GigaCheck: Detecting LLM-generated Content via Object-Centric Span Localization

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, Aleksandr Gordeev, Vladimir Dokholyan, Maksim Kuprashevich

SALUTEDEV LLC, Tashkent, Uzbekistan

Correspondence: irinakr4snova@gmail.com

Abstract

With the increasing quality and spread of LLM assistants, the amount of generated content is growing rapidly. In many cases and tasks, such texts are already indistinguishable from those written by humans, and the quality of generation continues to increase. At the same time, detection methods are advancing more slowly than generation models, making it challenging to prevent misuse of generative AI technologies. We propose GigaCheck, a dual-strategy framework for AI-generated text detection. At the document level, we leverage the representation learning of fine-tuned LLMs to discern authorship with high data efficiency. At the span level, we introduce a novel structural adaptation that treats generated text segments as "objects." By integrating a DETR-like vision model with linguistic encoders, we achieve precise localization of AI intervals, effectively transferring the robustness of visual object detection to the textual domain. Experimental results across three classification and three localization benchmarks confirm the robustness of our approach. The shared fine-tuned backbone delivers strong accuracy in both scenarios, highlighting the generalization power of the learned embeddings. Moreover, we successfully demonstrate that visual detection architectures like DETR are not limited to pixel space, effectively generalizing to the localization of generated text spans. To ensure reproducibility and foster further research, we publicly release our source code.

1 Introduction

The rapid development of Large Language Models (LLMs) has made their outputs difficult to distinguish from human-written text, raising concerns about the spread of spam and misinformation (Mirsky et al., 2023; Hanley and Durumeric, 2024), fraud (Grbic and Dujlovic, 2023; Roy et al., 2023), and academic cheating (Stokel-Walker, 2022; Kasneci et al., 2023; Perkins et al., 2023; Vasilatos et al., 2023). LLMs produce hallucinations (Ji et al., 2023; Thorp, 2023) and outdated information, thereby spreading incorrect knowledge. Detecting LLM-generated content remains challenging, especially in mixed-authorship scenarios (Human-Machine collaborative texts), where existing document-level detectors lack sufficient reliability (Liu et al., 2023c; Wu et al., 2023a).

Recent approaches have shifted towards analyzing collaborative texts by identifying boundaries between sections of different authorship (Zeng et al., 2024b,a; Wang et al., 2023) or employing fine-grained token classification to extract spans (Yin and Wang, 2026).

In this paper, we propose a unified framework for generated text analysis, targeting both document-level classification and fine-grained span-level localization. For the latter, we introduce a paradigm shift by reformulating text span detection as an *object detection* problem. We employ a DETR-based architecture (Carion et al., 2020) that leverages representations from a fine-tuned LLM to predict character-based segments directly. Unlike previous sequence labeling methods that require manual post-processing to group tokens (Kushnareva et al.; Zeng et al., 2024b; Wang et al., 2023), our encoder-decoder transformer predicts continuous intervals end-to-end.

To keep the study focused and directly comparable with existing benchmarks, we limit this first investigation to English texts; adapting GigaCheck to new languages is straightforward and left for promising future work.

To assess our approach, we adopt a two–step evaluation strategy. We begin with the challenging *span-level localization* setting, demonstrating that the proposed DETR head can precisely pinpoint LLM-generated spans across three Human–Machine collaborative datasets. We then turn to three well-established binary-classification corpora. Although binary detection is less novel, these

Findings of the Association for Computational Linguistics: ACL 2026, pages 4349–4364

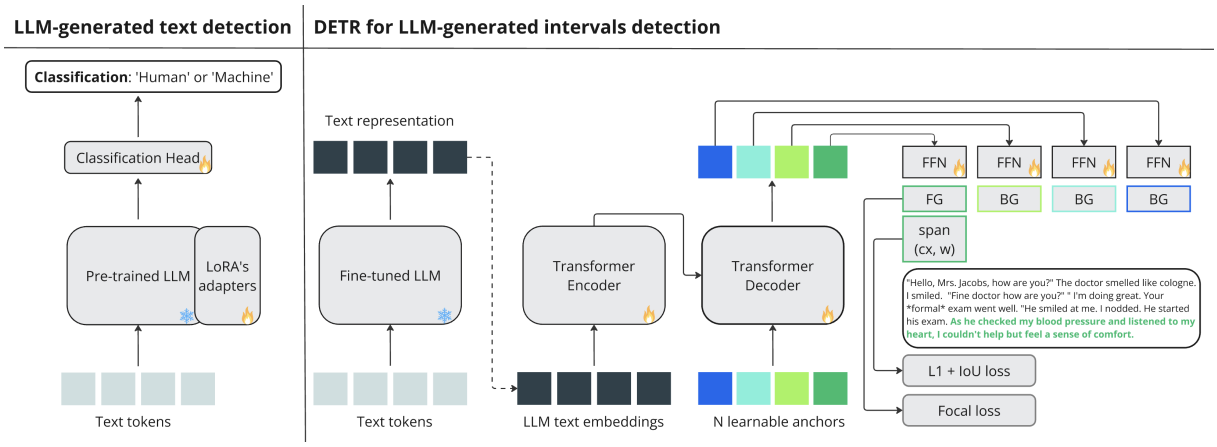

Figure 1: Overall architecture of GigaCheck framework. Document-level detection is performed by fine-tuning an LLM. For span-level localization, we adopt a two-stage pipeline: (1) a fine-tuned LLM produces token embeddings, and (2) a detection transformer treats generated spans as objects and directly predicts character-level intervals. FG and BG denote the foreground and background labels assigned to each anchor.

experiments verify that the very same LoRA-tuned backbone used by the DETR head learns embeddings that remain robust and discriminative for independent downstream tasks.

Our contributions are:

1. **Object Detection paradigm for text spans.** To the best of our knowledge, DETR-style models have not yet been applied to locating intervals within natural language texts. We take this first step by adapting the architecture to detect LLM-generated segments as discrete objects, achieving strong results across three localization benchmarks. This approach eliminates the need for heuristic post-processing common in token-classification methods.

2. **Robust backbone for both detection and classification.** The same LoRA-tuned backbone delivers state-of-the-art performance on three binary-classification datasets, proving that its embeddings transfer reliably between fine-grained localization and global document-level detection tasks.

3. **Open Source Availability.** To facilitate reproducibility and encourage future developments in the field, we make our source code publicly available at `https://github.com/ai-forever/gigacheck`.

2 Related Works

2.1 Text Classification Methods

Detecting machine-generated content has been widely studied. Work mainly focuses on binary classification (human vs. AI) (Zhang et al., 2024; Liu et al., 2023c; Bhattacharjee and Liu, 2024; Liu et al., 2023a; Uchendu et al., 2020) and multi-class tasks to identify the specific generation model (Uchendu et al., 2020, 2021, 2023; Wang et al., 2024; Mitchell et al., 2023; Wu et al., 2023b).

