Management Science and Technology
Athens University of Economics and Business

February 2025

Konstantinos Eleftheriou, p3200283

SemEval-2025 Task 10: A Continual Learning Approach to Propaganda Analysis in Online News

February 2025

Supervisor: Prof. John Pavlopoulos

Co-Supervisor: Prof. Panos Louridas

Athens University of Economics and Business

Department of Informatics

Athens, 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.

Acknowledgements

I would like to thank John Pavlopoulos and Panos Louridas for their guidance and support.

Contents

1 Introduction

1.1 Motivation and Problem Statement

From early days, propaganda has been a tool in shaping people's beliefs, actions, and behaviours. The most effective propaganda techniques often act 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). The large volume of online content expanding with more than 500 million tweets per day on Twitter (or X) (Sayce, 2025) by 2022, makes manual identification of propaganda impractical in the digital era, highlighting the need for automated tools and systems.

At research level, most work on propaganda detection has focused on high-resource languages, such as English, and little effort has been made to detect propaganda for low-resource languages. 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, Mahmoud, 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. It offers three subtasks on news articles: Entity Framing, Narrative Classification and Narrative Extraction.

This study focuses on the Narrative Classification subtask of SemEval-2025 Task 10. Unlike previous tasks, previously discussed, it centers around the identification of both the broader narratives of articles and their specific subnarratives. We explore how hierarchical neural networks 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.

We started by creating hierarchical neural networks (Section 3.2) motivated by the initial structure of narratives and subnarratives, and then explored variations of them, as for

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

experiments. We then proceeded to experiment with different training strategies (Section 3.3) that build upon these hierarchical models and tackle challenges from different angles. We focus on a continual/sequential learning approach that leverages the multilingual nature that our dataset presents. Researchers have studied whether language order affects catastrophic forgetting in continual learning (Subsection 2.3.2), but optimal order could vary across tasks and language sets. This thesis 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 with strong results in Portuguese (3rd), English (5th), Russian (5th), Hindi (5th) and Bulgarian (5th) out of on average 18 teams[2]. Our analysis revealed (Subsection 4.1.3) that the order in which our model is trained matters significantly in model performance, with certain language orders outperforming others. An important aspect of our approach was stability, showing lower variance in predictions compared to similarly ranked systems for certain languages.

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.

1.2 Thesis Structure

Chapter 2: Task Background – Introduction to the participated SemEval Task, discussion of dataset characteristics and challenges, and brief overview of related work in similar tasks.

Chapter 3: System overview – Outline of our proposed technique for handling article embeddings, in-depth description of the hierarchical model architectures developed, and overview of different training strategies tackling problem challenges from different angles.

Chapter 4: Evaluation – Comprehensive experimental results across different model variants, embedding strategies and training approaches, along with a statistical analysis focusing on showing the effectiveness of language order when training models.

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

Chapter 5: Results and Discussion – Discussion on test set performance across languages, strengths/weaknesses of our approach and future work.

Task and Background

2

2.1 Background

SemEval (Semantic Evaluation)[1] is a series of international natural language processing (NLP) research workshops whose mission is to advance the current state-of-the-art in semantic analysis and help create high-quality annotated datasets in a range of increasingly challenging problems in natural language semantics. Each year's workshop features a collection of shared tasks in which computational semantic analysis systems designed by different teams are presented and compared. Typically, SemEval tasks attract participation from academic institutions and industry research labs internationally, serving as important benchmarks for advancing NLP research.

2.2 Task Overview

The Narrative Classification subtask[2] of SemEval 2025 Task 10 consists of a hierarchical multi-label, multi-class classification problem fostering systems to classify news articles according to a predefined taxonomy[3]of narratives and their corresponding subnarratives. Table 2.1 shows an example classification for a Russia-Ukraine war article.

Narrative(s)	**Subnarrative(s)**
Discrediting Ukraine	Ukraine is associated with nazism, Discrediting Ukrainian military.
Praise of Russia	Praise of Russian military might.

Tab. 2.1: *Example classification of narratives and subnarratives for a Ukraine-Russian War article.*

Systems are then evaluated based on two primary metrics:

- **Fine-grained F1** Averaged samples F1 computed for complete narrative-subnarrative pair labels, where both the narrative and subnarrative parts must be correct for the predicted label to be considered correct.
- **Coarse-grained F1** Averaged samples F1 computed for narratives only, by ignoring the subnarrative part of the narrative:subnarrative pair.

