*future internet*

MDPI

*Article*

# Beyond Lexical Boundaries: LLM-Generated Text Detection for Romanian Digital Libraries

**Melania Nitu [1] and Mihai Dascalu [1,2,*]**

[1] Faculty of Automatic Control and Computers, National University of Science and Technology Politehnica Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania; suzana_melania.nitu@upb.ro
[2] Academy of Romanian Scientists, Str. Ilfov, Nr.3, 050044 Bucharest, Romania
* Correspondence: mihai.dascalu@upb.ro

**Abstract:** Machine-generated content reshapes the landscape of digital information; hence, ensuring the authenticity of texts within digital libraries has become a paramount concern. This work introduces a corpus of approximately 60 k Romanian documents, including human-written samples as well as generated texts using six distinct Large Language Models (LLMs) and three different generation methods. Our robust experimental dataset covers five domains, namely books, news, legal, medical, and scientific publications. The exploratory text analysis revealed differences between human-authored and artificially generated texts, exposing the intricacies of lexical diversity and textual complexity. Since Romanian is a less-resourced language requiring dedicated detectors on which out-of-the-box solutions do not work, this paper introduces two techniques for discerning machine-generated texts. The first method leverages a Transformer-based model to categorize texts as human or machine-generated, while the second method extracts and examines linguistic features, such as identifying the top textual complexity indices via Kruskal–Wallis mean rank and computes burstiness, which are further fed into a machine-learning model leveraging an extreme gradient-boosting decision tree. The methods show competitive performance, with the first technique's results outperforming the second one in two out of five domains, reaching an F1 score of 0.96. Our study also includes a text similarity analysis between human-authored and artificially generated texts, coupled with a SHAP analysis to understand which linguistic features contribute more to the classifier's decision.

**Keywords:** machine-generated text detection; large language models; natural language generation; text analysis



**Citation:** Nitu, M.; Dascalu, M. Beyond Lexical Boundaries: LLM-Generated Text Detection for Romanian Digital Libraries. *Future Internet* **2024**, *16*, 41. https://doi.org/10.3390/fi16020041

Academic Editors: Francesca Fallucchi and Ernesto William De Luca

Received: 10 January 2024
Revised: 22 January 2024
Accepted: 23 January 2024
Published: 25 January 2024



## 1. Introduction

The outstanding capability of the Large Language Models (LLMs) to generate human-like texts has raised content authenticity concerns for researchers and users around the world [1,2]. LLMs demonstrated great potential in NLP tasks such as machine translation [3], summarization [4], dialogue systems [5], question answering [6–8], or information retrieval [9,10]. In these scenarios, the objective is to produce qualitative texts that meet the user's requirement rather than deceive or mislead. Among their various applications, the LLMs can be misused, for instance, to produce academic essays and research papers or even generate fake news. They can produce nearly identical content with human-authored texts when the LLM distribution is human-like, thus making the detection harder and requiring the collection of additional samples [11,12].

A legitimate question is raised by Clark et al. [12] who stated that "Human evaluations are considered the gold standard in Natural Language Generation (NLG), but as models' fluency improves, how well can evaluators detect and judge machine-generated text (MGT)?". Hence, there is an emergent need for performant automated detection systems and tailored approaches to maximize their potential.

Additionally, despite the frequent attention given to high-resource languages such as English, it is crucial not to overlook the importance of Natural Language Processing (NLP) tools for languages with limited resources. Consequently, this research centers on using NLP for Romanian, a language facing distinct challenges due to insufficient resources. Nevertheless, the knowledge derived from this investigation offers flexible methodologies and approaches that overcome the constraints imposed by limited data and can serve as a starting point for studies on other low or less-resourced languages.

*Current Study Objective*

This paper examines existing techniques for identifying machine-generated text, explores their limitations, and proposes novel approaches that leverage advanced linguistic analysis, contextual understanding, and pattern recognition to enhance the accuracy and reliability of detection systems. By addressing this pressing issue, the research aims to contribute to the development of more resilient and effective tools for safeguarding against the proliferation of machine-generated text with malicious intent in digital contexts.

In this work, we propose and evaluate two detection methods to distinguish the automated generated texts in the context of digital libraries, leveraging two models' architectures, namely Transformer-based and classic ML-based. As part of this study, starting from a human-authored set of texts (7 k), we developed an extensive corpus of artificially generated texts of around 52 k documents using six different LLMs and three different generation methods, comprising texts across five domains. Among the text generation techniques, we mention text completion, backtranslation, or paraphrasing. Text completion is an NLP technique used to generate text by predicting the next word or sequence of words in a given context. This method uses statistical language models that have been trained on a large amount of textual data to generate the most probable words that would complete a sentence or a paragraph. The language models used for text completion in this experiment are based on GPT architecture [13] and are pre-trained on Romanian corpus. Backtranslation is a text generation method that involves iterative translations from one language to another and then translating it back to the original language. This method is used to generate variations of the original text that may have different sentence structures, words, or meanings. Paraphrasing algorithms generate a semantically similar text and a grammatically correct output, preserving the input's original meaning. A new text is generated with the same sense while using different words and sentence structures. Paraphrasing can also be used as a simplification instrument for rephrasing complex texts, facilitating text comprehension by decreasing the reading difficulty.

