Fine-Grained Detection of AI-Generated Text Using Sentence-Level Segmentation

L. D. M. S. Sai Teja[1] **Annepaka Yadagiri**[1] **Partha Pakray**[1]

Chukhu Chunka[1] **Mangadoddi Srikar Vardhan**[1]

[1]National Institute of Technology Silchar

{lekkalad_ug_22, annepaka22_rs, partha}@cse.nits.ac.in

{chukhu, mangadoddis_ug_22}@cse.nits.ac.in

Abstract

Generation of Artificial Intelligence (*AI*) texts in important works has become a common practice that can be used to misuse and abuse AI at various levels. Traditional AI detectors often rely on document-level classification, which struggles to identify AI content in hybrid or slightly edited texts designed to avoid detection, leading to concerns about the model's efficiency, which makes it hard to distinguish between human-written and AI-generated texts. A sentence-level sequence labeling model proposed to detect transitions between human- and AI-generated text, leveraging nuanced linguistic signals overlooked by document-level classifiers. By this method, detecting and segmenting AI and human-written text within a single document at the token-level granularity is achieved. Our model combines the state-of-the-art pre-trained Transformer models, incorporating Neural Networks (*NN*) and Conditional Random Fields (*CRFs*). This approach extends the power of transformers to extract semantic and syntactic patterns, and the neural network component to capture enhanced sequence-level representations, thereby improving the boundary predictions by the CRF layer, which enhances sequence recognition and further identification of the partition between Human- and AI-generated texts. The evaluation is performed on two publicly available benchmark datasets containing collaborative human and AI-generated texts. Our experimental comparisons are with zero-shot detectors and the existing state-of-the-art models, along with rigorous ablation studies to justify that this approach, in particular, can accurately detect the spans of AI texts in a completely collaborative text. All our source code and the processed datasets are available in our GitHub repository[1].

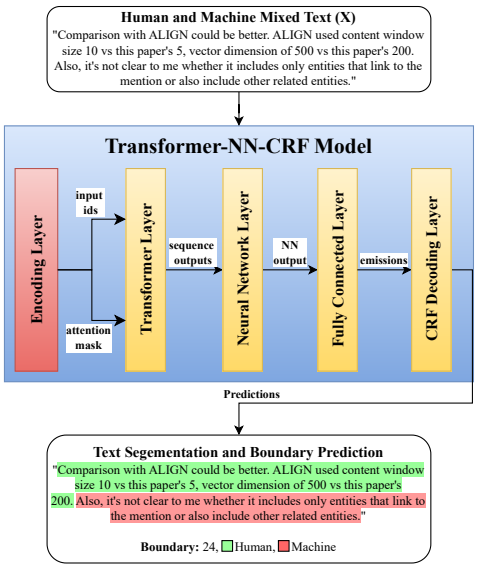


Figure 1: Text Segmentation between the Human and Machine with boundary prediction. The text in Green is Human written text, and the text in Red is Machine Generated Text.

1 Introduction

Recent advancements in Large Language Models (LLMs) (Chang et al., 2024; Annepaka and Pakray, 2025), such as ChatGPT[2], Grok[3], Gemini[4], and DeepSeek[5], have demonstrated exceptional performance in Natural Language Processing (NLP) (Patwardhan et al., 2023). These models are capable of generating highly fluent and realistic human-like text, significantly enhancing applications such as text classification (Dogra et al., 2022; Kowsari et al., 2019), sentiment analysis (Nasukawa and Yi, 2003), machine translation (Stahlberg, 2020), and question-answering systems (He et al., 2024; Al-

[1]https://github.com/saitejalekkala33/GenAI_Detect_Sentence_Level

[2]https://chatgpt.com/

[3]https://grok.com/

[4]https://gemini.google.com/

[5]https://chat.deepseek.com/

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 814–828

lam and Haggag, 2012). These foundational models demonstrate significant potential in addressing a wide spectrum of NLP tasks, ranging from Natural Language Understanding (*NLU*) to Natural Language Generation (*NLG*), and even contributing to the development of Artificial General Intelligence (AGI) (Yang et al., 2024). AI-plagiarism has been a growing concern in the modern world, where the distinction between human-generated and AI-generated work has become increasingly subtle. Due to the rapid growth of AI-generation tools and LLMs, academic integrity and content originality have become significant challenges. Detecting AI generation is a significant challenge for modern researchers. Traditional plagiarism detection systems, such as Turnitin, Scribbr, etc., have fought this. Generally, detection systems such as the above systems identify plagiarism by using Machine Learning (ML) algorithms to compare the user-written text against a vast database of online content and highlight the parts that closely resemble the existing material; however, this system is often bypassed using simple techniques like paraphrasing or restructuring the sentence without altering the core meaning. This is generally due to the nature of detection methods, where general detection methods, such as perplexity-based methods, are geared towards incremental or document-level modification. This has become a significant challenge that requires great efforts from NLP researchers worldwide.

To address the above-stated problems, we propose a novel integration of a pre-trained Transformer encoder with Neural Networks and a Conditional Random Field (*CRF*) layer, specifically designed for boundary-aware sentence-level authorship segmentation. Unlike prior CRF pipelines, our architecture introduces dynamic dropout and hierarchical loss-aware training, tailored to mixed-authorship detection. Unlike traditional Transformer-CRF pipelines applied to NER or POS tagging, we adapt this architecture for fine-grained authorship segmentation, incorporating sequence-aware regularization and token boundary-aware loss tailored for mixed-authorship text, a novel problem space. Our contribution lies in the hierarchical integration of authorial cues at different granularity levels, use of dynamic dropout to mitigate overfitting to local styles, and a custom-designed loss that emphasizes boundaries. To our knowledge, no existing model has applied these optimizations specifically for sentence-level segmentation in human-AI collaborative texts.

2 Related Work

Based on the evolution of AI models, this mixed text data is prone to plagiarism and authenticity issues. There have been works based on the mixed text data classification by RoFT (Dugan et al., 2020) created a dynamic benchmark where users attempted to identify transition points between human and AI texts. In the follow-up RoFT-ChatGPT (Kushnareva et al., 2023) adapted SOTA detectors to locate the boundaries and compared the perplexity-based approaches. The work *'Towards Automatic Boundary Detection for Human-AI Collaborative Hybrid Essay in Education'* by (Zeng et al., 2024) introduced a new segmentation approach that learns distinct human and AI authorship prototypes within an embedding space, while treating the boundaries as points of maximum prototype distance. SemEval 2024 shared task by (Wang et al., 2024), in which many participated for the task of detecting the boundaries, in that Qu and Meng (2024) (*TM-TREK*) achieved 1st place in the human-machine mixed-text detection subtask by reframing boundary detection as token classification and leveraging ensembles of LLMs with segmentation loss. Another work, SeqXGPT (Wang et al., 2023), introduced a sequence labeling approach using GPT-style decoder-only models as the core features.