[1] https://semeval.github.io/
[2] https://propaganda.math.unipd.it/semeval2025task10/index.html
[3] https://propaganda.math.unipd.it/semeval2025task10/NARRATIVE-TAXONOMIES.pdf

2.2.1 Dataset Characteristics

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 2.1.

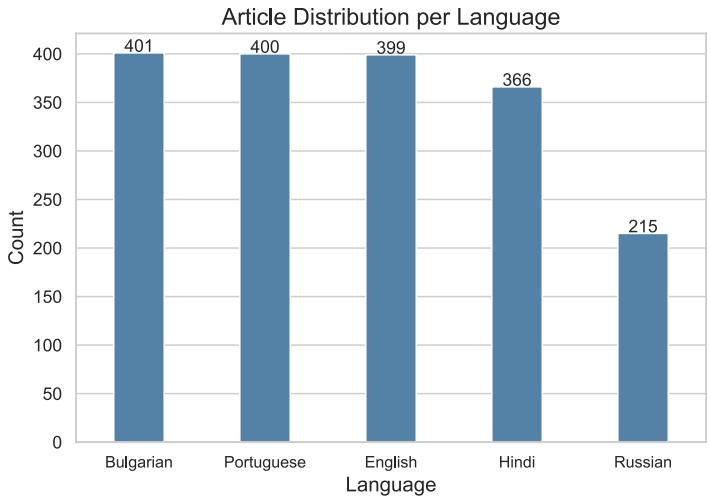

Fig. 2.1: *Distribution of articles across the five languages in the training dataset.*

The articles vary significantly in length, a characteristic that introduces challenges we discuss in Subsection 2.2.2 and address in Section 3.1.

2.2.2 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 2.2, which shows a partial taxonomy of narratives and subnarratives for the Ukraine-Russia War domain.

Cross-lingual Variations Working across multiple languages presents several specific challenges. Different languages tend to favour certain narrative patterns over others due to geopolitical factors. Figure 2.3 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

Fig. 2.2: *Partial taxonomy for the Ukraine-Russia War domain, illustrating the hierarchical relationship between narratives (inner ring) and their corresponding subnarratives (outer ring).*

frequently than others, something we address in Subsection 3.2.2.

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 3.1 to overcome a situation where article representation adversely affects the classification task.

2.3 Related Work

2.3.1 Related Tasks

Coan et al. (2021) presents 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, Nikolaidis, 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 preserve also label consistency, when trying

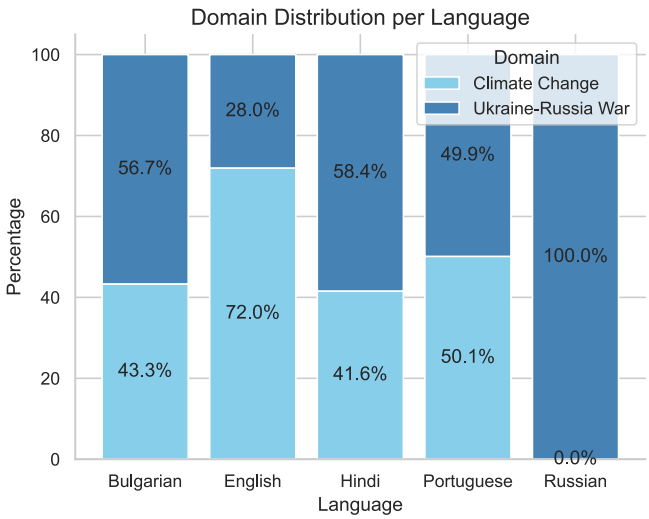

Fig. 2.3: *Distribution of domain across the five languages in the training set.*

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 over 58,000 articles in a similar multilingual setting.

2.3.2 Continual Learning in NLP

Continual learning has gained attention in NLP for its ability to transfer knowledge across different tasks. It is a known method in which input data is continuously used to extend the existing model's knowledge, i.e. to further train the model. The major challenge towards this goal is catastrophic forgetting, meaning that a continually trained model tends to forget the knowledge it has learned before (Wang et al., 2023; 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. Their approach overcame the catastrophic forgetting challenge by replaying important past examples. 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 showed to reduce catastrophic forgetting.

System Overview

3

3.1 Article Representation Strategy

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

Chunking Approach 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 (Table 3.1).

