Gradient Descenders at DeSegMa-IT: Leveraging Monolingual Transformer for LLM-Generated Text Detection and Boundary Identification

Tran Phuoc Thanh Nhan[1,2,*], Bui Hong Son[1,2] and Dang Van Thin[1,2]

[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

In this paper, we present the Gradient Descenders systems developed for the DeSegMa-IT shared task at EVALITA 2026. Our approach yielded highly competitive results: we achieved the 1st rank in Subtask A with an accuracy of 94.58% and secured the 3rd rank in Subtask B with a Mean Absolute Error (MAE) of 62.66. In the Machine-Generated Text (MGT) detection task, we demonstrated the effectiveness of UmBERTo, a RoBERTa-based model pre-trained on Italian text, validating the advantage of monolingual baselines over multilingual counterparts. In the Boundary Detection task, we approached the problem as a fine-grained token classification task using DeBERTa-base-Italian as the backbone encoder. Looking ahead, we aim to address the identified limitations regarding domain generalization and sequence dependency. Future work will focus on integrating Domain-Adversarial Neural Networks (DANN) to handle domain shifts and exploring more complex classification layers, such as Bi-LSTM, to further refine boundary localization precision.

Keywords

Machine-Generated Text Detection, Boundary Identification, Transformer Models, Italian NLP

1. Introduction

The rapid proliferation of Large Language Models (LLMs), such as GPT-4, Claude, and LlaMa [1], has fundamentally transformed the landscape of digital content creation. While these models offer unprecedented capabilities in generating coherent and contextually relevant text, they also introduce significant challenges regarding information integrity, academic honesty, and the potential for large-scale misinformation. As Machine-generated content becomes increasingly indistinguishable from human-written text, the need for robust, automated detection systems has never been more urgent.

To address these challenges within the Italian landscape, the DeSegMa-IT [2] shared task at EVALITA 2026 [3] has been established. The task focuses on the nuanced differentiation between human and machine authorship and is structured into two distinct but complementary subtasks:

- **Subtask A: MGT Detection in the Wild.** This subtask requires systems to determine whether a given text snippet was authored by a human or generated by an LLM.
- **Subtask B: Human-Machine Text Segmentation.** This subtask proposes human-machine mixed text detection, which provides a text where the first part is human-written and the second part is LLM-generated. The goal is to identify the exact point where the transition occurs.

In this paper, we describe our participation in both of the subtasks of the DeSegMa-IT Shared Task. Our core philosophy involves moving away from generic multilingual models to leverage language-specific adaptations that better capture the morphological richness of Italian. In the context of the DeSegMa-IT shared task, which focuses on Italian, using multilingual models such as XLM-RoBERTa [4] and mDeBERTa [5] may fail to capture the subtle linguistic nuances and morphological richness of

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
*Corresponding author.
24521241@gm.uit.edu.vn (T. P. T. Nhan); 22521246@gm.uit.edu.vn (B. H. Son); thindv@uit.edu.vn (D. V. Thin)
0009-0007-2395-3567 (T. P. T. Nhan); 0009-0006-7420-9212 (B. H. Son); 0000-0001-8340-1405 (D. V. Thin)

the language. Therefore, we propose a language-specific approach leveraging monolingual Italian pre-trained models.

For subtask A, we treat the problem as a sequence classification task. We employ UmBERTo [6], a RoBERTa-based model trained on the massive Italian component of the OSCAR corpus (Open Super-large Crawled ALMAnaCH coRpus). This choice is motivated by UmBERTo's superior ability to model Italian syntax and semantics compared to standard multilingual baselines, allowing for more accurate discrimination between human and machine stylistic patterns.

For subtask B, we approach the boundary detection challenge as a fine-grained Token classification problem. We utilize DeBERTa-base-Italian [7] as our backbone encoder. Technically, this model is built upon the mDeBERTa-v3-base [5] architecture. To focus on the Italian language, the backbone has been adapted to Italian by replacing the original embedding layer [8] and undergoing continued pre-training on Italian data. By attaching a linear classification head on top of this specialized backbone, our system predicts a binary label for every token, enabling it to pinpoint the precise transition index where the authorship shifts from human to machine with high granularity.