Statistical methods (Mitchell et al., 2023; Gehrmann et al., 2019; Su et al., 2023; Fröhling and Zubiaga, 2021) use metrics like entropy, perplexity, and n-gram frequency, and typically require access to the investigated LLMs. Neural-based approaches (Antoun et al., 2023; Wang et al., 2024; Guo et al., 2023; Liu et al., 2023b; Zellers et al., 2019; Solaiman et al., 2019; Uchendu et al., 2020), primarily using RoBERTa (Liu, 2019), provide more accurate results than statistical methods (Li et al., 2024; Liu et al., 2023b), but lack robustness (Li et al., 2024; Koike et al., 2024; Krishna et al., 2024; Chakraborty et al., 2023; Tulchinskii et al., 2024). Recent works incorporate topological data analysis (TDA) (Uchendu et al., 2023; Kushnareva et al., 2021; Tulchinskii et al., 2024) or leverage LLMs as detectors. The authors of Bhattacharjee and Liu (2024) apply GPT-3.5-turbo (OpenAI, 2023a) and GPT-4 (OpenAI, 2023b) models for the zero-shot binary classification task and demonstrate that both models have a very high misclassification rate. Our method extends neural-based detectors by fine-tuning an LLM to distinguish real and machine-generated text.

2.2 Co-Written Text Analysis

Several studies (Zhang et al., 2024; Liu et al., 2023c) utilize neural-based classification models

Table 1: Datasets used for training and evaluating the proposed approach (adapted from (Uchendu et al., 2021; Fagni et al., 2021; Zhang et al., 2024; Li et al., 2024; Dugan et al., 2023; Kushnareva et al.; Zeng et al., 2024b)). The tasks include both classification and detection. Note that "#" represents "number of".

Task	Dataset	Generators	Domains	# Texts	# Boundaries
Classification	TuringBench	FAIR wmt20	News	17,163	-
	TuringBench	GPT-3	News	17,018	-
	TweepFake	Markov Chains, RNN, RNN+ Markov, LSTM, GPT-2	Tweets	25,572	-
	MAGE	27 LLMs from seven groups: GPT, LLaMA, GLM-130B, FLAN-T5, OPT, BigScience, EleutherAI	Reddit opinions, review, news, question answering, story, commonsense reasoning, Wikipedia paragraph, scientific writing	447,674	-
Detection	RoFT	GPT-2/XL, CTRL	Speeches, recipes, news, short stories	8,943	1
	RoFT-ChatGPT	GPT-3.5 Turbo	Speeches, recipes, news, short stories	6,940	1
	TriBERT	ChatGPT	Educational essays	17,136	1-3

to classify Human-Machine collaborative texts. Kushnareva et al. address the boundary detection task to determine where human-written text ends and machine-generated text begins, using fine-tuned RoBERTa and TDA-based time series. Zeng et al. (2024b) measure distances between adjacent segments to identify transitions, while Zeng et al. (2024a) employ segmentation and classification of segments into AI-generated, human-written, or collaborative. A simpler approach by Wang et al. (2023) identifies exact authorship for each sentence.

More recently, Yin and Wang (2026) introduced Sci-SpanDet, a structure-aware framework designed specifically for scientific papers. They combine BIO-CRF sequence labeling with pointer networks to detect contiguous AI-generated spans, relying on section-specific contrastive learning that leverages the IMRaD structure (Introduction, Methods, Results, Discussion) of scientific documents. While effective in its target domain, Sci-SpanDet is inherently tied to structured document formats and cannot be directly applied to arbitrary texts lacking such explicit organization.

In contrast, our approach is domain- and structure-agnostic: by reformulating span detection as 1D object detection over character-level intervals, we eliminate the dependency on predefined document layouts or sentence-level granularity, enabling flexible detection of multiple generated segments in any text.

2.3 Transformer-based detection models

DETR (Carion et al., 2020) is an end-to-end object detector based on transformers. DETR-like architectures have proven effectiveness in object detection (Zong et al., 2023; Hou et al., 2024; Huang et al., 2022) and related tasks like video action detection (Zhang et al., 2021) and moment retrieval (Lei et al., 2021; Moon et al., 2023; Gordeev et al., 2024), where it is used to find temporal intervals in videos corresponding to a given text query. Inspired by these works, we propose to use a detection transformer model to perform span-level detection in texts.

Recent DETR modifications improve efficiency and accuracy: DeformableDETR (Zhu et al., 2020) speeds up convergence with deformable attention; DN-DETR (Li et al., 2022) uses denoising training to accelerate the training process and improve detection accuracy; DAB-DETR (Liu et al., 2022) refines predictions by introducing learnable anchor boxes as DETR positional queries. DINO DETR (Zhang et al., 2022) combines these features and integrates RPN, while CO-DETR (Zong et al., 2023) enhances efficiency with auxiliary heads.

We adopt DN-DAB-DETR for its strong baseline and high localization accuracy (Li et al., 2022). We also tested DAB-DETR, DeformableDETR, and CO-DETR, but DN-DAB-DETR consistently yielded the best results, so we adopt it throughout.

3 Methodology

Figure 1 illustrates the architecture of GigaCheck. Our framework addresses two complementary tasks using a unified text-representation strategy: span-level localization and document-level classification. We employ a LoRA-tuned LLM whose token embeddings feed into two specialized heads. Below, we first present the backbone, followed by

our novel object-centric span detector, and finally the classification head.

3.1 Unified text-representation backbone

We fine-tune a general-purpose decoder LLM, namely Mistral-7B,[1] with LoRA (Hu et al., 2021). LoRA decomposes the weight matrix into two low-rank trainable matrices while keeping pre-trained weights frozen, yielding parameter-efficient fine-tuning (PEFT). We chose LoRA because (i) most of the datasets we use are small (see in Table 1), where PEFT often generalises better than full fine-tuning, and (ii) it converges much faster, saving GPU hours. Although results are reported with Mistral, the backbone is model-agnostic and any decoder-style LLM can be swapped in with minimal changes.

Proxy task. The LLM is tuned on a lightweight proxy classification task with two variants:

1. **three-class proxy** (*human*, *machine*, *collaborative*): used as a *frozen feature extractor* for the DETR training.

2. **two-class proxy** (*human*, *machine*): is *trainable* along with the binary-classification head.

For a document $\mathbf{X}$ we obtain tokens and embeddings via

$$\begin{aligned} \mathbf{T} &= \text{Tokenizer}(\mathbf{X}), \\ \mathbf{E} &= \text{LLM}_{ft}(\mathbf{T}), \quad e_i \in \mathbb{R}^{d_{\text{model}}}, \end{aligned} \tag{1}$$

where Tokenizer is the BPE tokenizer shipped with Mistral and LLM_{ft} is the LoRA-tuned required by the downstream head. If fine-tuning is infeasible, pre-trained LLM embeddings may be substituted (see Appendix A).

3.2 Object-centric Span Localization (DETR)

Our core contribution is the reformulation of text analysis as an object detection problem. We introduce a DETR-like head that treats LLM-generated segments as discrete objects, directly regressing 1-D character spans parameterized by c and width w (normalised to $[0, 1]$). This approach avoids the limitations of token-level sequence labeling and operates independently of sentence boundaries.