Section	Characteristics
Header	Short to medium length, introduces main topic
Primary Body	Extended content, core narrative
Secondary Body	Optional extended content
Tertiary Body	Optional medium to long content
Footer	Optional closing content, short to medium

Tab. 3.1: *Structural composition of news articles in the dataset.*

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:
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 Paragraph 4.1.2 and Subsection 4.1.3.

3.2 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.

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.

Multi-Head Base Architecture Our base architecture 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.

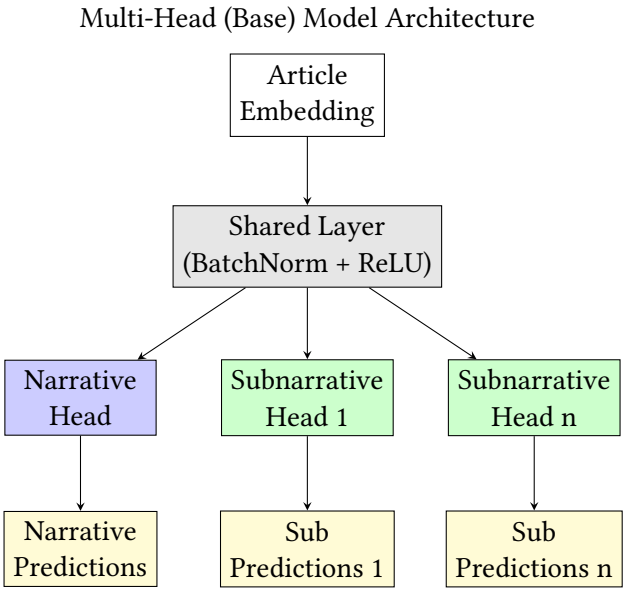

Fig. 3.1: *Architecture of the base multi-head model showing the flow from article embedding through shared layer to narrative and subnarrative heads.*

3.2.1 Hierarchical Variants

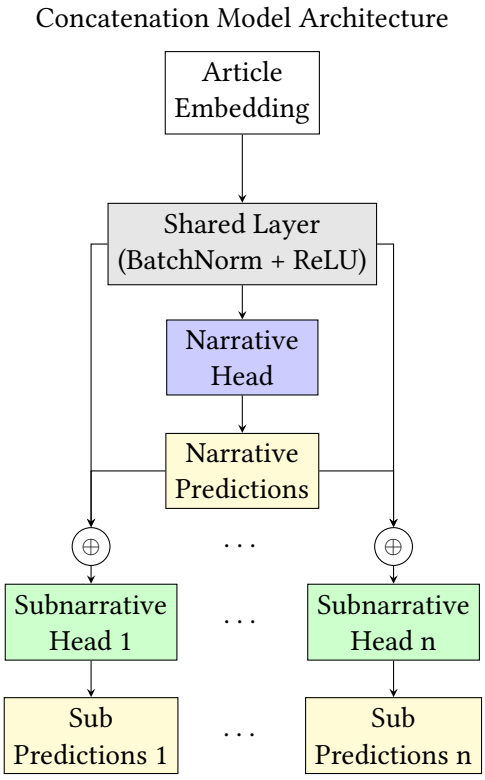

Fig. 3.2: *Architecture overview of the architecture for the concatenation model, showing how narrative predictions are combined with shared layer output to feed into subnarrative heads.*

Concatenation Model Our base model treated narrative and subnarrative predictions independently. That is, subnarrative predictions were computed as:

$$P(subnarr_j|x) = \sigma(h(x)) \tag{3.1}$$

where $h(x)$ the output of the shared layer (the gray box in Figure 3.1) for article embedding x. We enhanced this by concatenating the narrative probabilities with the shared layer output:

$$P(subnarr_j|x) = \sigma([h(x); P(narr_i|x)]) \tag{3.2}$$

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.