With the rapid advancement of LLMs, collaborative writing between humans and AI has become increasingly seamless. For instance, previous work by Buschek et al. (2021) explored the use of GPT-2 to provide phrase-level writing suggestions during email composition, aiming to support and enhance human writing. This tells about the growing relevance of detecting and understanding hybrid texts that combine contributions from both humans and AI. Lee et al. (2022) extended this line of work by employing the more advanced GPT-3 model to offer sentence-level suggestions in human-AI collaborative essay writing. As this type of collaborative writing between humans and LLMs becomes more prevalent, it introduces a critical challenge for the AI text detection field: identifying AI-generated segments within jointly authored text. Addressing this issue, Dugan et al. (2023) reframed the task of detecting AI text in hybrid documents as a boundary detection problem, aiming to precisely locate the transition points between human and

AI-generated text. The study specifically evaluated human ability to identify boundaries within hybrid texts containing a single transition point. Although detection accuracy improved with some level of training, overall performance remained limited; participants were only able to correctly identify the boundary in 23.4% of the cases. Building upon the work of Dugan et al. (2023), our research also focuses on boundary detection. However, key differences include: (1) this work explores automated methods for identifying boundaries, and (2) the hybrid texts used in our experiments contain multiple boundaries.

OUR KEY CONTRIBUTIONS:

1. CRF tagged Transformer Models.
2. Layer-wise Learning Rate Decay.
3. Dynamic Dropout.
4. Sequence Predictions via CRF Decoding.
5. Xavier Initialization for Weights Stability.
6. CRF Loss Calculation with Masking.
7. Multiple Boundary Prediction Flexibility.

3 Dataset Descriptions

In this section, we used two publicly available datasets: (1) TriBERT from the paper *Towards Automatic Boundary Detection for Human-AI Collaborative Hybrid Essay in Education* (Zeng et al., 2024) and (2) *M4GT-Bench Task 3: Mixed Human-Machine Text Detection* (Wang et al., 2024).

TriBERT: This dataset contains various mixed author[6] sequences such as *"H-M", "M-H", "H-M-H", "M-H-M", "H-M-H-M-H" and "M-H-M-H-M"*, totaling six types, as seen in the Table 10.

M4GT: This dataset includes only the ChatGPT and LLaMA reviews as shown in Table 11. It follows a single pattern, *"H Initiated and M Ended"*, and provides word-level boundaries indicating the exact transition point. The dataset statistics, such as train, dev, and test splits, can be seen in Table 1. Further detailed dataset descriptions are given in Appendix A.

4 Proposed Methodology

The methodology we propose integrates a hierarchical architecture that combines contextual encoding, sequential pattern modeling, and prediction to perform fine-grained sequence labeling.

[6]H-Human, M-Machine

Dataset	Train	Dev	Test
TriBERT (Zeng et al., 2024)	12,049	2,527	2,560
M4GT (Wang et al., 2024)	18,245	2,525	11,123

Table 1: Dataset statistics from TriBERT (Zeng et al., 2024) and M4GT (Wang et al., 2024) showing the number of instances in train, dev, and test splits.

4.1 Problem Formulation

Given a hybrid text paragraph $\mathcal{T} = \langle s_1, \ldots, s_n \rangle$, where sentences s_i are human-initiated and machine-ended or mixed in *HM, MH, HMH, MHM, HMHMH, and MHMHM* setting, *boundary detection* identifies token-level authorship transitions. Unlike sentence-level detection, we target sub-sentence boundaries, using a token sequence $\mathcal{W} = \langle w_1, \ldots, w_m \rangle$ to predict $\mathcal{Y} = \langle y_1, \ldots, y_m \rangle$, with $y_j = 1$ for AI tokens and $y_j = 0$ for humans. This supports fine-grained segmentation of human-AI collaborative text.

4.2 Model Description

The flow of input token sequence processing begins with a pretrained Transformer-based encoder to produce rich contextual representations, and also for capturing both local and global dependencies in the overall sequence. These embeddings are then passed through a neural network layer, enabling the model to further refine token-level features by explicitly modeling sequential patterns in both forward and backward directions (a bidirectional mode). The output of the sequential layer is fed into a linear classification head that projects each token representation into a distribution over target labels. To enforce global consistency in the predicted label sequences and effectively capture dependencies among adjacent labels, the model employs a Conditional Random Field (*CRF*) as the final decoding layer, and the reason for choosing the CRF is discussed in the following Section 4.3. During training, the model tries to minimize the negative log-likelihood of the correct label sequence under the CRF, while during inference time, it uses *Viterbi* decoding to predict or output the most likely sequence of labels. While this combined-component design yields powerful representational capacity, it also increases the complexity of the model architecture and creates a chance for overfitting.

Model Optimization: To stabilize the above concerns, several optimization techniques are incorpo-

rated into the model, those are 1) including *Layer-wise Learning Rate Decay* (*LLRD*) to stabilize fine-tuning across the layers, 2) *Dynamic Dropout* to regularize learning by adapting to training dynamics, and 3) *Xavier initialization* to ensure controlled variance propagation in the linear classification layer. This hybrid-hierarchical architecture with the optimization techniques allows the model to leverage deep contextual knowledge, sequential structure, and label inter-dependencies, making it particularly suitable for the segmentation of human and AI texts while considering the spans of the predicted labels, and it can be said that this model is well-suited for the structured prediction tasks in natural language processing. A clear text segmentation and boundary prediction workflow is shown in Figure 1.

4.3 Why CRFs?

Conditional Random Fields (*CRFs*) are discriminative models well suited to prediction tasks, particularly sequence labeling. Unlike Hidden Markov Models (*HMMs*) and Maximum Entropy Markov Models (*MEMMs*), CRFs avoid issues like label bias by modeling the conditional probability of label sequences directly.

We employ a standard linear-chain CRF layer to model inter-label dependencies. For mathematical formulation, refer Lample et al. (2016). For a given sequence input $x = (x_1, x_2, \ldots, x_n)$ and label sequence $y = (y_1, y_2, \ldots, y_n)$, CRF defines the probability as follows:

$$P(y \mid x) = \frac{\exp\left(\sum_{t=1}^{n} \psi_t(y_t, x, t) + \sum_{t=1}^{n-1} \phi_t(y_t, y_{t+1}, x)\right)}{Z(x)} \tag{1}$$

where: $\psi_t(y_t, x, t)$: The score function for assigning label y_t to the t-th token in x. $\phi_t(y_t, y_{t+1}, x)$: The score function for the transition between labels y_t and y_{t+1}. $Z(x)$: The partition function, normalizing the probabilities:

$$Z(x) = \sum_{y' \in \mathcal{Y}(x)} \exp\left(\sum_{t=1}^{n} \psi_t(y_t, x, t) + \sum_{t=1}^{n-1} \phi_t(y_t, y_{t+1}, x)\right) \tag{2}$$

CRF Score Function: The unnormalized score $S(x, y)$ for a given sequence y is:

$$S(x, y) = \sum_{t=1}^{n} \psi_t(y_t, x, t) + \sum_{t=1}^{n-1} \phi_t(y_t, y_{t+1}, x) \tag{3}$$

CRFs vs HMMs. Hidden Markov Models (*HMMs*) represent observed data as a sequence of events, where each observation depends on a hidden state in a hidden Markov chain. Compared to CRFs, HMMs impose more significant constraints, as each state relies on a fixed set of previous hidden states, allowing them to capture local context effectively. In contrast, CRFs incorporate both local and global contexts more comprehensively using feature functions. For example, the feature function label the first word as a verb if the sequence ends with a question mark and this isn't possible with HMMs. As a result, CRFs can be more accurate than HMMs if we define the right feature functions.

CRFs vs MEMMs. MEMM combines HMM with the Maximum Entropy (log-linear) classifiers. Unlike generative models like HMMs, MEMMs are discriminative, but they suffer from the 'label bias problem', where states with fewer outgoing transitions may ignore observations. CRFs addresses them by modeling joint probability distributions to capture dependencies between labels while mitigating bias. This makes CRFs particularly suitable for sentence-level classification tasks.