[1] https://huggingface.co/mistralai/Mistral-7B-v0.3

Architecture. Embeddings $\mathbf{E}$, obtained in Equation 1 from the frozen backbone, are first linearly projected to a lower dimension and then passed through a Transformer encoder to obtain contextual features:

$$\begin{aligned} \overline{\mathbf{E}} &= \text{Linear}(\mathbf{E}), \\ \mathbf{R} &= \text{TransformerEncoder}(\overline{\mathbf{E}}) \end{aligned} \tag{2}$$

We then follow DAB-DETR (Liu et al., 2022). A set of N *anchor-based learnable queries* $\mathbf{q} = \{q_0, \ldots, q_{N-1}\}$ is initialised with reference points (c, w), which act as initial hypotheses for the locations and lengths of LLM-generated spans. These queries are fed to the Transformer decoder, where sinusoidal encodings inject the anchor positions, and each cross-attention block concatenates positional and content embeddings, allowing the decoder to refine each anchor iteratively. At decoder layer ℓ the decoder predicts an offset $(\Delta c^{(\ell)}, \Delta w^{(\ell)})$ for each anchor and updates it as

$$(c, w)^{(\ell+1)} = (c, w)^{(\ell)} + (\Delta c^{(\ell)}, \Delta w^{(\ell)}).$$

After L layers the decoder produces N refined spans:

$$\mathbf{o} = \text{TransformerDecoder}(\mathbf{q}, \mathbf{R}), \tag{3}$$

where $\mathbf{o} = \{o_0, \ldots, o_{N-1}\}$ corresponds one-to-one with the anchor queries.
As the model output, for each query the detector outputs a triplet (c, w, p) comprising the refined centre c, width w, and a confidence score $p \in [0, 1]$ that the span is LLM-generated. Thresholding p yields up to N one-dimensional spans flagged as machine-written.
The number of queries N is a dataset-level hyperparameter set according to the maximum expected span density.

Stabilising early training. As in DN-DETR (Li et al., 2022), the decoder is trained with two types of inputs: (i) the learnable anchor queries, and (ii) noisy versions of the ground-truth (GT) spans. The model is trained to denoise these GT queries, while an attention mask prevents them from leaking information to the anchor queries.

Training loss. Before computing losses, we use Hungarian matching to pair each prediction with a GT span; the noised GT queries are excluded from this matching. The final objective is a weighted sum of L1, gIoU (Rezatofighi et al., 2019), and Focal (Lin, 2017) losses for the matched predictions,

plus the same L1 and gIoU terms applied to the denoised GT queries.

We refer to the described detection transformer model as **GigaCheck (DN-DAB-DETR)**.

3.3 Binary classification head

The second head answers the document-level question *"Is this text human-written or LLM-generated?"*. Formally, for a document X we learn

$$f_\theta : X \longrightarrow \{0,1\}, \quad f_\theta(X) = \begin{cases} 0, & \text{human}, \\ 1, & \text{machine}. \end{cases}$$

We attach a two-layer MLP to the hidden state of the final <EOS> token of the *two-class* LoRA variant and train it with binary cross-entropy. The resulting model is referred to as **GigaCheck (Mistral-7B)**.

4 Datasets and Metrics

Table 1 lists all datasets used in this work. We use the original train–test splits in Section 5, enabling comparison with other approaches trained on the same data.

Classification datasets. We evaluate the proposed approach for machine-written text classification using three datasets: TuringBench (Uchendu et al., 2021), TweepFake (Fagni et al., 2021), and MAGE (Li et al., 2024). We prioritized these benchmarks while noting that other existing corpora, such as MixSet (Zhang et al., 2024) or Ghostbusters (Verma et al., 2023), consist of a limited amount of data. Such small-scale datasets are known to be easily solvable and often fail to reflect the complexity of real-world detection scenarios (Gritsai et al., 2024). Regarding TuringBench, we specifically use the two subsets generated by FAIR wmt20 (Chen et al., 2020) and GPT-3 (Brown et al., 2020), as these models produce texts most indistinguishable from human-written ones according to the dataset authors.

Detection datasets. We considered three datasets for Human-Machine collaborative text analysis, which have been created to address the task of identifying a boundary between human-written and machine-generated text: RoFT (Dugan et al., 2023), RoFT-ChatGPT (Kushnareva et al.), and TriBERT (Zeng et al., 2024b).

Classification metrics. We evaluate GigaCheck as an LLM-generated content detector using classification accuracy (Acc), F1 score, AUROC, and average recall (AvgRec) (Li et al., 2024), calculated as the average of recall scores for human-written (HumanRec) and machine-generated (MachineRec) texts.

Detection metrics. We use metrics such as sentence-wise MSE, Accuracy, and Soft Accuracy from Kushnareva et al., as well as a specialized form of the F1 score from Zeng et al. (2024b), to assess the quality of the model's predictions of the boundaries between sentences written by a human or an LLM. The authors of Zeng et al. (2024b) consider L_{topK}, which represents the top-K boundaries identified by the algorithm, and L_{Gt}, which refers to the number of ground-truth boundaries. The F1 score is then determined using the following formula:

$$F1@K = 2 \cdot \frac{|L_{topK} \cap L_{Gt}|}{|L_{topK}| + |L_{Gt}|} \tag{4}$$

Further details on the calculation of each metric are provided in Appendix E.

5 Experimental Results

In this section we first report span-detection results on three Human-Machine collaborative datasets, then present an extensive evaluation on three binary-classification benchmarks. While the classification task itself is well studied, these additional experiments serve to verify that the proposed text-representation backbone produces embeddings that remain robust and discriminative for a separate downstream task. Training details for all runs are provided in Appendix B.

5.1 Detection Results

To provide a comprehensive assessment, we benchmark GigaCheck against a diverse spectrum of baselines operating at varying granularities. We evaluate our span-detection method on the RoFT and RoFT-GPT datasets against approaches operating at the token level, sentence level, and document level. This inclusion allows us to compare our object-centric approach directly with traditional fine-grained methods. For the TriBERT dataset, following established protocols, we compare our method with sentence-level approach.

RoFT and RoFT-ChatGPT results. In experiments on the RoFT and RoFT-ChatGPT datasets, we fine-tuned Mistral-7B to distinguish between human-written texts and texts co-written with LLMs. Features from the model's last layer were used to train the `GigaCheck (DN-DAB-DETR)` model. Since each text in these datasets contains at

Table 2: Boundary detection results on RoFT and RoFT-ChatGPT datasets. The results for all methods, except ours, were taken from Kushnareva et al..

Method	RoFT			RoFT-ChatGPT		
	Acc	SoftAcc1	MSE	Acc	SoftAcc1	MSE
RoBERTa + SEP (Cutler et al., 2021)	0.50	0.80	2.63	0.55	0.79	3.06
RoBERTa (Liu, 2019)	0.46	0.75	3.00	0.39	0.75	3.15
GigaCheck (DN-DAB-DETR)	**0.65**	**0.87**	**1.51**	**0.68**	**0.89**	**1.03**
Based on Perplexity						
Phi-1.5 (Li et al., 2023) Perpl. + GB regressor	0.17	0.45	6.11	0.32	0.71	3.07
Phi-1.5 (Li et al., 2023) Perpl. + LR classifier	0.27	0.50	11.9	0.47	0.73	4.77
Based on TDA						
PHD + TS ML (Kushnareva et al.)	0.24	0.46	14.40	0.17	0.36	14.45
TLE + TS Binary (Kushnareva et al.)	0.13	0.30	22.23	0.20	0.35	18.52
Human baseline (Cutler et al., 2021)	0.23	0.40	13.88	-	-	-

Table 3: Accuracy for leave-one-out cross-domain evaluation on *RoFT-ChatGPT*. The results for all methods, except ours, were taken from Kushnareva et al..