$$P(subnarr_j|x) = \sigma(h(x) \odot P(narr_i|x)) \tag{3.3}$$

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

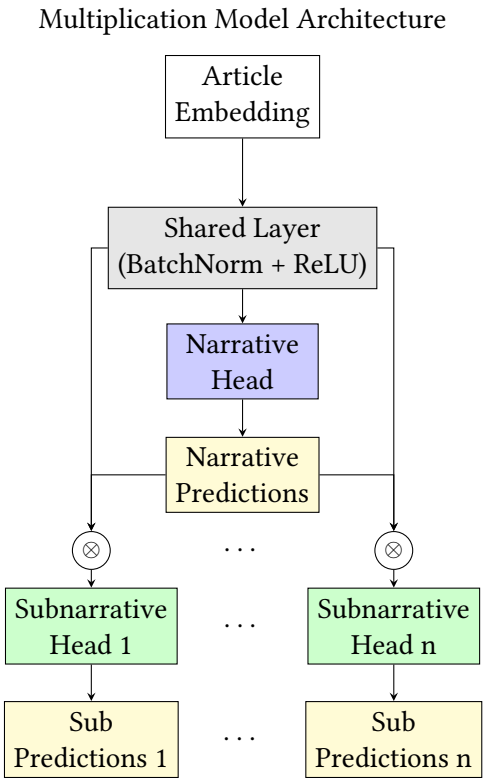

Fig. 3.3: *A simplified architecture overview of the multiplication model, showing how narrative predictions act as gates by multiplying with shared layer output before feeding into subnarrative heads.*

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.

Heatmap Classifier We also developed a simplified approach that views the problem as a 2D classification task. Instead of managing multiple prediction heads, this model directly outputs a probability matrix where:

- Rows (n) represent narratives.
- Columns (c) represent subnarratives.
- Each cell $H_{i,j}$ represents the probability of the narrative-subnarrative pair.

3.2.2 Loss Function

Our loss function is designed to handle both imbalanced labels and the need to remain 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. More specifically, the loss consists of:

$$\mathcal{L}_{\text{total}} = (1 - W_{\text{sub}}) \cdot \mathcal{L}_{\text{narr}} + W_{\text{sub}} \cdot \mathcal{L}_{\text{sub}} + W_{\text{cond}} \cdot \mathcal{L}_{\text{cond}} \tag{3.4}$$

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

$$\mathcal{L}_{\text{cond}} = \text{mean}(|p_{\text{sub}} \cdot (1 - p_{\text{narr}})| + p_{\text{narr}} \cdot |p_{\text{sub}} - y_{\text{sub}}|) \tag{3.5}$$

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 that the subnarrative predictions match ground truth when their parent narrative is active.

3.3 Training Strategies

Our initial experiments with the base architectures revealed significant performance instability across training runs (Section 4). This motivated us to explore the idea of ensembling, which has been shown to reduce variance and improve generalization in classification tasks.

In addition, we establish three training strategies, each one addressing the problem from a different angle.

3.3.1 N-fold Cross-Ensembling

We employed an n-fold cross ensembling approach that leverages our data more effectively. This training strategy splits the combined train and validation data into n folds and then trains n distinct models. Each model learns from a different subset of the data, thus capturing slightly different patterns. During inference, we average the predictions from all models to produce the final output.

3.3.2 Continual Learning

The instability problem motivated us to try an alternative approach, one that changes the way the model learns from the training data. 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 are Slavic languages.
- Every single language that follows keeps enriching the model's understanding with its unique characteristics.

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.

3.3.3 Checkpoint Ensembling

Checkpoint Ensembling has also proven an effective method (Chen et al., 2017). A checkpoint represents a saved model state during the training phase. Different checkpoints might be better at detecting different types of narratives.

Early stages of our model may be better at capturing certain patterns, while later stages may perform better on others after more training.
We follow a basic strategy for collecting checkpoints:

- Select the best-performing checkpoint based on the lowest validation loss achieved.
- Choose additional checkpoints evenly distributed across the training phase, to ensure diversity, as each checkpoint may learn different narrative/subnarrative patterns, thus improving generalization.

During inference, we consider the loss of each checkpoint as a factor. Each checkpoint's contribution is weighted inversely by its validation loss. The final predictions are calculated as the weighted average across all selected checkpoints.

Evaluation

4

4.1 System evaluation

4.1.1 Performance Analysis

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 recognised 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[1] and Stella[2]. 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[3]. 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."

4.1.2 Architecture Variants Comparisons

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

Tab. 4.1: *Mean performance comparison between the base hierarchical model and its variants (averaged over 5 runs).*

[1] https://huggingface.co/HIT-TMG/KaLM-embedding-multilingual-mini-instruct-v1.5
[2] https://huggingface.co/NovaSearch/stella_en_1.5B_v5
[3] https://huggingface.co/spaces/mteb/leaderboard

Table 4.1 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.

Embedding Model Comparisons Tables 4.2 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

Tab. 4.2: *Performance comparison across embedding models.*

KaLM embeddings consistently appear to outperform Stella in all metrics.