The effectiveness of our proposed approaches is validated through extensive experiments on the official datasets provided by the DeSegMa-IT shared task. Our systems demonstrated strong performance in both subtasks:

- **For Subtask A:** Our UmBERTo-based approach achieved state-of-the-art results, securing the 1st rank on the leaderboard with an accuracy of 0.945755.
- **For Subtask B:** Our token classification strategy using DeBERTa backbone yielded competitive results, achieving a Mean Absolute Error (MAE) of 62.66, placing us 3rd on the leaderboard.

This paper is structured as follows: In Section 2, we review related work on LLM-generated text detection and boundary detection in human-machine mixed texts. We also highlight some studies and their limitations, then position our work in this context. In Section 3, we provide a detailed overview of our proposed systems for both subtasks, including model architectures and training strategies. In Section 4, we describe the datasets used, evaluation metrics for each subtask, and our experimental setup. In Section 5, we present and discuss the results obtained by our systems on the official test sets. In Section 6, we analyze the strengths and limitations of our approaches and suggest potential avenues for future research. Finally, we conclude the paper in Section 7 and summarize our findings.

2. Related Work

2.1. Detecting LLM-Generated Text

The detection of LLM-generated text is often framed as a binary classification problem, where the objective is to train a classifier to distinguish between human-written text and LLM-generated text [9]. One traditional approach involves employing statistical techniques to identify generation artifacts. Such approaches utilize features including perplexity, entropy, Zipfian distribution of words, and n-gram frequency analysis. Feature-based methodologies typically operate by constructing high-dimensional vectors representing various stylistic and statistical properties of the text. Once extracted, these feature vectors serve as inputs for traditional machine learning classification algorithms, such as a support-vector machine (SVM), random forest (RF). For instance, Gehrmann et al. [10] developed a system that combines logistic regression with GLTR, a tool employing baseline statistical methods to detect generation artifacts across common sampling schemes.

Another prominent line of research involves leveraging Neural Language Models (NLMs) for text classification. Several works have fine-tuned large bidirectional language models, such as BERT and RoBERTa [11], on datasets containing both human-written and machine-generated text. These models utilize self-attention mechanisms to capture long-range dependencies and semantic coherence, which are often the hallmarks of neural generation. For example, Solaiman et al. [12] demonstrated that a fine-tuned RoBERTa model could achieve detection accuracy upward of 90% on GPT-2 generated text, significantly outperforming statistical baselines like GLTR. Adopting a generative approach, Chen et

al. [13] utilized the Text-to-Text Transfer Transformer (T5) classifier on a dataset named OpenLLMText, and then used these classifiers to predict the conditional probability of the next word, thereby classifying multiple text sources. By framing MGT detection as a text-to-text generation task, they utilized the model's perplexity and next-token probabilities to classify texts from diverse sources, demonstrating superior generalization compared to traditional classification heads. Zellers et al. [14] introduced 'Grover', a model designed to defend against neural fake news. They empirically showed that the most effective detector for a specific generator is often the generator itself, fine-tuned as a discriminator. Wang et al. [15] proposed SeqX-GPT, a method that utilizes log probability lists from white-box LLMs as features for sentence-level AIGT detection.

Furthermore, while multilingual models like mBERT [16] and XLM-RoBERTa [4] provide strong baselines, previous studies on the 'curse of multilinguality' [4] indicate that allocating model capacity across too many languages can dilute performance on individual ones. Empirical evidence was provided by Martin et al. [17] in the context of French NLP tasks, where they demonstrated that CamemBERT, a RoBERTa-based model trained exclusively on French text, outperformed multilingual counterparts by a significant margin.