5 Experiments

Experimental Setup: We conducted all our experiments on an Amazon Web Services (AWS) cloud server. In the EC2 instance, we initiated an instance for Accelerated Computing. The specifications are **g6e.xlarge** instance, which provides 3rd generation AMD EPYC processors (AMD EPYC 7R13), with a **NVIDIA L40S Tensor Core GPU with 48 GB GPU memory**, and 4x vCPU with 150 GiB memory, and our OS type is Ubuntu Server 24.04 LTS (HVM).

We conducted our experiments by combining CRF layer to the sequential models starting from the Neural Networks to the Transformer models, where the neural network models we have taken are Convolutional Neural Network (*CNN*), Recurrent Neural Network (*RNN*), Long Short-Term Memory (*LSTM*), Bidirectional LSTM (*BiLSTM*), and Bidirectional Gated Recurrent Unit (BiGRU), and the transformer models we have taken are the BERT (Devlin, 2018), DistilBERT (Sanh, 2019), RoBERTa (Liu, 2019), DeBERTa (He et al., 2020) and ModernBERT (Warner et al., 2024), and our proposed architecture is the combination of a Transformer, a Neural Network and a CRF layer. We conducted several combinations of the above models. As we mentioned, the emergence of CRF with the prior models HMMs and MEMMs, we

experimented with these by replacing the CRF with HMM and MEMM in the best-performing model with CRF. All our experiments can be seen in the Table 3, and the hyperparameters can be seen in Table 2.

Hyperparameter	Value
batch_size	32
epochs	3
gradient_clip	1.0
hidden_dim	512
num_layers	3
num_labels	2
weight_decay	1e-2
max_len	512 tokens
learning_rate	1e-6, 5e-6, 1e-5, 1e-4

Table 2: Hyperparameters

Type	Models
Neural Network + CRF	CNN, RNN, LSTM, BiLSTM, **BiGRU**
Transformer + CRF	BERT, DistilBERT , RoBERTa, **DeBERTa**, ModernBERT
Transformer + NN + CRF	DeBERTa + CNN, DeBERTa + RNN, DeBERTa + LSTM, DeBERTa + BiLSTM, **DeBERTa + BiGRU**, BERT + BiGRU, DistilBERT + BiGRU, RoBERTa + BiGRU, ModernBERT + BiGRU
T + NN + HMM/MEMM	DeBERTa + BiGRU + HMM, DeBERTa + BiGRU + MEMM

Table 3: All our experimental combinations with the Neural Networks and the transformer models, the best performing models in each setting are highlighted in Bold.

5.1 Comparisons With SOTA Techniques

We have compared our approach's best performing model with the zero-shot training free models and others with the prior models that were experimented on these datasets previously, and lastly with the HMM and MEMM models. In total they are:

1. **Zero-Shot methods for both M4GT and TriBERT datasets**
   (a) FastDetectGPT - '*falcon-7b-instruct*'
   (b) FastDetectGPT - '*gpt-neo-2.7b*'
   (c) Glimpse - '*davinci-002*'
   (d) Glimpse - '*babbage-002*'
   (e) Binoculars
2. **HMM & MEMM for both M4GT and TriBERT datasets**
   (a) Best Performing Model[7] + HMM
   (b) Best Performing Model + MEMM
3. **For TriBERT** - Zeng et al. (2024)
   (a) TriBERT (p=2) - (Zeng et al., 2024)
   (b) GigaCheck (DN-DAB-DETR) - (Tolstykh et al., 2024)
4. **For M4GT** - Wang et al. (2024)
   (a) Longformer - (Wang et al., 2024)
   (b) DeBERTa-V3 - (Wang et al., 2024)
   (c) TM-TREK - (Qu and Meng, 2024)
   (d) AIpom - (Qu and Meng, 2024)
   (e) USTC-BUPT - (Guo et al., 2024)

[7]Best Performing Model - This is the model with the CRF that has the best results in all metrics.

5.2 Zero-Shot Methods

As mentioned above, we have utilized the zero-shot models for both datasets. The zero-shot models are: 1) **Fast-DetectGPT** (Bao et al., 2023; Mitchell et al., 2023), with two variations: (a) *falcon-7b/falcon-7b-instruct*, where `falcon-7b` is used as the sampling model and `falcon-7b-instruct` as the scoring model, and (b) *gpt-neo-2.7b*, where the same model is used as both the sampling and scoring models. Fast-DetectGPT is built to detect AI-generated text by giving the perturbing samples and evaluating the text's likelihood under different model prompts, which makes the model highly adaptable when combined with different Language Models. 2) **Glimpse** (Bao et al., 2023), with two additional variations: (a) *davinci-002* and (b) *babbage-002*, where each model is used solely as a scoring model. Glimpse works on the principle of uncertainty-guided token perturbation, where this methods makes the model effective at identifying unnatural patterns in text. 3) The final detector is **Binoculars** (Hans et al.), which performs fine-grained AI detection by comparing representations between base- and instruction-tuned language models. To adapt these zero-shot methods for sentence-level segmentation, we apply them individually to each sentence using standardized prompts and aggregate the outputs.

5.3 HMM and MEMM

Training follows a two-stage process. The neural front-end (Transformer+NN) is trained conventionally using a token-level negative log-likelihood (cross-entropy) loss. This fine-tunes the network to produce meaningful emission probabilities. The HMM itself, implemented using hmmlearn.GaussianHMM is trained in an unsupervised

manner. It is fitted on the sequence of emission probability vectors generated from the first training batch. This step learns the initial state probabilities and the state transition matrix (A) based on the data's sequential patterns, as well as the parameters for the Gaussian emission model (B). During inference, the trained neural front-end generates emission probabilities for a given text sequence. The fitted HMM then uses the Viterbi algorithm (hmm.predict()) to find the most likely sequence of hidden states (labels) that could have generated the observed emission probabilities.

The MEMM's effectiveness is from its ability to use rich, overlapping features. For each token t in a sequence, we construct a feature vector by concatenating: The training: The MEMM's LR classifier is trained on-the-fly within each training batch to learn the conditional probability $P(y_t|y_{t-1}, obeservations_t)$. This classifier learns to predict the current label based on the rich feature vector described above. Concurrently, the neural front-end is trained using a weighted negative log-likelihood loss to optimize the quality of the emission probabilities it generates. Decoding is performed using a greedy, step-by-step approach. For the first token, the prediction is based only on the observation features. For every subsequent token t, the feature vector is constructed using the observation at t and the label predicted for token t-1. The trained logistic regression model then predicts the label for token t, and this process repeats for the entire sequence. This approach allows the model to leverage local label dependencies, which is the core principle of an MEMM.

5.4 Existing SOTA Models

TriBERT model that is proposed by Zeng et al. (2024), is a boundary detection model that uses a tri-branch structure for identifying human and AI segments in hybrid essays. GigaCheck (Tolstykh et al., 2024) uses a DeNoising - Dynamic Anchor Box - DEtection TRansformer (*DN-DAB-DETR*) architecture designed to detect LLM-generated text using dense attention and boundary-aware features. M4GT (Wang et al., 2024) benchmarked a variety of detectors, including Longformer, which uses long-range attention to identify stylistic shifts and DeBERTa-V3, a strong pretrained model adapted for segment-wise detection, TM-TREK (Qu and Meng, 2024), which uses LLMs with a temporal-aware retrieval strategy, AIpom (Shirnin et al., 2024) includes prompt engineering and contrastive learning, and the last one, USTC-BUPT (Guo et al., 2024), enhances detection by aligning domain adversarial learning with LLM-derived features.