Pred.	Model	Context	Pres. Speeches	Recipes	New York Times	Short Stories
Text	**GigaCheck (DN-DAB-DETR)**	global	0.50	**0.33**	**0.55**	**0.64**
Text	RoBERTa SEP (Cutler et al., 2021)	global	0.31	0.13	0.38	0.29
Text	RoBERTa (Liu, 2019)	global	0.36	0.15	0.38	0.36
Perpl.	Phi1.5 (Li et al., 2023), GB	sent.	**0.52**	0.24	0.46	0.56
Perpl.	Phi1.5 (Li et al., 2023), LR	sent.	0.41	0.21	0.45	0.52
PHD	TS multi (Kushnareva et al.)	100 tkn	0.13	0.20	0.17	0.18
TLE	TS Binary (Kushnareva et al.)	20 tkn	0.15	0.16	0.17	0.11

Table 4: Evaluation of GigaCheck (DN-DAB-DETR) on RoFT and RoFT-GPT datasets using mAP@0.5-0.95. The table compares the leave-one-out cross-domain setting against models trained on all domains combined.

Dataset	mAP@0.5-0.95
RoFT-ChatGPT Short Stories	0.7626
RoFT-ChatGPT Recipes	0.6046
RoFT-ChatGPT Pres Speeches	0.5933
RoFT-ChatGPT New York Times	0.7034
RoFT-ChatGPT All domains	0.8135
RoFT All domains	0.7972

Table 5: Boundary detection results (F1@3) on the TriBERT (Zeng et al., 2024b) dataset. #Bry denotes the number of ground-truth boundaries in the texts. Measurements are presented in original and rescaled formats.

Methods	#Bry=1	#Bry=2	#Bry=3	All
Original values				
TriBERT (p=2)	**0.455**	0.692	0.622	0.575
GigaCheck (DN-DAB-DETR)	0.444	**0.693**	**0.801**	**0.646**
Rescaled values				
TriBERT (p=2)	**0.910**	0.865	0.622	-
GigaCheck (DN-DAB-DETR)	0.888	**0.867**	**0.801**	-

most one human-to-machine transition, the detector uses a single learnable query (N=1).

`GigaCheck (DN-DAB-DETR)` natively predicts continuous character-level intervals end-to-end, without any heuristic post-processing. Since the official RoFT metrics operate on sentence boundaries, we apply a deterministic character-to-sentence projection solely for evaluation purposes (details in Appendix F).

Table 2 shows that `GigaCheck (DN-DAB-DETR)` beats the RoBERTa baseline by 15% on RoFT and 13% on RoFT-ChatGPT, and reduces MSE on RoFT-ChatGPT by a factor of 3. Table 3 shows cross-domain results on RoFT-ChatGPT, where models trained on three domains and tested on the fourth. Our approach achieves the best cross-domain generalization, though performance on the *Recipes* domain remains relatively low.

We additionally report the standard mean Average Precision (mAP) adapted for one-dimensional intervals (Table 4). An interval is considered a true positive if its IoU with a ground-truth interval exceeds a given threshold; mAP@0.5:0.95 averages over thresholds from 0.5 to 0.95. Unlike the sentence-level metrics above, mAP operates directly on character-level predictions and requires no projection, confirming that the model achieves strong localization at the native output granularity.

Examples of raw model output on RoFT-ChatGPT are provided in Appendix G.

TriBERT results. TriBERT texts contain up to three authorship boundaries, yielding denser spans;

accordingly, the detector uses 18 learnable queries (N=18) to provide sufficient capacity. Because the TriBERT dataset is small, we keep Mistral-7B-v0.3 frozen and feed its embeddings to `GigaCheck (DN-DAB-DETR)`. The detector outputs character spans, which we map to sentence boundaries to compute $F1@3$ (Eq. 4; mapping details in Appendix F).

Results are reported by boundary count (1, 2, 3) and for the full set. With $K \neq 3$ the ideal $F1@3$ scores are 0.5, 0.8, 1.0 (Zeng et al., 2024b). We rescale them to a common scale, where the ideal $F1@3$ is 1.0, for clarity. Table 5 shows a 7.1% gain over TriBERT model on the full set and higher scores for 2- and 3-boundary texts, while performance is similar for the 1-boundary group. Unlike TriBERT, our model stays stable as the number of boundaries increases.

5.2 Classification Results

We fine-tuned Mistral-7B v0.3 with LoRA on five datasets, comparing to baselines provided by the authors of these datasets. All our models were trained on the same training sets used by the authors.

Tables 6 and 7 show strong results on **TweepFake** and **TuringBench**, outperforming statistical methods and fine-tuned LM baselines across diverse domains and generators.

MAGE results. Table 8 compares `GigaCheck (Mistral-7B)` with the strongest baseline reported by the dataset authors (full results in Appendix C) and shows that our model reaches AUROC = 0.99 and AvgRec = 0.96 on the full large-scale split. It keeps strong generalisation: AvgRec = 0.89 in the *unseen-domain + unseen-model* test, 0.69 under paraphrase attacks, and AUROC = 0.98 / AvgRec = 0.92 in the *out-of-model* setting, where texts from specific generators were excluded during training.

Effect of backbone size. To gauge the impact of scale we repeated the full-data experiment on MAGE (the largest corpus in our experiments) using three larger LoRA-tuned backbones: Mistral-Nemo-Base-2407 (12 B), Mistral-Small-24B-Base-2501 (24 B), and Qwen2.5-72B-Instruct (72 B). As reported in Table 9, accuracy rises with backbone size overall, yet the 72B Qwen variant drops to the lowest score, hinting at overfitting. Because the gains beyond 7B are modest relative to the added compute, we keep the 7 B backbone for all other datasets; it trains quickly, fits standard memory limits, and is less prone to overfitting on small corpora even with LoRA.

In summary, our approach with 7B backbone effectively distinguishes LLM-generated texts from human-written ones when trained on both small and large datasets. The experiments demonstrate the robustness of our method for out-of-domain and out-of-model detection, as well as its resistance to paraphrasing attacks. Additionally, Appendix D presents a comparison between the fine-tuned `GigaCheck (Mistral-7B)` models and the Mistral-7B-Instruct-v0.3 model, evaluated in a zero-shot setting across each test set.

Table 6: Experimental results on TweepFake test set. F1 scores are reported as 'human' / 'machine'.

Method	F1	Acc
BERT (Devlin, 2018)	0.890 / 0.892	0.891
DistilBERT (Sanh, 2019)	0.886 / 0.888	0.887
RoBERTa (Liu, 2019)	0.895 / 0.897	0.896
XLNet (Yang, 2019)	0.871 / 0.882	0.877
GigaCheck (Mistral-7B)	**0.944 / 0.942**	**0.943**

Table 7: Experimental results (F1) on two TuringBench subsets. F1 is calculated for the machine-generated category.