2.2. Boundary Detection for Human-Machine Mixed Text

Beyond numerous studies focused on machine-generated text detection, a less-explored yet significant challenge lies in identifying precise boundaries within machine-human mixed texts. These texts consist of two segments: the first part is human-written, and the second part is generated by LLMs. Dugan et al. [18] conducted a comprehensive study to understand human ability to discern boundaries between human-written and machine-generated text. Their findings revealed significant variations in annotator proficiency and highlighted the impact of various factors on detection accuracy, such as model size, decoding strategy, fine-tuning,… etc. Zeng et al. [19] were among the first to formally define the task of identifying transition points between human-written and LLM-generated content. They proposed a two-step approach aimed to separate human and machine representations. First, they separate the LLM-generated text from human-written text during the encoder training process. Second, they calculated the distances between adjacent text prototypes, positing that the transition boundary is located between the two prototypes exhibiting the furthest distance from each other.

While these pioneering studies have proposed effective methodologies for boundary detection in hybrid texts, they were typically limited to the sentence level rather than at the token level. As a consequence, they may lead to poor granularity in realistic writing scenarios, where an LLM might complete a sentence started by a user. Addressing this gap, we approach boundary detection as a sequence labeling problem, enabling precise identification of transitions at the token level.

3. System Overview

Our systems for both subtasks are built upon pre-trained Italian transformer-based models. We utilize these models as backbone encoders and attach task-specific heads to adapt them to the respective tasks.

In Subtask A, we employ UmBERTo [6] as our backbone encoder for binary classification. We append a linear classification head on top of the [CLS] token representation to predict whether the input text is human-written or LLM-generated. See Section 3.1 for more details.

In Subtask B, the target is to identify the exact token index where the transition from human-written to machine-generated text occurs. As mentioned in Section 2.2, previous studies typically approached this problem at the sentence level, which may not provide sufficient granularity for practical applications. Therefore, we approach this task as a sequence labeling problem, employing DeBERTa-base-Italian [7] as our backbone encoder and attaching a token-level classification head to predict binary labels for each token. See Section 3.2 for more details.

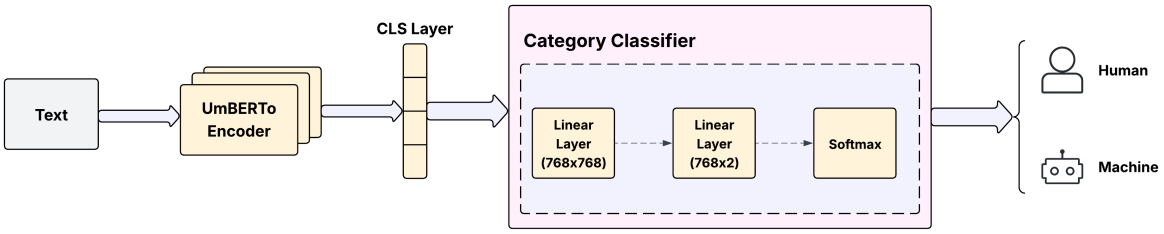

Figure 1: System overview for Subtask A

3.1. Subtask A: Binary Classification with UmBERTo

The overview of our system is illustrated in Figure 1. Our proposed system for Subtask A consists of two main components: a feature extractor layer (such as UmBERTo) that acquires text representation and a category classifier that predicts whether the input text is human-written or LLM-generated.

The feature extractor and the category classifier constitute a feedforward neural network architecture. Specifically, the feature extractor $G_f(\cdot)$ initially receives the text input, which consists of $\mathbf{n}$ tokens $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and then transforms it into high-dimensional contextualized embeddings. We utilize the output vector corresponding to the CLS token as the aggregate semantic summary of the entire sequence, denoted as $G_f(\mathbf{x})$. Subsequently, this semantic representation $G_f(\mathbf{x})$ is passed to the category classifier $G_y(\cdot)$, yielding $G_y(G_f(\mathbf{x}))$, representing the predicted probability of two labels through the Softmax function. Finally, the predicted label $\hat{y}$ is determined by selecting the class with the highest probability.