6 Evaluation Metrics

The evaluation metrics were considered separately for each of the datasets. For the TriBERT dataset, we utilized the same metrics that were given in the paper by Zeng et al. (2024) which is the F1 score. The authors of the paper (Zeng et al., 2024) considered two things a) L_{topK} and b) L_{Gt}, which give top-K boundaries detected by the model's algorithm, and the number of ground-truth boundaries respectively, and the value of K is set to 3 in their analysis. The F1 score is then determined using the following formula:

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

For the M4GT dataset, the authors mentioned the Mean Absolute Error (MAE) of the actual boundary y_i and the predicted boundary $\hat{y}_i$ as the main metric and n as total boundary pairs, so we have taken the same as for the comparison purpose.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{5}$$

Besides these official metrics, we also computed standard evaluation metrics such as Accuracy, Precision, Recall, F1-Score, Matthews correlation coefficient (*MCC*), and Cohen's Kappa score. These metrics are computed based on the token-level prediction, as the spans or boundaries are determined by the binary labels (0-'human' or 1-'AI') of token sequences. Each token in the dataset is labeled as 0 or 1 for the CRF computation during the training, and during the inference, these labels will be predicted using the Viterbi algorithm of the CRF, such that these are computed whether the prediction is 0 or 1. The Viterbi algorithm finds the sequence $\hat{y}$ with the maximum score as in Equation 6 and the recursive formulation used in the Viterbi algorithm is showed in the below Equation 7.

$$\hat{y} = \arg \max_{y \in \mathcal{Y}(x)} S(x, y) \tag{6}$$

$$\delta_t(y_t) = \max_{y_{t-1}} [\delta_{t-1}(y_{t-1}) + \phi_t(y_{t-1}, y_t, x) + \psi_t(y_t, x, t)] \tag{7}$$

where s(x, y) is the CRF score function which can be seen in the Equation 3, $\delta_t(y_t)$ is the maximum score of sequences ending in y_t at position t.

7 Results Analysis

We conclude that our method overcame not only the results of the existing zero-shot methods but also the models that were previously experimented on these datasets. We hypothesize that our model's superior performance is due to its explicit modeling of inter-sentence transitions using neural networks in between, which captures stylistic changes more effectively than token-level detectors. Zero-shot methods underperformed in comparison to our proposed model, indicating limited reliability when applied in real-world hybrid texts. On the M4GT dataset, our best-performing model (DeBERTa + BiGRU + CRF with all optimizations) achieved a MAE of 8.47, substantially outperforming the best-performing zero-shot method, which recorded an MAE of 42.37. While some prior supervised models (DeBERTa-V3) reported competitive results (MAE of 15.55), our model still demonstrates a significant margin of improvement. Conversely, in the TriBERT dataset, a few zero-shot detectors such as FastDetectGPT (`falcon-7b-instruct`) and Binoculars showed comparatively stronger performance, achieving an average $F1@K$ of 0.608 and outperforming several prior supervised baselines. Nevertheless, our proposed model consistently outperformed all baselines, including both zero-shot and prior supervised models, across varying author-mixture types. Comprehensive results for M4GT and TriBERT comparisons are provided in Tables 4 and 5, respectively.

The evaluation metrics like Accuracy, Precision, Recall, F1-Score, MCC, and Kappa's scores are really high, scoring around 98% and 91% of accuracy in the TriBERT dataset and the M4GT dataset, respectively, which can be seen in the Tables 12 and 13. These results might initially suggest excellent model performance and say that the model is highly effective at distinguishing between human-written and AI-generated collaborative text. However, these traditional metrics primarily reflect token-level or word-level classification, meaning they only assess whether a single word is correctly labeled or not among the binary labels (0 and 1). In automatic boundary detection, these metrics cannot assess to capture the border structural coherence and make the correct segmentation. In contrast, the main evaluation considered by the authors involves identifying the correct segmentation of the human and AI parts, which are better and task-relevant criteria. This segmentation-based evaluation provides a clearer picture of how well the model understands distinct regions of authorship within a collaborative text, offering a more appropriate measure of performance for this work. We conclude that the Transformers model has long-range dependencies and lacks explicit span modeling. By adding Neural Network and CRF layers, we align local contextual cues with structured transitions, enabling more accurate detection of authorship shifts.

Type	Model	MAE
Zero-shot	Glimpse (babbage-002)	78.84
	Glimpse (davinci-002)	72.63
	FastDetectGPT (gpt-neo-2.7b)	75.19
	FastDetectGPT (falcon-7b-instruct)	48.91
	Binoculars	42.37
	Binoculars (Sliding Window)	39.55
Prior Works	Longformer (Wang et al., 2024)	22.12 (21.54)
	USTC-BUPT (Guo et al., 2024)	19.91 (17.70)
	AIpom (Shirnin et al., 2024)	16.42 (15.94)
	TM-TREK (Qu and Meng, 2024)	15.03 (15.68)
	DeBERTa-V3 (Wang et al., 2024)	16.08 (15.55)
HMM & MEMM	*DeBERTa + BiGRU + MEMM*	26.73
	DeBERTa + BiGRU + HMM	22.19
Proposed	**Ours* - (*DeBERTa + BiLSTM + CRF*)**	**13.95**
	Ours* - (*DeBERTa + BiGRU + CRF*)	**8.47**

Table 4: Comparison on the M4GT (Wang et al., 2024) dataset with both the zero-shot methods and the prior models that utilized this dataset. The best results are marked in bold and * denotes the model with all optimizations and best hyperparameters. The one in the brackets '()' are the reported values in the respective papers.

Type	Model	Bry=1	Bry=2	Bry=3	All
Zero-shot	Glimpse (babbage-002)	0.194	0.256	0.303	0.251
	Glimpse (davinci-002)	0.427	0.506	0.564	0.499
	FastDetectGPT (gpt-neo-2.7b)	0.253	0.317	0.362	0.310
	FastDetectGPT (falcon-7b-instruct)	0.482	0.553	0.619	0.551
	Binoculars	0.517	0.624	0.683	0.608
	Binoculars (Sliding Window)	0.534	0.641	0.702	0.625
Prior Works Reported	TriBERT (p=2) (Zeng et al., 2024)	0.455	0.692	0.622	0.575
	GigaCheck (Tolstykh et al., 2024)	0.444	0.693	0.801	0.646
Prior Works Reproduced	TriBERT (p=2) (Zeng et al., 2024)	0.428	0.671	0.594	0.564
	GigaCheck (Tolstykh et al., 2024)	0.472	0.715	0.826	0.671
HMM & MEMM	*DeBERTa + BiGRU + MEMM*	0.562	0.717	0.754	0.677
	DeBERTa + BiGRU + HMM	0.593	0.749	0.787	0.710
Proposed	**Ours* - (*DeBERTa + BiLSTM + CRF*)**	**0.612**	**0.734**	**0.817**	**0.721**
	Ours* - (*DeBERTa + BiGRU + CRF*)	**0.695**	**0.846**	**0.878**	**0.806**

Table 5: Comparison on the TriBERT (Zeng et al., 2024) dataset with the zero-shot methods, the prior model that utilized this dataset and and the HMM & MEMM models, the best ones are marked in bold and * denotes the model with all optimizations and best hyperparameters.