Aggregation Strategy Comparisons Tables 4.3 and 4.4 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
CK-Ens	**0.338**	0.335	0.312

Tab. 4.3: *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	**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
C-Ens	0.300	0.308	0.322

Tab. 4.4: *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.

Threshold Optimizations 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 discovered that the weighted aggregation strategy benefits from increasing the threshold range up to 0.9, with the most noticeable improvement for Stella Embeddings. Table 4.5 presents these results.

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
CK-Ens	0.566	0.410	0.343

Tab. 4.5: *Performance metrics for Stella embeddings with weighted aggregation and a threshold range up to 0.9.*
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

Other strategies with also different embedding models seem to receive relatively higher variance.

4.1.3 Continual Learning

Table 4.6 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	0.348	0.349	**0.357**

Tab. 4.6: *Impact of language ordering on Fine-F1 scores across different embedding combination strategies using Stella embeddings and a threshold range up to 0.6.*

Impact of Aggregation Strategy At first glance, we see that the combination strategy is sensitive to 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 (RU → BG → PT → HI → EN) 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.

Impact of Embedding Choice Interestingly, while KaLM embeddings outperformed Stella in our stand-alone experiments (Section 4.2), 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.

Threshold Optimization for Continual Learning Following our discovery that weighted aggregation strategy benefits from higher thresholds (Subsection 4.1.2), we applied the same approach to our continual learning training method. Table 4.7 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

Tab. 4.7: *Performance of continual learning models, using Stella embeddings with weighted aggregation strategy and a threshold range up to 0.9*
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

Higher thresholds (0.75/0.55 or 0.80/0.60) lead to better performance, with the RU→BG→PT→HI→EN sequence at 0.75/0.50 yielding the highest Fine-F1 score of 0.449.

In Subsection 4.1.6 we do an in-depth analysis to show order significance.

4.1.4 N-fold Cross-Ensembling

Table 4.8 shows the coarse and fine F1 scores across the Simple and Concat variants for each embedding model using the sum aggregation strategy approach. Table 4.9 presents the same results but with the weighted aggregation strategy approach.

Model	Threshold	C-F1	F-F1	F- std
Stella-Simple	0.60/0.55	0.506	0.385	0.378
Stella-Concat	0.60/0.55	0.499	0.390	0.371
KaLM-Simple	0.60/0.55	0.457	0.341	0.344
KaLM-Concat	0.50/0.45	0.503	0.373	0.346

Tab. 4.8: *Cross-validation ensemble performance with sum aggregation strategy with a threshold range up to 0.9.*
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

Model	Threshold	C-F1	F-F1	F-std
Stella-Simple	0.60/0.55	0.555	0.406	0.353
Stella-Concat	0.65/0.60	**0.560**	**0.419**	0.362
KaLM-Simple	0.55/0.30	0.518	0.354	0.338
KaLM-Concat	0.60/0.35	0.529	0.382	0.355

Tab. 4.9: *Cross-validation ensemble performance with weighted aggregation strategy with a threshold range up to 0.9*
C-F1: Coarse-F1, F-F1: Fine-F1, F-std: Fine-std

The weighted aggregation strategy shows strong performance compared to the sum approach across all model variants and embedding models. The results should be taken into account, knowing that the approach utilizes all available data. This approach also reduces overfitting to language-specific patterns by exposing each fold or model to different language distributions.

4.1.5 Heatmap Classifier

Table 4.10 presents the performance of our Heatmap classifier approach which is not multi-head based but follows a rather simpler architecture.
Like our base model and it's variants, the Heatmap classifier shows similar instability issues. Increasing threshold values up to 0.9 further increased variance substantially.

Configuration	Coarse-F1	Fine-F1	Fine std
KaLM-Sum	0.474 ± 0.016	0.340 ± 0.014	0.319
Stella-Sum	0.482 ± 0.011	0.329 ± 0.009	0.323
KaLM-Weighted	0.442 ± 0.017	0.299 ± 0.009	0.332
Stella-Weighted	0.450 ± 0.013	0.293 ± 0.018	0.335

Tab. 4.10: *Heatmap classifier performance with different embedding models and aggregation strategies with 0.6 thresholds.*

4.1.6 Qualitative 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.

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}

Tab. 4.11: *Impact of language order on model performance across different article batches and random seeds for sum aggregation strategy.*

Language Order Analysis with Sum 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.