In the category classifier, we employ a two-layer Multi-Layer Perceptron (MLP) with a Tanh activation function between the layers. We then apply a Dropout operation to mitigate overfitting before passing the features through a second linear layer to map them to the binary label space.

To train the model, we employ the Cross-Entropy loss function, which minimizes the discrepancy between the predicted probabilities and the true labels. The loss function is defined as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{1}$$

where N is the number of samples in the batch, C is the number of classes (2 in this case), y_{ij} is the true label (one-hot encoded) for class j of sample i, and $\hat{y}_{ij}$ is the predicted probability for class j of sample i (obtained through the Softmax function).

3.2. Subtask B: Fine-Grained Boundary Detection via Token Classification

In subtask B, the objective is to identify the exact token index where the transition from human-written text to machine-generated text occurs. As mentioned in Section 2.2, we approach this task as a sequence labeling problem at the token level.

One of the main challenges in this subtask is aligning the ground-truth boundary index with the tokenized sequence produced by the tokenizer of the pre-trained model. To address this, we utilize the tokenizer's offset mapping feature, which provides the start and end character positions of each token in the original text. For each input sentence, the tokenizer generates a mapping (s_i, e_i) for each token t_i, where s_i and e_i denote the start and end character indices of the token in the original text, respectively. Then each token t_i is assigned a binary label y_i based on the ground-truth boundary index as follows:

$$y_i = \begin{cases} 0(\text{Human}) & \text{if } s_i < C_{trans} \\ 1(\text{Machine}) & \text{else} \end{cases} \tag{2}$$

where C_{trans} is the ground-truth character index of the transition point.

Table 1
Statistics of the DeSegMa-IT datasets for Subtask A and Subtask B

Subtask	Category	Train	Dev	Test
Subtask A	Human	16,569	–	–
	Machine	16,569	–	–
	Total	33,138	5,059	15,135
Subtask B	Total	19,945	4,579	23,211

We employ DeBERTa-base-Italian [7] as our backbone encoder for feature extraction. On top of the feature extractor, we attach a linear classification head that predicts binary labels for each token in the input sequence. Technically, the input sequence is initially encoded into hidden state $H = \{h_1, \ldots, h_n\}$. These states are then passed through the classification head, which consists of a single linear layer followed by a Softmax function, yielding predicted probabilities for each token:

$$P(\hat{y}_i|x_i) = \text{softmax}(W \cdot h_i + b) \tag{3}$$

Between the backbone encoder and the classification head, we apply a Dropout layer to mitigate overfitting.

4. Experimental setup

4.1. Datasets and Evaluation Metrics

Datasets We utilized the official datasets provided by organizers of the DeSegMa-IT [2] shared task. The corpus is exclusively in Italian for both subtasks.

- **For subtask A:** The dataset consists of over 33000 unique text samples, which are labeled as either human-written (0) or LLM-generated (1). The dataset is split into training, validation, and test sets. The texts vary in length from short paragraphs to longer articles, covering a diverse range of topics and styles.
- **For subtask B:** The dataset contains nearly 20000 text samples. Each document begins with a human-authored segment and transitions into machine-generated text at a specific, unmarked boundary. The objective is to pinpoint the exact token index of this transition.

See Table 1 for more details about the datasets of both Subtask A and Subtask B.

Evaluation Metrics The evaluation metric used in subtask A is accuracy, which is defined as the ratio of correctly predicted samples to the total number of samples. For subtask B, the evaluation metric is Mean Absolute Error (MAE), which measures the average absolute difference between the predicted boundary index and the actual boundary index. The performance is better when the MAE value is lower.