8 Ablation Study

We mentioned that our approach model is too complex due to its hybrid-hierarchical component architecture. To reduce the complexity while maintaining the performance good, we used techniques, 1) Layer-wise Learning Rate Decay (*LLRD*), 2) Dynamic Dropout, and 3) Xavier initialization of weights. To check the usability of these techniques, we went on an ablation by including one-on-one, and finally, all techniques were added. An overview of the significance of these techniques is given below.

1. **Significance of these techniques:**
   (a) Xavier Initialization is used in the fully connected layer to ensure stable weight initialization.
   (b) In LLRD, earlier layers of the transformer (embeddings and encoder layers) have lower learning rates, while the later encoder layers, NN, FC, and CRF layers, have progressively larger learning rates. (LR: 1e-6 → 5e-6 → 1e-5 → 1e-4)
   (c) Dynamic Dropout changes the dropout rate based on the layer depth or training progression, significantly ensuring an optimal balance between regularization and learning capacity.

Other than these techniques, we also mentioned the usage of HMM and MEMM instead of CRF, for that, we built the model with the same configuration as CRF model with the HMM[8] and MEMM[9].
We did this ablation for only the Best Performing model, which is the TNC-DBGC**[10] **model with all the optimization techniques and the best hyperparameters.

According to the above Tables 6 and 7, which present a detailed ablation study on the effects of inclusion and exclusion optimization techniques alongside traditional probabilistic sequence models such as HMMs and MEMMs, it shows that the involvement of these optimization strategies plays a very crucial role in enhancing model performance. These clearly show the necessity for the model's progressive improvement in performance when optimization techniques are sequentially applied to the base TNC-DBGC model.

[8]DBGH - DeBERTa + BiGRU + HMM
[9]DBGM - DeBERTa + BiGRU + MEMM
[10]TNC-DBGC: Transformer + Neural Network + CRF - DeBERTa + BiGRU + CRF

Model	MAE
DBGM*	26.73
DBGH*	22.19
TNC-DBGC	19.85
+ Dynamic Dropout	16.42
+ Xavier Initialization	12.76
+ LLRD	10.03
+ All (TNC-DBGC*)	**8.47**

Table 6: Ablation study on the M4GT dataset with the HMMs and MEMMs and the inclusion and exclusion of the optimization techniques.

Model	Bry=1	Bry=2	Bry=3	All
DBGM*	0.562	0.717	0.754	0.677
DBGH*	0.593	0.749	0.787	0.710
TNC-DBGC	0.625	0.775	0.812	0.737
+ Dynamic Dropout	0.645	0.798	0.838	0.760
+ Xavier Initialization	0.670	0.821	0.867	0.786
+ LLRD	0.681	0.835	0.871	0.795
+ All (TNC-DBGC*)	**0.695**	**0.846**	**0.878**	**0.806**

Table 7: Ablation study on the TriBERT dataset with the HMMs and MEMM and the inclusion and exclusion of the optimization techniques.

In the Table 6, which focuses on the M4GT dataset, we can observe a significant difference in the MAE between the models TNC-DBGC and All (TNC-DBGC*) from 19.85 to 8.47, respectively. This nearly 60% reduction in MAE underscores the effectiveness and necessity of these enhancements in refining the model's segmentation accuracy.

In such a way, Table 7, which presents the TriBERT dataset, follows the same trend as in M4GT dataset, where the TNC-DBGC alone has given a overall result of 0.737, however upon the incremental addition of these optimization techniques have given us the value of 0.806 in the fully optimized TNC-DBGC* configuration.

Our current ablation study focuses on the optimization techniques, which we believe are critical to our model's performance. To further justify our architectural choices, we conducted an additional experiment where we replaced the CRF layer with a simple token-level softmax classifier on top of the DeBERTa+BiGRU encoder. This resulted in a significant performance drop: On M4GT, the MAE increased from 8.47 to 14.21, and on TriBERT, the overall F1@K score dropped from 0.806 to 0.743. This says that CRF layer's ability to model depen-

dencies between adjacent labels is crucial for predicting coherent authorship spans and accurately identifying boundaries. We will add a brief discussion of this finding to the Ablation Study section to further justify our choice of the CRF component.

8.1 Multiple Random seeds for statistical significance

We have re-run our best-performing model ($Ours* -DeBERTa+BiGRU+CRF$) Seeds: 42, 123, 1024, 2025, and 8888.

Seed	Prposed Model
42	8.61
123	8.49
1024	8.28
2025	8.52
8888	8.45
Mean ± Std	**8.47 ± 0.15**

Table 8: Random seed with our proposed model $Ours* -DeBERTa+BiGRU+CRF$ on the M4GT dataset.

Seed	Bry=1	Bry=2	Bry=3	All
42	0.688	0.839	0.869	0.798
123	0.697	0.845	0.876	0.805
1024	0.704	0.854	0.886	0.815
2025	0.693	0.844	0.873	0.803
8888	0.694	0.847	0.884	0.809
Mean ± Std	**0.695 ± 0.006**	**0.846 ± 0.005**	**0.878 ± 0.007**	**0.806 ± 0.009**

Table 9: Random seed with our proposed model $Ours* -DeBERTa + BiGRU + CRF$ on the TriBERT dataset.

9 Conclusion and Future Work

The widespread availability of LLMs has introduced significant challenges across multiple domains, where controlling the inappropriate or unintended use of these models has become increasingly difficult. To overcome these problems, in this paper, we propose an approach to segment the human and AI spans in a collaboratively written text. Our approach uses a hybrid component architecture that combines 1) Transformers, 2) Neural Networks, and 3) CRF. For the justification to prove our model performs better than existing ones, we evaluated our method on two Human-AI collaborative text datasets: 1) TriBERT and 2) M4GT, and achieved state-of-the-art results on all of them, and compared the results with zero-shot approaches that include FastDetectGPT, Glimpse, and Binoculars, and also performed the experimentation with HMMs and MEMMs.

As a future work, we wanted to enhance our model by incorporating style features, such as lexical diversity or syntactic complexity, to improve boundary detection even under adversarial conditions, syntactical or semantical ones. Exploring multi-task learning to jointly predict authorship and boundary points may further improve the performance. Additionally, evaluating the model on diverse datasets with various AI-generated texts and testing its generalizability across languages could broaden its applicability.

10 Limitations

Although our models and methodologies achieved promising results, a key limitation is the lack of robustness against both syntactic and semantic adversarial attacks. The model has not been explicitly trained or tested on adversarial samples, which could potentially manipulate linguistic structures or semantics to mislead sequence labeling. Addressing these challenges in future work could significantly enhance model reliability and generalization in real-world applications.

References

Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. 2012. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3).

Yadagiri Annepaka and Partha Pakray. 2025. Large language models: a survey of their development, capabilities, and applications. *Knowledge and Information Systems*, 67(3):2967–3022.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Daniel Buschek, Martin Zürn, and Malin Eiband. 2021. The impact of multiple parallel phrase suggestions on email input and composition behaviour of native and non-native english writers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

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

Varun Dogra, Sahil Verma, Kavita, Pushpita Chatterjee, Jana Shafi, Jaeyoung Choi, and Muhammad Fazal Ijaz. 2022. A complete process of text classification system using state-of-the-art nlp models. *Computational Intelligence and Neuroscience*, 2022(1):1883698.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. Roft: A tool for evaluating human detection of machine-generated text. *arXiv preprint arXiv:2010.03070*.