Method	FAIR_wmt20	GPT-3
GLTR (Gehrmann et al., 2019)	0.4907	0.3476
BERT (Devlin, 2018)	0.4701	0.7944
RoBERTa (Liu, 2019)	0.4531	0.5209
GigaCheck (Mistral-7B)	**0.9966**	**0.9709**

6 Conclusions

We presented `GigaCheck`, a unified framework that combines a LoRA-tuned backbone LLM with two lightweight heads: (i) a DN-DAB-DETR module for precise character-level localization of LLM-generated spans, and (ii) a streamlined MLP for document-level authorship verification.

Our experiments on three Human–Machine collaborative datasets demonstrate that DETR-style transformers can be successfully translated from computer vision to the textual domain, treating generated spans as discrete objects to achieve high-fidelity localization. Simultaneously, the shared backbone matches or surpasses prior baselines on three binary-classification corpora, confirming that the learned representations are both *robust* and *transferable* across tasks of varying granularity.

Crucially, unlike methods constrained by sentence boundaries or explicit document structures, `GigaCheck` offers flexible, boundary-free detection. It operates effectively without predefined

Table 8: Classification performance on MAGE dataset in different scenarios including performance on the two challenging test sets. To test on challenging test sets (Unseen Domains & Unseen Model, Paraphrasing Attack) the model trained on Arbitrary-domains & Arbitrary-models dataset was used. Metrics for the Longformer (Beltagy et al., 2020) method was taken from the authors of MAGE dataset.

Methods	AvgRec	AUROC
Arbitrary-domains & Arbitrary-models		
Longformer	0.91	0.99
GigaCheck (Mistral-7B)	**0.96**	0.99
Unseen Domains & Unseen Model		
Longformer	0.76	0.94
GigaCheck (Mistral-7B)	**0.89**	0.96
Paraphrasing Attack		
Longformer	0.67	0.75
GigaCheck (Mistral-7B)	**0.69**	0.74
Out-of-distribution Detection: Unseen models		
Longformer	0.87	0.95
GigaCheck (Mistral-7B)	**0.92**	0.98

Table 9: Impact of backbone size on MAGE full set.

Model	AvgRec	AUROC
GigaCheck (Mistral-7B)	0.9611	0.9923
GigaCheck (Mistral-12B)	0.9630	**0.9941**
GigaCheck (Mistral-24B)	**0.9685**	0.9937
GigaCheck (Qwen-72B)	0.8338	0.9697

segmentation, showing strong generalization capabilities across diverse setups (from pre-trained to fine-tuned backbones) and in challenging out-of-domain scenarios.

7 Limitations

Context Window Constraints. To optimize computational efficiency during training, we explicitly restrict the input sequence length, although the backbone supports longer contexts. Consequently, documents exceeding this limit are processed in independent chunks, potentially obscuring long-range dependencies across segment boundaries. However, this is a hyperparameter choice; the core architecture scales naturally to larger context windows given sufficient computational resources.

Language Scope. This study is intentionally scoped to English to ensure rigorous comparison with established benchmarks. Since the unified backbone is multilingual by design, extending GigaCheck to other languages requires no architectural modifications, only the curation of appropriate training data.

Backbone Dependency. We report results using Mistral-7B due to its favourable quality-to-compute trade-off. However, the pipeline is model-agnostic; the framework permits swapping the backbone for any decoder-style LLM (e.g., LLaMA, Qwen) to adapt to specific resource constraints or domain requirements.

Benchmark Saturation. Near-perfect scores on smaller corpora like TuringBench may reflect their limited diversity rather than unsolved challenges. In datasets with few source domains and generator models, distinct artifacts persist, simplifying detection (Gritsai et al., 2024). Thus, these results may overstate real-world performance. To address this limitation, in concurrent work we assembled a substantially larger and more diverse benchmark and evaluated GigaCheck on it (Tolstykh et al., 2025).

8 Ethical Statement

Interpretability and Misuse. While GigaCheck improves transparency by localising specific AI-generated spans rather than providing a black-box document-level verdict, it does not achieve perfect accuracy. Performance can fluctuate based on the generator model, text length, and domain. Consequently, the detector should be used as an *assistive tool* for human verification, not as the sole basis for high-stakes decisions (e.g., academic disciplinary actions). We disclaim responsibility for any reputational damage or adverse consequences arising from the unverified reliance on its outputs.

References

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model generated text: Is chatgpt that easy to detect? *arXiv preprint arXiv:2306.05871*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Amrita Bhattacharjee and Huan Liu. 2024. Fighting fire with fire: can chatgpt detect ai-generated text? *ACM SIGKDD Explorations Newsletter*, 25(2):14–21.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33:*

Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection. *arXiv preprint arXiv:2304.04736*.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook ai's wmt20 news translation task submission. *arXiv preprint arXiv:2011.08298*.

Joseph Cutler, Liam Dugan, Shreya Havaldar, and Adam Stein. 2021. Automatic detection of hybrid human-machine text boundaries.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7:e443.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Aleksandr Gordeev, Vladimir Dokholyan, Irina Tolstykh, and Maksim Kuprashevich. 2024. Saliency-guided detr for moment retrieval and highlight detection.

Dijana Vukovic Grbic and Igor Dujlovic. 2023. Social engineering with chatgpt. In *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE.

German Gritsai, Anastasia Voznyuk, Andrey Grabovoy, and Yury Chekhovich. 2024. Are ai detectors good enough? a survey on quality of datasets with machine-generated texts. *arXiv preprint arXiv:2410.14677*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Hans WA Hanley and Zakir Durumeric. 2024. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 542–556.

Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, Badong Chen, and Xuguang Lan. 2024. Relation detr: Exploring explicit position relation prior for object detection. *arXiv preprint arXiv:2407.11699*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. 2022. Monodtr: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4012–4021.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. *arXiv preprint arXiv:2109.04825*.

Laida Kushnareva, Tatiana Gaintseva, Dmitry Abulkhanov, Kristian Kuznetsov, German Magai, Eduard Tulchinskii, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. Ai-generated text boundary detection with roft. In *First Conference on Language Modeling*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. 2022. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13619–13627.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. Mage: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

T Lin. 2017. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*.

Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. 2022. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023a. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023b. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt. *arXiv preprint arXiv:2306.05524*.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023c. On the detectability of chatgpt content: benchmarking, methodology, and evaluation through the lens of academic writing. *arXiv e-prints*, pages arXiv–2306.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, and Battista Biggio. 2023. The threat of offensive ai to organizations. *Computers Security*, 124:103006.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

WonJun Moon, Sangeek Hyun, SuBeen Lee, and Jae-Pil Heo. 2023. Correlation-guided query-dependency calibration for video temporal grounding.

OpenAI. 2023a. ChatGPT: A Large Language Model. Online; accessed February 13, 2024. Available at https://www.openai.com/.

OpenAI. 2023b. Gpt-4 technical report.

Mike Perkins, Jasper Roe, Darius Postma, James McGaughran, and Don Hickerson. 2023. Game of tones: faculty detection of gpt-4 generated content in university assessments. *arXiv preprint arXiv:2305.18081*.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666.