4.2. Training

Subtask A In subtask A, we divided the training into two phases: (1) fine-tuning the UmBERTo [6] model on the training set and (2) hyperparameter tuning on the validation set with the ratio of 80:20. For input processing in Subtask A, given the maximum context window of 512 tokens inherent to UmBERTo, we employed a standard truncation strategy, retaining only the initial 512 tokens of each document. The entire model, including both the UmBERTo backbone and the classification head, was fine-tuned end-to-end using the AdamW [20] optimizer coupled with a linear learning rate scheduler. The learning rate scheduler was configured with a fixed warm-up period of 100 steps. The model

Table 2
Official Leaderboard Results for Subtask A.

Team	Accuracy	Rank
Kenji Endo	0.9426	2
UniTor	0.9288	3
Niclas	0.9243	4
Stochastic G.D.	0.9216	5
Gradient Descenders	**0.9458**	**1**

Table 3
Ablation Study: Monolingual vs. Multilingual.

Model	Accuracy	Macro-F1	Micro-F1
mBERT (Base)	0.9393	0.9392	0.9393
XLM-RoBERTa (Base)	0.9397	0.9396	**0.9397**
mDeBERTa (Base)	0.9248	0.9245	0.9248
UmBERTo (Ours)	**0.9458**	**0.9376**	0.9378

is optimized by minimizing the Cross-Entropy loss function. To mitigate overfitting, we applied an Early Stopping strategy based on the validation loss, with a patience of 5 epochs. Furthermore, we incorporated a Dropout layer within the classification head and weight decay during optimization. For reproducibility, a comprehensive list of all hyperparameters, search spaces, and hardware specifications is provided in Appendix A.1

Subtask B In subtask B, we followed a similar training procedure as in Subtask A. We divided the dataset into training and validation sets with an 80:20 ratio. For input processing in Subtask B, we also truncated the input sequences to a maximum length of 512 tokens and assigned the special tokens ([CLS], [SEP]) with the label -100 to exclude them from loss computation. The entire model, including the DeBERTa-base-Italian [7] model and classification layer, was trained by minimizing the Cross-Entropy loss function using the AdamW [20] optimizer with weight decay. We also maintained the linear learning rate scheduler with a fixed warm-up period of 100 steps. During the evaluation, we observed that the token-level predictions can be noisy, resulting in unstable boundary detection and spurious label flipping. To address this issue, we implemented a consecutive consistency constraint during the inference phase. Specifically, a token at index k was identified as the transition point only if the model predicted the label 1 (Machine) for at least m consecutive tokens starting from index k. This approach effectively filters out isolated misclassifications and enhances the robustness of boundary detection. For reproducibility, more details about the hyperparameters and hardware specifications are provided in Appendix A.2.

5. Results

In this section, we present and discuss the results obtained by our proposed systems on both subtasks of the DeSegMa-IT [2] shared task. In subtask A, our system achieved the highest accuracy on the official test set, securing the 1st position on the leaderboard. In subtask B, our approach demonstrated competitive performance, achieving the 3rd rank on the leaderboard.

5.1. Subtask A: MGT Detection in the Wild

Table 2 presents the official leaderboard for Subtask A. Our proposed system, submitted under the team name *Gradient Descenders*, achieved the top position with an accuracy of 0.945755. As observed, the competition was intense, particularly between our system and the runner-up (*Kenji Endo*), with a margin

Table 4
Official Leaderboard Results for Subtask B.

Team	MAE	Rank
Stochastic G.D.	**52.54**	**1**
MINDS	56.53	2
UniTor	81.6	4
Nicla	102.04	5
Gradient Descenders	62.66	3

Table 5
Ablation Study: Backbone Comparison.

Model	MAE
UmBERTo (Base)	65.22
mDeBERTa-v3 (Base)	63.34
DeBERTa-IT (Ours)	**62.66**

of approximately 0.3%. However, our model demonstrated a significant lead over other competitors, outperforming the third-ranked team (*UniTor*) by over 1.7 percentage points.