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.

Zikang Guo, Kaijie Jiao, Xingyu Yao, Yuning Wan, Haoran Li, Benfeng Xu, Licheng Zhang, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2024. Ustcbupt at semeval-2024 task 8: Enhancing machine-generated text detection via domain adversarial neural networks and llm embeddings. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1511–1522.

A Hans, A Schwarzschild, V Cherepanova, H Kazemi, A Saha, M Goldblum, J Geiping, and T Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. *URL: https://arxiv.org/abs/2401.12070*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Laida Kushnareva, Tatiana Gaintseva, German Magai, Serguei Barannikov, Dmitry Abulkhanov, Kristian Kuznetsov, Eduard Tulchinskii, Irina Piontkovskaya, and Sergey Nikolenko. 2023. Ai-generated text boundary detection with roft. *arXiv preprint arXiv:2311.08349*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–19.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

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.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77.

Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. 2023. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242.

Xiaoyan Qu and Xiangfeng Meng. 2024. Tm-trek at semeval-2024 task 8: Towards llm-based automatic boundary detection for human-machine mixed text. *arXiv preprint arXiv:2404.00899*.

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

Alexander Shirnin, Nikita Andreev, Vladislav Mikhailov, and Ekaterina Artemova. 2024. Aipom at semeval-2024 task 8: Detecting ai-produced outputs in m4. *arXiv preprint arXiv:2403.19354*.

Felix Stahlberg. 2020. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418.

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, Aleksandr Gordeev, Vladimir Dokholyan, and Maksim Kuprashevich. 2024. Gigacheck: Detecting llm-generated content. *arXiv preprint arXiv:2410.23728*.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. Seqxgpt: Sentence-level ai-generated text detection. *arXiv preprint arXiv:2310.08903*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *arXiv preprint arXiv:2402.11175*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer:

A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guangliang Chen. 2024. Towards automatic boundary detection for human-ai collaborative hybrid essay in education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22502–22510.

A Details on Datasets

Table 10 tells about the TriBERT (Zeng et al., 2024) Hybrid Mixed Text Dataset statistics, which details the distribution of academic essays with 1, 2, or 3 human-AI authorship boundaries. It includes metrics such as essay counts (*17,136* total), average words (*287.6*) and sentences (*13.7*) per essay, average lengths of AI-generated (*22.2* words) and human-written (*22.4* words) sentences, and at last the proportion of AI-generated sentences (65.3%). Table 11 gives an overview about the M4GT Boundary Identification Dataset (Wang et al., 2024), which includes the PeerRead and OUTFOX subsets. PeerRead includes *5,676* samples each for ChatGPT and LLaMA-2 models (*7B, 13B, 70B*), split into train (*3,649*), dev (*505*), and test (*1,522*) sets, with an additional *5,189* samples for LLaMA-2-7B*. OUTFOX provides *1,000* test samples each for GPT-4 and LLaMA-2 variants.

	Boundaries			All
	1	2	3	
Hybrid essay	7488	6429	3219	17136
Words per essay	275.3	279.5	332.6	287.6
Sentences per essay	12.9	13.4	16.1	13.7
Avg len of AI-gen sent	22.7	21.8	21.7	22.2
Avg len of human-written sent	22.7	22.6	21.2	22.4
Ratio of AI-gen sent per essay	67.4%	58.8%	73.2%	65.3%

Table 10: TriBERT Hybrid Mixed text Dataset statistics by (Zeng et al., 2024)

Domain	Generator	Train	Dev	Test	Total
PeerRead	ChatGPT	3,649	505	1,522	5,676
	LLaMA-2-7B*	3,649	505	1,035	5,189
	LLaMA-2-7B	3,649	505	1,522	5,676
	LLaMA-2-13B	3,649	505	1,522	5,676
	LLaMA-2-70B	3,649	505	1,522	5,676
OUTFOX	GPT-4	–	–	1,000	1,000
	LLaMA-2-7B	–	–	1,000	1,000
	LLaMA-2-13B	–	–	1,000	1,000
	LLaMA-2-70B	–	–	1,000	1,000

Table 11: M4GT Boundary identification data based on GPT and LLaMA-2 series (Wang et al., 2024).

B Other Experimental Results

As mentioned above, other than the main evaluation metrics, we have also computed the Accuracy, Precision, Recall, F1-Score, MCC, and Cohen's Kappa score for both datasets. All best model in each setting are highlighted in bold. According to the metrics, in *NN_CRF* setting, the BiGRU_CRF model has the highest scores among the other models. In *Transformer_CRF* setting, DeBERTa_CRF model got the highest accuracy with 97.82%. Finally, among the *Transformer_NN_CRF* setting, the model DeBERTa_BiGRU_CRF has the highest scores in all the metrics and outperforms all the models, including the zero-shot and the models that were previously tested on. The results of these metrics can be seen in the Tables 12 and 13.

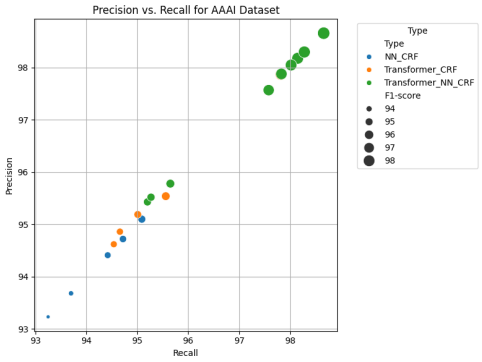
According to figures 2, 5, 4, it is certain that the increase in the combination of number of blocks in a model leads to have high values in the evaluation metrics and were able to outperform the other models with their previous setting. In conclusion, the results from the TriBERT Dataset are quite high, and the results on the M4GT dataset are comparatively low. This shows that the robustness of the data is higher in the M4GT dataset, where a model cannot easily predict the boundary in the M4GT dataset and can predict easily in the TriBERT dataset. The results comparison plot for both datasets can be seen in Figure 3.

Dataset	Type	Model	Accuracy	Precision	Recall	F1-score	MCC	Kappa
AAAI	NN_CRF	CNN_CRF	94.72	94.72	94.72	94.69	88.28	88.22
		RNN_CRF	93.7	93.68	93.7	93.66	85.98	85.92
		LSTM_CRF	93.25	93.23	93.25	93.2	84.97	84.9
		BiLSTM_CRF	94.42	94.41	94.42	94.39	87.61	87.55
		BiGRU_CRF	**95.09**	**95.1**	**95.09**	**95.06**	**89.1**	**89.04**
	Transformer_CRF	BERT_CRF	94.66	94.86	94.66	94.57	88.15	87.76
		DistilBERT_CRF	94.54	94.62	94.54	94.47	87.8	87.57
		RoBERTa_CRF	95.08	95.19	95.01	94.92	88.65	88.37
		ModernBERT_CRF	95.56	95.54	95.56	95.54	89.92	89.9
		DeBERTa_CRF	**97.82**	**97.87**	**97.82**	**97.8**	**95.18**	**95.09**
	Transformer_NN_CRF	DeBERTa_CNN_CRF	98.15	98.18	98.15	98.14	95.91	95.86
		DeBERTa_RNN_CRF	97.83	97.88	97.83	97.82	95.22	95.13
		DeBERTa_LSTM_CRF	98.02	98.05	98.02	98	95.62	95.55
		DeBERTa_BiLSTM_CRF	98.28	98.3	98.28	98.27	96.2	96.16
		DeBERTa_BiGRU_CRF	**98.66**	**98.66**	**98.66**	**98.66**	**97.02**	**97.02**
		BERT_BiGRU_CRF	95.65	95.78	95.65	95.6	90.35	90.09
		DistilBERT_BiGRU_CRF	95.2	95.43	95.2	95.12	89.49	88.99
		RoBERTa_BiGRU_CRF	95.27	95.52	95.27	95.18	89.41	88.95
		ModernBERT_BiGRU_CRF	97.58	97.57	97.58	97.57	94.51	94.49