Sayak Saha Roy, Krishna Vamsi Naragam, and Shirin Nilizadeh. 2023. Generating phishing attacks using chatgpt. *arXiv preprint arXiv:2305.05133*.

V Sanh. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Chris Stokel-Walker. 2022. Ai bot chatgpt writes smart essays-should academics worry? *Nature*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

H Holden Thorp. 2023. Chatgpt is fun, but not an author.

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, and Maksim Kuprashevich. 2025. Llmtrace: A corpus for classification and fine-grained localization of ai-written text.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2024. Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, 36.

Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Toproberta: Topology-aware authorship attribution of deepfake texts. *arXiv preprint arXiv:2309.12934*.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis. *arXiv preprint arXiv:2305.18226*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023a. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023b. Llmdet: A third party large language models generated text detection tool. *arXiv preprint arXiv:2305.15004*.

Zhilin Yang. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhen Yin and Shenghua Wang. 2026. Span-level detection of ai-generated scientific text via contrastive learning and structural calibration. *Knowledge-Based Systems*, 334:115123.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Zijie Zeng, Shiqi Liu, Lele Sha, Zhuang Li, Kaixun Yang, Sannyuya Liu, Dragan Gašević, and Guanliang Chen. 2024a. Detecting ai-generated sentences in human-ai collaborative hybrid texts: Challenges, strategies, and insights.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guangliang Chen. 2024b. Towards automatic boundary detection for human-ai collaborative hybrid essay in education.

Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. 2021. Temporal query networks for fine-grained video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4486–4496.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and Lichao Sun. 2024. LLM-as-a-coauthor: Can mixed human-written and machine-generated text be detected? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 409–436, Mexico City, Mexico. Association for Computational Linguistics.

X Zhu, W Su, L Lu, B Li, X Wang, and J Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. arxiv 2020. *arXiv preprint arXiv:2010.04159*.

Zhuofan Zong, Guanglu Song, and Yu Liu. 2023. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758.

A Pre-trained VS fine-tuned models' embeddings

Table 10 presents a comparison of detection model performance on the RoFT and RoFT-ChatGPT datasets using two different setups. In the first experiment, we fine-tuned the Mistral-7B model to perform a text classification task with two labels: *'Human'* and *'AI-Human Collaborative'*, and used this model to extract text features for DETR model training. In the second experiment, we utilized the pre-trained Mistral-7B v0.3 model for feature extraction. Two DN-DAB-DETR models were then trained using these two types of features. The results indicate that the detection model performs better with features from the fine-tuned model; however, the model trained with text representations from the pre-trained model also achieves strong results on both datasets. We also provide results from Kushnareva et al. for comparison.

B Hyperparameters and experimental setup

We fine-tune Mistral-7B-v0.3[2] for a binary classification task to distinguish between human-written and machine-generated content using LoRA. Models training were done using Hugging Face Transformers[3] with bfloat16 precision. LoRA settings via the PEFT[4] library include: $r = 8$, *lora_alpha* = 16, *lora_dropout* = 0.1, and $bias =$ "$none$". Only query and value projection matrices in attention modules were adapted. We used AdamW (Loshchilov, 2017) with a cosine learning rate scheduler (Loshchilov and Hutter, 2016). The DETR model's encoder and decoder each had 3 layers. The loss weights were set to 10.0 for L1, 1.0 for gIoU, 4.0 for Focal Loss, 9.0 for denoised L1, and 3.0 for denoised gIoU.

During training, we augmented the data by randomly selecting between *'minimum sequence length'* to *'maximum sequence length'* tokens from each text. To optimize the models, we used the AdamW optimizer with a cosine learning rate schedule and also applied a weight for the *'Human'* category in the cross-entropy function. The dataset-specific hyperparameters used for the experiments are listed in the table 11.

When training a detection model to find LLM-generated intervals in text, we follow three steps: 1) fine-tune the Mistral-7B model on two or three categories, 2) extract features for the dataset from the trained model, 3) train the DETR model using extracted features as input data. The training is divided into three stages, firstly because this significantly speeds up the training process, and secondly because LLM and DETR models converge at different rates.

To train DN-DAB-DETR models, we also used the AdamW optimizer with a cosine learning rate schedule. During training we did not apply any text augmentations. The number of learnable queries N reflects the maximum span density per text in each dataset (see Section 5.1). The dataset-specific hyperparameters used for the experiments are listed in the table 12.

C MAGE comparison

Table 13 shows the results of comparing GigaCheck with Mistral-7B with all detectors considered by the authors of the MAGE dataset. We also report GigaCheck's performance on the MAGE full set (Arbitrary-domains & Arbitrary-models) using backbones of different sizes. We fine-tuned three large backbones: Mistral-Nemo-Base-2407[5] (12B), Mistral-Small-24B-Base-2501[6] (24B), and Qwen2.5-72B-Instruct[7] (72B).

D Mistral-7B-v0.3 zero-shot classification results

Table 14 presents the results of comparing GigaCheck with Mistral-7B fine-tuned with LoRA on five classification datasets against the Mistral-7B-Instruct-v0.3[8] model, evaluated in a zero-shot setting. The comparison was conducted on the test sets.

E Evaluation metrics for detection datasets

For each detection dataset, we compute specific metrics.

Followed the approach of the authors in Kushnareva et al., we compute mean squared error **MSE**$= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ between the predicted boundaries $\hat{y}$ and the true boundaries y, where a

[2]https://huggingface.co/mistralai/Mistral-7B-v0.3
[3]https://github.com/huggingface/transformers
[4]https://github.com/huggingface/peft
[5]https://huggingface.co/mistralai/Mistral-Nemo-Base-2407
[6]https://huggingface.co/mistralai/Mistral-Small-24B-Base-2501
[7]https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
[8]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Table 10: Boundary detection results on RoFT and RoFT-ChatGPT datasets. '†' denotes the DETR model was trained on text features from pre-trained Mistral-7B v0.3 model. **Bold** shows the best method, underlined - second best.

Method	**RoFT**			**RoFT-ChatGPT**		
	Acc	SoftAcc1	MSE	Acc	SoftAcc1	MSE
RoBERTa + SEP	49.64 %	79.71 %	2.63	54.61 %	79.03 %	3.06
RoBERTa	46.47 %	74.86 %	3.00	39.01 %	75.18 %	3.15
GigaCheck (DN-DAB-DETR)†	60.10 %	81.48 %	2.77	51.37 %	80.12 %	1.93
GigaCheck (DN-DAB-DETR)	**64.63** %	**86.68** %	**1.51**	**67.65** %	**88.98** %	**1.03**

Table 11: Hyperparameters for the classification experiments.

Parameter	**MAGE**	**TuringBench**	**TweepFake**
max sequence length	1024	1024	1024
minimum sequence length for augmentatoins	900	15	900
train batch size	64	32	32
gradient accumulation steps	1	2	2
learning rate	3e-4	3e-4	3e-4
cross entropy weight for human category	2	1	1
num train epochs	3	5	4
GPUs	1xNvidia H100	1xNvidia H100	1xNvidia H100
the fine-tuning time	48h	2h	2h

Table 12: Hyperparameters for the span-detection (DN-DAB-DETR) experiments.