To validate the effectiveness of our design choices, Table 3 compares our monolingual backbone against various baselines. The result empirically supports our hypothesis that leveraging a monolingual Italian pre-trained model, specifically UmBERTo [6], provides a substantial advantage in capturing the linguistic nuances necessary for effective discrimination between human and machine-generated text. Notably, UmBERTo outperforms the multilingual XLM-RoBERTa, which shares a similar architecture, by a margin of 0.64%, highlighting the benefit of language-specific pre-training. Moreover, it is particularly instructive to compare our system with *Stochastic Gradient Descenders*, a team from our same research group that employed a fundamentally different approach. While our system relies on an encoder-only architecture with full fine-tuning, the *Stochastic Gradient Descenders* team employed a generative Large Language Model (LLM) (Qwen2.5–0.5B model [21]) adapted via Low-Rank Adaptation (LoRA) [22]. Although Qwen2.5–0.5B is a highly effective multilingual model, our proposed system outperforms it by 2.42%. This result indicates that for the specific task of MGT detection in Italian, a well-tuned encoder-based model can outperform a generative LLM approach.

5.2. Subtask B: Human-Machine Text Segmentation

Table 4 shows the official leaderboard for Subtask B. Our system, *Gradient Descenders*, secured the 3rd position with a Mean Absolute Error (MAE) of 62.66. The system exhibited strong performance, establishing a significant gap over lower-ranked teams; specifically, we outperformed *Nicla* and *UniTor* by margins of 39.38 and 18.94 points, respectively.

Table 5 presents an ablation study comparing different backbone architectures for the boundary detection task. The empirical experiments have presented the effectiveness of both the DeBERTa architecture and monolingual pre-training. DeBERTa-IT surpasses its multilingual counterparts, mDeBERTa-V3 (63.34), restating the "monolingual advantage" observed in Subtask A.

It is noteworthy that our proposed system solely relied on a simple token classification layer, without incorporating more sophisticated techniques such as Conditional Random Fields (CRF) [23] or Bi-LSTM [24] layers. While computational budget constraints limited our exploration of larger backbones or more complex architectures, this result confirms that a well-tuned baseline with robust post-processing can achieve top-tier performance. Consequently, we believe there is ample room for improvement by scaling up the model size or integrating sequence modeling layers in future work.

6. Discussion

6.1. Subtask A Analysis

Our performance in Subtask A demonstrates the effectiveness of using a monolingual Italian pre-trained model, UmBERTo [6], for the task of MGT detection. These results align with prior findings on the 'curse of multilinguality' [4], which suggest that language-specific models can better capture linguistic nuances compared to multilingual counterparts. Although the performance gap is modest and comparable in terms of F1-score, we hypothesize that scaling up the volume and diversity of the Italian pre-training

data could significantly amplify this advantage, allowing the monolingual model to fully leverage its specialized capacity.

However, a notable challenge observed is the system's sensitivity to domain shift, where the distribution of the test data diverges from the training set. This is a challenge commonly faced in real-world applications, where the distribution of test data may differ significantly from the training data. To mitigate this limitation in future work, we propose exploring domain adaptation techniques. Guo et al. [25] addressed this issue by employing DANN (Domain-Adversarial Neural Networks) [26] to learn domain-invariant features, thereby enhancing the model's robustness to domain shifts. By employing a gradient reversal layer, such architectures can effectively align feature distributions across domains, thereby enhancing robustness.

Furthermore, augmenting the training data with synthetic examples generated by various LLMs could further enhance the model's generalization capabilities. The diversity in generation styles and artifacts in different LLMs can provide a richer training signal, enabling the model to generalize better to unseen data.

6.2. Subtask B Analysis

As discussed in Section 5, our approach for Subtask B, which treats boundary detection as a token classification problem, yielded competitive results despite its simplicity. The performance shows the strong ability of DeBERTa-base-Italian [7] in capturing token-level nuances necessary for identifying transition points between human and machine-generated text. Furthermore, this also validates our hypothesis that token-level classification can provide finer granularity compared to sentence-level approaches discussed in Section 2.2.