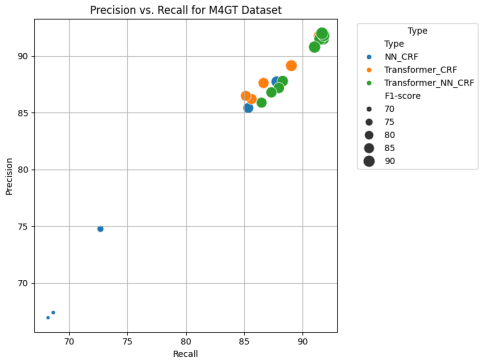
Table 12: Conventional metrics results table that included the metrics Accuracy, Precision, Recall, F1-Score, MCC and Kappa score on the AAAI-TriBERT dataset by Zeng et al. (2024).

Dataset	Type	Model	Accuracy	Precision	Recall	F1-score	MCC	Kappa
M4GT	NN_CRF	CNN_CRF	74.72	74.77	72.68	72.68	41.98	39.18
		RNN_CRF	68.19	66.95	68.19	67.11	27.45	26.98
		LSTM_CRF	68.64	67.4	68.64	67.62	28.39	27.87
		BiLSTM_CRF	87.35	85.42	85.35	84.96	67.43	66.67
		BiGRU_CRF	**87.79**	**87.73**	**87.79**	**87.75**	**73.24**	**73.22**
	Transformer_CRF	BERT_CRF	86.2	86.2	85.63	85.63	69.67	68.04
		DistilBERT_CRF	89.04	89.14	89.04	88.83	75.82	75.32
		RoBERTa_CRF	86.66	87.61	86.66	86.05	70.94	68.96
		ModernBERT_CRF	85.76	86.48	85.14	85.14	68.71	66.94
		DeBERTa_CRF	**91.41**	**91.75**	**91.46**	**91.22**	**81.28**	**80.57**
	Transformer_NN_CRF	DeBERTa_CNN_CRF	91.13	91.5	91.77	90.92	80.65	79.92
		DeBERTa_RNN_CRF	91.5	91.5	91.5	91.32	81.43	80.82
		DeBERTa_LSTM_CRF	90.78	90.78	91.03	90.57	79.8	79.15
		DeBERTa_BiLSTM_CRF	91.64	91.81	91.81	91.49	81.68	81.22
		DeBERTa_BiGRU_CRF	**91.82**	**91.99**	**91.67**	**91.67**	**82.08**	**81.62**
		BERT_BiGRU_CRF	85.88	85.88	86.49	85.3	68.9	67.31
		DistilBERT_BiGRU_CRF	87.78	87.78	88.3	87.36	73.21	71.94
		RoBERTa_BiGRU_CRF	87.18	87.18	87.97	86.65	72.02	70.31
		ModernBERT_BiGRU_CRF	86.79	86.79	87.33	86.3	70.96	70.57

Table 13: Conventional metrics results table that included the metrics Accuracy, Precision, Recall, F1-Score, MCC, and Kappa score on the M4GT dataset by Wang et al. (2024).



(a) TriBERT Precision vs Recall



(b) M4GT Precision vs Recall

Figure 2: Precision vs Recall plots for TriBERT and M4GT datasets.

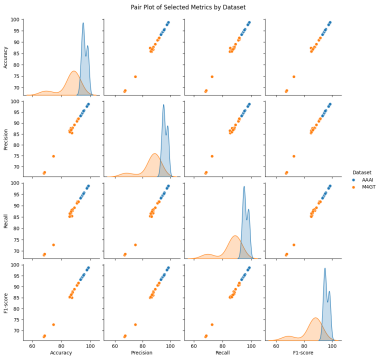
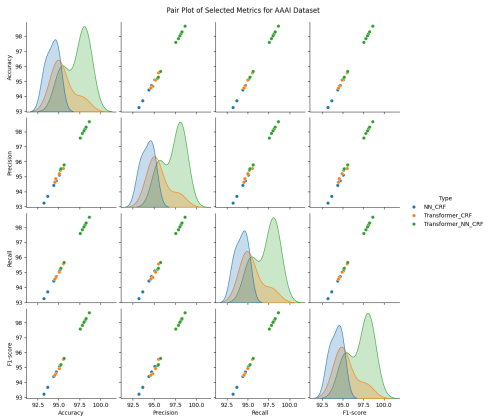
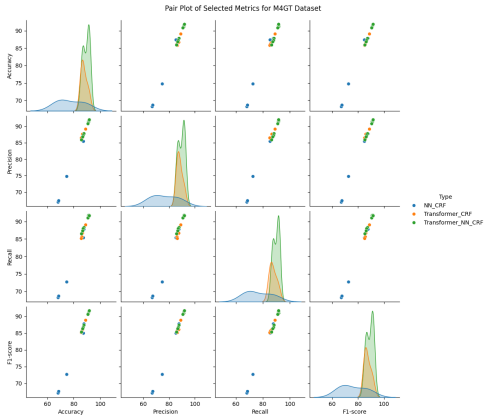


Figure 3: Combined Result values over the two Datasets.

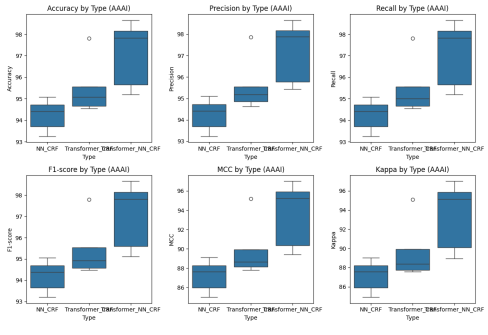


(a) TriBERT Pair Plot

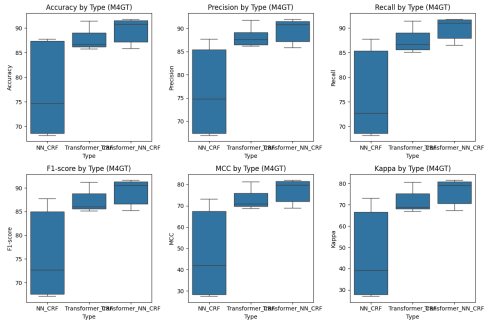


(b) M4GT Pair Plot

Figure 4: Pair plots for TriBERT and M4GT datasets.



(a) TriBERT Box Plot



(b) M4GT Box Plot

Figure 5: Box plots for TriBERT and M4GT datasets.

C Loss Plots

Our model includes the CRF layers such that the loss during the training of the model is the CRF loss, where CRF tries to maximize the log-likelihood of the correct label sequence. Such that the loss at the first epochs would be very high, and upon predicting the token-label correctly, the loss would eventually decrease. The numerical values during the training of the models on two datasets can be seen in the Tables 14, 15. A better visualization of the decrease over the training epochs can be seen in the Figures 6, 7, and 8.