Parameter	**RoFT**	**RoFT-ChatGPT**	**TriBERT**
number of queries	1	1	18
max sequence length	512	512	1024
train batch size	32	32	64
gradient accumulation steps	2	2	1
learning rate	1e-4	1e-4	2e-4
num train epochs	75	75	75
GPUs	1xNvidia H100	1xNvidia H100	1xNvidia H100
the DETR training time	5h	3h	6h
the Mistral fine-tuning time	3h	2h	(without fine-tuning)

boundary is the sentence number at which authorship in the text changes from human to LLM, and N represents the number of samples. It is worth noting that in both datasets from Kushnareva et al., each text contains no more than one boundary. The authors also propose reporting accuracy (**Acc**) of boundary detection and soft accuracy (**SoftAcc1**), the proportion of predictions that are off from the correct label by no more than one.

Finally, the authors of (Wang et al., 2024) evaluate model prediction quality using the mean absolute error **MAE**= $\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$, where $\hat{y}$ denotes the predicted word number that separates human and AI-generated parts of the text, y represents ground-truth word number, and N is the number of samples. The problem statement in (Wang et al., 2024) implies that there is only one such word boundary per text.

F1@K metric proposed by Zeng et al. (2024b) to asses the performance of model in boundaries detection task is described in Eq. 5. K was set to 3 for all measurements on TriBERT dataset.

$$F1@K = 2 \cdot \frac{|L_{topK} \cap L_{Gt}|}{|L_{topK}| + |L_{Gt}|} \tag{5}$$

F Interval post-processing

The DETR predictions are post-processed as follows for experiments on the **RoFT** and **RoFT-ChatGPT** datasets: let t_I be the start of the interval I, and $start_i$, end_i be the indexes of the first and last characters of the i-th sentence. If the i-th sentence contains t_I, the sentence number i', to which we map DN-DAB-DETR's prediction, is calculated as follows:

Table 13: Classification performance on MAGE dataset in different scenarios including performance on the two challenging test sets. To test on challenging test sets the model trained on Arbitrary-domains & Arbitrary-models dataset was used.

Methods	HumanRec	MachineRec	AvgRec	AUROC
Arbitrary-domains & Arbitrary-models				
FastText (Joulin et al., 2016)	86.34%	71.26%	78.80%	0.83
GLTR (Gehrmann et al., 2019)	12.42%	98.42%	55.42%	0.74
DetectGPT (Mitchell et al., 2023)	86.92%	34.05%	60.48%	0.57
Longformer (Beltagy et al., 2020)	82.80%	98.27%	90.53%	0.99
GigaCheck (Mistral-7B)	95.72%	96.49%	96.11%	0.99
GigaCheck (Mistral-12B)	95.29%	97.32%	96.30%	0.99
GigaCheck (Mistral-24B)	96.94%	96.76%	**96.85%**	0.99
GigaCheck (Qwen-72B)	83.38%	96.62%	83.38%	0.97
Unseen Domains & Unseen Model				
FastText (Joulin et al., 2016)	71.78%	68.88%	70.33%	0.74
GLTR (Gehrmann et al., 2019)	16.79%	98.63%	57.71%	0.73
Longformer (Beltagy et al., 2020)	52.50%	99.14%	75.82%	0.94
GigaCheck (Mistral-7B)	79.71%	97.38%	**88.54%**	0.96
Paraphrasing Attack				
FastText (Joulin et al., 2016)	71.78%	50.00%	60.89%	0.66
GLTR (Gehrmann et al., 2019)	16.79%	82.44%	49.61%	0.47
Longformer (Beltagy et al., 2020)	52.16%	81.73%	66.94%	0.75
GigaCheck (Mistral-7B)	79.66%	58.24%	**68.95%**	0.74
Out-of-distribution Detection: Unseen models				
FastText (Joulin et al., 2016)	83.12%	54.09%	68.61%	0.74
GLTR (Gehrmann et al., 2019)	25.77%	89.21%	57.49%	0.65
DetectGPT (Mitchell et al., 2023)	48.67%	75.95%	62.31%	0.60
Longformer (Beltagy et al., 2020)	83.31%	89.90%	86.61%	0.95
GigaCheck (Mistral-7B)	95.65%	89.00%	**92.32%**	0.98

Table 14: Experimental results (F1 scores) on the test sets for classification datasets. F1 is calculated for the machine-generated category. We compare the Mistral-7B-Instruct-v0.3 model evaluated in a zero-shot setting with fine-tuned Mistral-7B-v0.3 models.

Method	TweepFake	TuringBench FAIR_wmt20	TuringBench GPT-3	MAGE
Mistral-7B-Instruct-v0.3	0.640	0.537	0.500	0.633
GigaCheck (Mistral-7B)	0.942	0.997	0.971	0.96

$$i' = i + \begin{cases} 1, & \text{if } t_I \geq \frac{start_i + end_i}{2}, \\ 0, & \text{if } t_I < \frac{start_i + end_i}{2}. \end{cases} \quad (6)$$

For the **TriBERT** experiments, DETR predictions undergo the following post-processing steps: let b_i and b_{i+1} denote the beginnings of the n and $n+1$ sentences in characters and let p_j denote the beginning or the end of the predicted interval in characters. Then the boundary B for p_j is calculated as:

$$B(p_j) = \begin{cases} b_i & \text{if } p_j < \frac{b_i + b_{i+1}}{2}, \\ b_{i+1} & \text{if } p_j \geq \frac{b_i + b_{i+1}}{2}. \end{cases} \quad (7)$$

Therefore, if the predicted start or end of the interval falls in the first half of sentence n, we map it to the beginning of sentence n. If it falls in the second half, we map it to the beginning of the next sentence, $n+1$. As a result, each boundary determines the sentence number where the text's authorship changes. Note that if a boundary is equal to the beginning or the end of the whole text, we remove it, since a boundary can only be between two sentences.

G Examples of the DETR model output

Tables 15 and 16 present examples of work of the model trained on the RoFT-ChatGPT dataset. Table 15 shows the ground truth and output result for test samples from the *'Short Stories'* and *'New York Times'* domains. Table 16 shows the ground truth and output result for test samples from the *'Recipes'* and *'Presidential Speeches'* domains.

Table 15: Examples from the test set of the raw model's output, trained on the RoFT-ChatGPT dataset. **Bold** text indicates either the ground truth interval or the predicted one.