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

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

Tab. 4.12: *Impact of language order using weighted average strategy across different article batches and random seeds.*

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 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.

Analysis and Discussion

5

5.1 Submission 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 the test set are shown in Table 5.1.

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

Tab. 5.1: *Test set performance across 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 is not enough to capture more complex ones.

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 (Table 5.2).

The results show improvements in 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.
Our updated positions for the test set are shown in Table 5.3.

	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

Tab. 5.2: *Post submission comparison of test set performance using threshold 0.6 vs threshold 0.9 limits with weighted strategy and Stella Embeddings.*

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

Tab. 5.3: *Updated test set performance across languages.*
C-F1: Coarse-F1, F-F1: Fine-F1, std-C/F: Standard deviation for coarse/fine metrics.

5.1.1 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.

Bibliography

Chen, Hugh, Scott Lundberg, and Su-In Lee (2017). "Checkpoint Ensembles: Ensemble Methods from a Single Training Process". In: *arXiv preprint arXiv:1710.03282*.

Coan, Travis G., Constantine Boussalis, John Cook, and Mirjam O. Nanko (2021). "Computer-assisted classification of contrarian claims about climate change". In: *Scientific Reports* 11.1, p. 22320.

Da San Martino, Giovanni, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov (Dec. 2020). "SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Ed. by Aurelie Herbelot, Xiaodan Zhu, Alexis Palmer, et al. Barcelona (online): International Committee for Computational Linguistics, pp. 1377–1414.

Dai, Xiang, Ilias Chalkidis, Sune Darkner, and Desmond Elliott (Dec. 2022). "Revisiting Transformer-based Models for Long Document Classification". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 7212–7230.

Gogoulou, Evangelia, Timothée Lesort, Magnus Boman, and Joakim Nivre (2024). "Continual Learning Under Language Shift". In: *arXiv preprint arXiv:2311.01200*. Accepted to TSD 2024, Correspondence: evangelia.gogoulou@ri.se.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, et al. (2017). "Overcoming catastrophic forgetting in neural networks". In: *arXiv preprint arXiv:1612.00796*.

Kotseva, Bonka, Irene Vianini, Nikolaos Nikolaidis, et al. (2023). "Trend analysis of COVID-19 mis/disinformation narratives–A 3-year study". In: *PLOS ONE* 18.11.

Mi, Fei, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings (Nov. 2020). "Continual Learning for Natural Language Generation in Task-oriented Dialog Systems". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 3461–3474.

Muller, Robert (2018). "Indictment of Internet Research Agency". In: Public domain document, U.S. Government, pp. 1–37.

Piskorski, Jakub, Tarek Mahmoud, Nikolaos Nikolaidis, et al. (July 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.

Piskorski, Jakub, Nikolaos Nikolaidis, Nicolas Stefanovitch, et al. (2022). "Exploring Data Augmentation for Classification of Climate Change Denial: Preliminary Study". In: *Proceedings of the Workshop on NLP for Climate Change (ClimateNLP 2022)*. Vol. 3117. CEUR Workshop Proceedings.

Rashkin, Hannah, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi (Sept. 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*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2931–2937.

Sayce, David (2025). *How Many Posts Are Published on X? (2025 Update)*.

Stefanovitch, Nicolas, Tarek Mahmoud, Nikolaos Nikolaidis, et al. (2025). *Multilingual Characterization and Extraction of Narratives from Online News: Annotation Guidelines*. Tech. rep. JRC141322. Ispra, Italy: European Commission Joint Research Centre.

Tardáguila, Cristina, Fabrício Benevenuto, and Pablo Ortellado (Oct. 2018). "Fake News Is Poisoning Brazilian Politics. WhatsApp Can Stop It". In: *The New York Times*. Opinion.

Tsirmpas, Dimitrios, Ioannis Gkionis, Georgios Th. Papadopoulos, and Ioannis Mademlis (2023). "Neural Natural Language Processing for Long Texts: A Survey on Classification and Summarization". In: *arXiv preprint arXiv:2305.16259*.

Wang, Liyuan, Xingxing Zhang, Hang Su, and Jun Zhu (2023). "A Comprehensive Survey of Continual Learning: Theory, Method and Application". In: *arXiv preprint arXiv:2302.00487*. Cited as: arXiv:2302.00487 [cs.LG].

List of Figures

List of Tables