The CRF loss is mathematically expressed as:

$$\mathcal{L}_{CRF} = -\log P(y \mid x) \tag{8}$$

where: $P(y \mid x)$ is the conditional probability and is given in the equation 1. This is further expanded as follows when Equation 1 is substituted above:

$$\mathcal{L}_{CRF} = -\left(S(x, y) - \log Z(x)\right) \tag{9}$$

Where: $S(x, y)$ is the sum of the transition scores and emission scores of the CRF model, and the equation is 3. $\log Z(x)$ is the log of the partition function $Z(x)$ and the Equation is 2. The loss minimization reflects the CRF's ability to capture dependencies between labels, improving boundary detection accuracy over epochs.

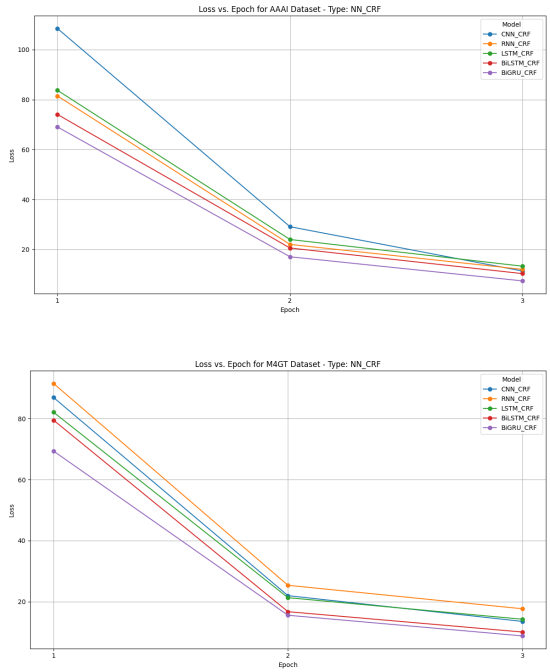


Figure 6: Loss curves for neural network with CRF models on TriBERT and M4GT datasets.

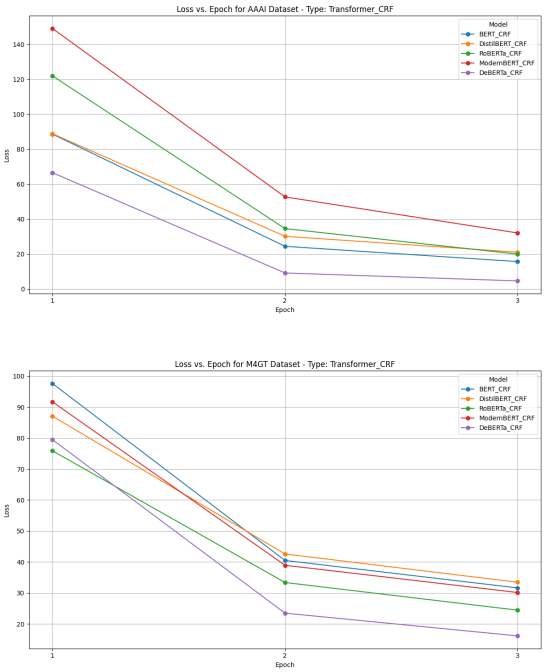


Figure 7: Loss curves for transformer with CRF models on TriBERT and M4GT datasets.

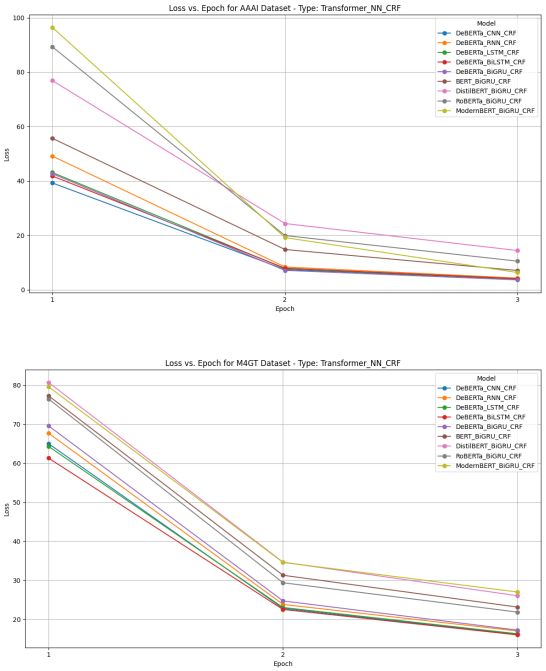


Figure 8: Loss curves for transformer and neural network with CRF models on TriBERT and M4GT datasets.

Dataset	Type	Model	Epoch 1	Epoch 2	Epoch 3
AAAI	NN_CRF	CNN_CRF	108.47	29.12	11.37
		RNN_CRF	81.48	22.04	12.04
		LSTM_CRF	83.72	24.01	13.31
		BiLSTM_CRF	74.07	20.59	10.34
		BiGRU_CRF	69.06	17.09	7.43
	Transformer_CRF	BERT_CRF	88.66	24.38	15.64
		DistilBERT_CRF	88.87	30.14	21
		RoBERTa_CRF	121.99	34.52	19.91
		ModernBERT_CRF	149.1	52.66	32.03
		DeBERTa_CRF	66.56	9.12	4.58
	Transformer_NN_CRF	DeBERTa_CNN_CRF	39.26	7.44	3.96
		DeBERTa_RNN_CRF	49.14	8.4	4.31
		DeBERTa_LSTM_CRF	43.11	7.8	4
		DeBERTa_BiLSTM_CRF	41.81	7.95	4.08
		DeBERTa_BiGRU_CRF	42.66	7.14	3.64
		BERT_BiGRU_CRF	55.67	14.82	7.1
		DistilBERT_BiGRU_CRF	76.88	24.36	14.42
		RoBERTa_BiGRU_CRF	89.3	19.94	10.53
		ModernBERT_BiGRU_CRF	96.41	19.19	6.34

Table 14: Model Loss Across Epochs during training of the model on the TriBERT dataset.

Dataset	Type	Model	Epoch 1	Epoch 2	Epoch 3
M4GT	NN_CRF	CNN_CRF	86.87	21.98	13.54
		RNN_CRF	91.48	25.37	17.69
		LSTM_CRF	82.05	21.32	14.26
		BiLSTM_CRF	79.45	16.73	10.07
		BiGRU_CRF	69.33	15.56	8.8
	Transformer_CRF	BERT_CRF	97.62	40.48	31.62
		DistilBERT_CRF	87.06	42.58	33.45
		RoBERTa_CRF	75.9	33.37	24.44
		ModernBERT_CRF	91.7	38.93	30.14
		DeBERTa_CRF	79.49	23.48	16.13
	Transformer_NN_CRF	DeBERTa_CNN_CRF	65	22.81	16.13
		DeBERTa_RNN_CRF	67.69	23.83	17.02
		DeBERTa_LSTM_CRF	64.3	23.04	16.31
		DeBERTa_BiLSTM_CRF	61.3	22.54	16.02
		DeBERTa_BiGRU_CRF	69.57	24.73	17.25
		BERT_BiGRU_CRF	77.28	31.3	23.15
		DistilBERT_BiGRU_CRF	80.68	34.65	26.05
		RoBERTa_BiGRU_CRF	76.46	29.38	21.87
		ModernBERT_BiGRU_CRF	79.57	34.59	27.01

Table 15: Model Loss Across Epochs during training of the model on the M4GT dataset