Domain: *Short Stories*
GT: Aryton blinked and rubbed his head. It had been a very high speed crash. He expected the impact to hurt more, but the whole thing just felt quite... fuzzy. There didn't seem to be any track marshals around, which was odd, Aryton looked back towards the corner where he'd lost control. Nothing there, he pulled himself out of the car and scurried over the crash barrier to safety. That's funny, he thought as he looked back at the crash, the car doesn't seem damaged. **Aryton walked back towards his car and inspected it closely. It was as if the crash had never happened, there wasn't a scratch on it. He checked the fuel gauge, it was full, and the tires were still warm to the touch. It was a brand new car and one of the fastest ones that he had ever driven.**
Output: Aryton blinked and rubbed his head. It had been a very high speed crash. He expected the impact to hurt more, but the whole thing just felt quite... fuzzy. There didn't seem to be any track marshals around, which was odd, Aryton looked back towards the corner where he'd lost control. Nothing there, he pulled himself out of the car and scurried over the crash barrier to safety. That's funny, he thought as he looked back at the crash, the car doesn't seem da**maged. Aryton walked back towards his car and inspected it closely. It was as if the crash had never happened, there wasn't a scratch on it. He checked the fuel gauge, it was full, and the tires were still warm to the touch. It was a brand new car and one of the fastest ones that he had ever drive**n.
Domain: *New York Times*
GT: ... For many in the industry, it was the final seal of approval on a technology that remained controversial as long as it was exclusive to smaller, less conservative computer makers. But that interpretation does not sit well with Irving Wladawsky-Berger, who is responsible for the supercomputing business at the International Business Machines Corporation. " For me to say now we've finally put our seal of approval on this would sound supremely arrogant," he said. " Let's just say we have committed to build a product family of parallel RISC systems that scale up from our RS/6000." RISC, or reduced instruction set computing, is a technology that speeds processing by relegating more tasks to software; the RS/6000 is the name for both a chip set and a computer work station produced by I.B.M. using RISC. Dr. Wladawsky-Berger said the impetus to create a massively parallel supercomputer came from RS/6000 customers who were creating a sort of virtual parallel processor by linking multiple work stations. " There were people pushing at I.B.M., but they were pushing in many different directions," he said. " Supercomputing is an area where if you get seven smart people together, you get 17 different architectures." " **But," he added, "we knew we had to do something because we were seeing more and more of our customers doing this and we knew we had to provide them with a scalable solution.**
Output: ... For many in the industry, it was the final seal of approval on a technology that remained controversial as long as it was exclusive to smaller, less conservative computer makers. But that interpretation does not sit well with Irving Wladawsky-Berger, who is responsible for the supercomputing business at the International Business Machines Corporation. " For me to say now we've finally put our seal of approval on this would sound supremely arrogant," he said. " Let's just say we have committed to build a product family of parallel RISC systems that scale up from our RS/6000." RISC, or reduced instruction set computing, is a technology that speeds processing by relegating more tasks to software; the RS/6000 is the name for both a chip set and a computer work station produced by I.B.M. using RISC. Dr. Wladawsky-Berger said the impetus to create a massively parallel supercomputer came from RS/6000 customers who were creating a sort of virtual parallel processor by linking multiple work stations. " There were people pushing at I.B.M., but they were pushing in many different directions," he said. " Supercomputing is an area where if you get seven smart people together, you get 17 different architectures." " But," he added, "we **knew we had to do something because we were seeing more and more of our customers doing this and we knew we had to provide them with a scalable soluti**on.

Table 16: Examples from the test set of the raw model's output, trained on the RoFT-ChatGPT dataset. **Bold** text indicates either the ground truth interval or the predicted one.

Domain: *Recipes*
GT: HOW TO MAKE: Make-Ahead Turkey Gravy Ingredients: 2 tablespoons canola oil 2 lbs turkey wings 1 cup dry white wine 3 tablespoons olive oil 1 medium yellow onion, halved 2 carrots, cut in 2 inch pieces 2 celery ribs, cut in 2 inch pieces plus a handful of the celery leaves 1 head garlic, cut in half 2 sprigs fresh thyme 2 sprigs fresh sage 2 sprigs fresh rosemary 10 black peppercorns 2 bay leaves 6 cups low sodium chicken broth 8 tablespoons flour 4 tablespoons butter, if needed 12 teaspoon white vinegar Kitchen Bouquet, if desired. **Instructions: 1. Preheat the oven to 375F.2. In a large roasting pan, toss the turkey wings with canola oil.3. Roast the turkey wings for about 1 hour, or until deeply golden brown.4. Transfer the turkey wings to a large pot and pour in the white wine.5. Over medium-high heat, bring to a simmer and scrape up any browned bits from the bottom of the roasting pan.6. Simmer for about 5 minutes, or until the wine has reduced by half.7. Pour the wine mixture over the turkey wings and set aside.8. In a large skillet, heat the olive oil over medium heat.9.**
Output: HOW TO MAKE: Make-Ahead Turkey Gravy Ingredients: 2 tablespoons canola oil 2 lbs turkey wings 1 cup dry white wine 3 tablespoons olive oil 1 medium yellow onion, halved 2 carrots, cut in 2 inch pieces 2 celery ribs, cut in 2 inch pieces plus a handful of the celery leaves 1 head garlic, cut in half 2 sprigs fresh thyme 2 sprigs fresh sage 2 sprigs fresh rosemary 10 black peppercorns 2 bay leaves 6 cups low sodium chicken broth 8 tablespoons flour 4 tablespoons butter, if needed 12 teaspoon white vinegar Kitchen Bouquet, if desired**. Instructions: 1. Preheat the oven to 375F.2. In a large roasting pan, toss the turkey wings with canola oil.3. Roast the turkey wings for about 1 hour, or until deeply golden brown.4. Transfer the turkey wings to a large pot and pour in the white wine.5. Over medium-high heat, bring to a simmer and scrape up any browned bits from the bottom of the roasting pan.6. Simmer for about 5 minutes, or until the wine has reduced by half.7. Pour the wine mixture over the turkey wings and set aside.8. In a large skillet, heat the olive oil over medium heat**.9.
Domain: *Presidential Speeches*
GT: "An Association of Nations" by President Warren G. Harding on July 22, 1920. My countrymen, we believe the unspeakable sorrows, the immeasurable sacrifices, the awakened convictions, and the aspiring conscience of humankind must commit the nations of the earth to a new and better relationship. It need not be discussed now what motives plunged the world into war. It need not be inquired whether we asked the sons of this republic to defend our national rights, as I believe we did, or to purge the Old World of the accumulated ills of rivalry and greed. The sacrifices will be in vain if we cannot acclaim a new order with added security to civilization and peace maintained. One may readily sense the conscience of our America. I am sure I understand the purpose of the dominant group of the Senate. We were not seeking to defeat a world aspiration. **We were not seeking to withhold our country from doing its part in the world's great work. We were seeking only to safeguard our own sovereignty and to enter into any relationship with other nations only after full and free discussion and deliberation.**
Output: "An Association of Nations" by President Warren G. Harding on July 22, 1920. My countrymen, we believe the unspeakable sorrows, the immeasurable sacrifices, the awakened convictions, and the aspiring conscience of humankind must commit the nations of the earth to a new and better relationship. It need not be discussed now what motives plunged the world into war. It need not be inquired whether we asked the sons of this republic to defend our national rights, as I believe we did, or to purge the Old World of the accumulated ills of rivalry and greed. The sacrifices will be in vain if we cannot acclaim a new order with added security to civilization and peace maintained. One may readily sense the conscience of our America. I am sure I understand the purpose of the dominant group of the Senate. We were not seeking to defeat a world aspiration. We were not seeking to withhold our country **from doing its part in the world's great work. We were seeking only to safeguard our own sovereignty and to enter into any relationship with other nations only after full and free discussion and deliberati**on.