Despite these strengths, we acknowledge several limitations. First, our backbone model (DeBERTa-base-Italian) was fine-tuned on a limited corpus, which may limit its generalization capabilities. Second, while the DeBERTa encoder captures input context effectively, the simple linear classification head treats each token's prediction independently. It lacks the mechanism to explicitly model output label dependencies, potentially leading to misclassifications around the transition boundary. Guo et al. [25] have experimented by using Bi-LSTM [24] layers on top of pre-trained models for similar token-level tasks, providing a significant performance boost rather than using only linear layers. Finally, the fixed truncation at 512 tokens is a bottleneck for long-document analysis. Future iterations will investigate models with extended context windows (e.g., Longformer) or sliding window inference strategies to address boundaries situated deep within the text.

7. Conclusion

In this paper, we presented the Gradient Descenders systems developed for the DeSegMa-IT [2] shared task at EVALITA 2026 [3]. Our approach yielded highly competitive results: we achieved the 1st rank in Subtask A with an accuracy of 94.58% and secured the 3rd rank in Subtask B with a Mean Absolute Error (MAE) of 62.66. These findings empirically demonstrate the effectiveness of leveraging monolingual Italian pre-trained models, specifically UmBERTo [6] and DeBERTa-base-Italian [7], for MGT detection and boundary detection tasks, respectively. We also discussed the strengths and limitations of our approaches, providing insights into potential avenues for future research. Looking ahead, we plan to enhance model robustness against domain shifts using Domain-Adversarial Neural Networks (DANN) [26] and to refine boundary precision by integrating structured prediction layers, such as Bi-LSTM [24], into our pipeline.

Declaration on Generative AI

Generative AI tools were employed only to enhance linguistic clarity. All data, analyses, and conclusions were conceived, executed, and validated by the authors. The authors retain full accountability for every

aspect of the manuscript's content.

References

[1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. URL: https://arxiv.org/abs/2302.13971. arXiv:2302.13971.

[2] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[3] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020. URL: https://arxiv.org/abs/1911.02116. arXiv:1911.02116.

[5] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. arXiv:2111.09543.

[6] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, https://github.com/musixmatchresearch/umberto, 2020.

[7] Osiria, Deberta-base-italian: Italian deberta model, https://huggingface.co/osiria/deberta-base-italian, 2023. Pre-trained Italian language model based on DeBERTa architecture.

[8] A. Abdaoui, C. Pradel, G. Sigel, Load what you need: Smaller versions of multilingual bert, 2020. URL: https://arxiv.org/abs/2010.05609. arXiv:2010.05609.

[9] H.-Q. Nguyen-Son, N.-D. T. Tieu, H. H. Nguyen, J. Yamagishi, I. E. Zen, Identifying computer-generated text using statistical analysis, in: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 1504–1511. doi:10.1109/APSIPA.2017.8282270.

[10] S. Gehrmann, H. Strobelt, A. M. Rush, Gltr: Statistical detection and visualization of generated text, 2019. URL: https://arxiv.org/abs/1906.04043. arXiv:1906.04043.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[12] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, J. Wang, Release strategies and the social impacts of language models, 2019. URL: https://arxiv.org/abs/1908.09203. arXiv:1908.09203.

[13] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, B. Raj, Token prediction as implicit classification to identify LLM-generated text, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 13112–13120. URL: https://aclanthology.org/2023.emnlp-main.810/. doi:10.18653/v1/2023.emnlp-main.810.

[14] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, 2020. URL: https://arxiv.org/abs/1905.12616. arXiv:1905.12616.

[15] P. Wang, L. Li, K. Ren, B. Jiang, D. Zhang, X. Qiu, Seqxgpt: Sentence-level ai-generated text detection, 2023. URL: https://arxiv.org/abs/2310.08903. arXiv:2310.08903.

[16] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers

for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[17] L. Martin, B. Muller, P. J. Ortiz Suarez, Y. Dupont, L. Romary, E. de la Clergerie, D. Seddah, B. Sagot, Camembert: a tasty french language model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 7203–7219. URL: http://dx.doi.org/10.18653/v1/2020.acl-main.645. doi:10.18653/v1/2020.acl-main.645.

[18] L. Dugan, D. Ippolito, A. Kirubarajan, S. Shi, C. Callison-Burch, Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text, 2022. URL: https://arxiv.org/abs/2212.12672. arXiv:2212.12672.

[19] Z. Zeng, L. Sha, Y. Li, K. Yang, D. Gašević, G. Chen, Towards automatic boundary detection for human-ai collaborative hybrid essay in education, 2023. URL: https://arxiv.org/abs/2307.12267. arXiv:2307.12267.

[20] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: https://arxiv.org/abs/1711.05101. arXiv:1711.05101.

[21] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, Z. Fan, Qwen2 technical report, 2024. URL: https://arxiv.org/abs/2407.10671. arXiv:2407.10671.

[22] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[23] C. Sutton, A. McCallum, An introduction to conditional random fields, 2010. URL: https://arxiv.org/abs/1011.4088. arXiv:1011.4088.

[24] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: K. Erk, N. A. Smith (Eds.), Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 207–212. URL: https://aclanthology.org/P16-2034/. doi:10.18653/v1/P16-2034.

[25] Z. Guo, K. Jiao, X. Yao, Y. Wan, H. Li, B. Xu, L. Zhang, Q. Wang, Y. Zhang, Z. Mao, USTC-BUPT at SemEval-2024 task 8: Enhancing machine-generated text detection via domain adversarial neural networks and LLM embeddings, in: A. K. Ojha, A. S. Doğruöz, H. Tayyar Madabushi, G. Da San Martino, S. Rosenthal, A. Rosá (Eds.), Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 1511–1522. URL: https://aclanthology.org/2024.semeval-1.217/. doi:10.18653/v1/2024.semeval-1.217.

[26] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, 2016. URL: https://arxiv.org/abs/1505.07818. arXiv:1505.07818.

A. Detailed Experimental Setup

A.1. Subtask A: Binary Classification with UmBERTo

We utilized UmBERTo (Cased-v1) [6] as our backbone encoder for Subtask A, coupled with the two-layer MLP classification head described previously. The model was fine-tuned using the AdamW [20] optimizer with a weight decay of 1e-2 to regularize the weights. We employed a linear learning rate scheduler with a warm-up period of 100 steps. After the warm-up phase, the learning rate decayed linearly to zero.

Regarding hyperparameters, the peak learning rate was set to 2e-5 with a batch size of 8. The input

sequences were truncated to a maximum length of 512 tokens. As mentioned in Section 4, we employed an Early Stopping strategy based on the validation loss, with a patience of 5 epochs. The Dropout rate in the classification head was set to 0.2. We conducted training for 10 epochs on NVIDIA Tesla T4 GPUs.

A.2. Subtask B: Human-Machine Text Segmentation

We utilized DeBERTa-base-Italian [7] as our backbone encoder for Subtask B, along with the token-level classification head described previously. The model was trained using the AdamW [20] optimizer with a weight decay of 1e-2. We employed a linear learning rate scheduler with a warm-up period of 100 steps, similar to Subtask A.

Regarding hyperparameters, the learning rate was set at 1e-5, and the batch size was 8. Input sequences were truncated to a maximum length of 512 tokens, with special tokens ([CLS], [SEP]) assigned a label of -100 to exclude them from loss computation. A dropout with a rate of 0.3 was applied before the classification head to mitigate overfitting. For the consecutive consistency constraint during inference, we set the parameter m to 2, meaning that a token is classified as the transition point only if at least 2 consecutive tokens are predicted as machine-generated starting from that token. The training phase was conducted for 6 epochs on NVIDIA Tesla T4 GPUs.