The main contributions of our work are as follows:

- Contribution to the development of resources for a less-resourced language, such as Romanian;
- The development of an extensive Romanian dataset, containing both human-authored and machine-generated texts, of around 60 k documents across five domains, generated via seven distinct methods, using several LLMs and generation techniques like text completion, backtranslation, and paraphrasing;
- Two detection models based on different architectures, namely Transformer-based and classic ML-based, designed to distinguish automatically generated texts for the Romanian language. The ML model slightly outperformed the Transformer model, reaching a macro F1 of 0.91, while the Transformer exhibited a macro F1 of 0.90;
- An analysis of textual similarities between human-authored and machine-generated texts for Romanian via similarity measures (cosine, BLEU, ROUGE) and non-parametric statistical tests;
- An exploration of linguistic differences between human and artificially generated texts via textual complexity indices.

We release as open-source the dataset (https://huggingface.co/datasets/readerbench/ro-human-machine-60k, accessed on 8 January 2024), and the codebase (https://github.com/readerbench/ro-mgt-detection, accessed on 8 January 2024).

## 2. Related Work

Artificial text generation has been an active area of NLP research in recent years. This section investigates the challenges associated with machine-generated text and the imperative need for robust detection mechanisms. The integration of text generation through generative language models has transformed NLP [14], facilitating content enrichment and opening the debate for ethical considerations as also highlighted by Bender [15], Brown [16], Michel-Villarreal [17], Farrelly [18], and other researchers in their studies. Ethical considerations play a critical role in the responsible development and deployment of these language models, prompting discussions on issues such as bias, misinformation, and the societal impact of AI-generated content. As advancements in NLP have led to the proliferation of sophisticated language models, the potential misuse of these models for generating deceptive or malicious content has become a growing concern. Detecting machine-generated text poses a unique set of challenges, including the ability of these models to mimic human language intricacies and bypass traditional detection methods. Additional challenges emerge when addressing less-resourced languages such as Romanian, necessitating dedicated detection systems due to language specificity.

The subsequent sections provide a brief overview of the main neural architectures employed in this study. Following this, we provide a comprehensive examination of dedicated language models designed for the Romanian language, along with an exploration of popular multilingual models that leverage the foundational architectures of these models. This contextualization lays the foundation for an exploration of the existing landscape in machine-generated text detection relevant to the scope of this research.

### 2.1. Neural LLM Architectures

GenerativePre-Trained Transformer, widely known as GPT, is a class of NLP models (https://platform.openai.com/docs/models/gpt-4, accessed on 8 January 2024) designed for language understanding and generation tasks. GPT is built on the Transformer architecture [3], incorporating self-attention mechanisms to model long-range textual dependencies efficiently. GPT models are pre-trained on a massive corpus of text using unsupervised learning, thus allowing the models to learn a rich language representation from the data without any task-specific labels. Moreover, during pre-training, GPT models leverage the Masked Language Model (MLM) objective to predict missing words in sentences, allowing the model to capture contextual information and relationships between words in a text. Additionally, GPT models are fine-tuned on downstream tasks using supervised learning. Fine-tuning leverages the knowledge acquired during pre-training, which allows GPT to achieve remarkable performance via transfer learning, even with limited data. The GPT architecture has evolved from GPT1 to GPT4, each released version being characterized by an increase in model size. GPT1 (117M parameters, max sequence length of 1024) [13] was originally trained on a combination of two datasets: Common Crawl and Book Corpus. This first version was, however, prone to generate repetitive text and could not track long-term dependencies in a text, producing coherent results only for short text sequences. GPT2 (1.5B parameters, max sequence length of 2048) [2] was trained on an exponentially larger corpus combining Common Crawl, Book Corpus, and Web Text, being capable of generating more human-like answers. Similar to its predecessor, it performed well on shorter texts, lacking coherence or reasoning on longer texts. GPT3 (175 B parameters, max sequence length of 4096) [16] kept increasing the training corpus, incorporating Wikipedia, books, articles, and other sources, building datasets of trillion words. GPT3 capabilities included generating coherent texts for longer sequences, understanding context, writing computer code, or even creating art. GPT4 [19] was pre-trained to predict the next token in a text and fine-tuned using Reinforcement Learning from Human Feedback [20]. Specifics on model training and size have not yet been publicly released. GPT4 provides significant improvements compared to its previous versions, being capable of processing images and audio and providing coherent answers. The GPT series introduced variations in its

architecture, such as layer normalization, gradient accumulation, and positional encodings, to enhance model training and performance.

Text-To-Text Transfer Transformer (T5) architecture [21] is a transformative approach in NLP, where language tasks are formulated as unified text-to-text problems leveraging transfer learning. T5 is also built on the Transformer architecture, relying on self-attention mechanisms and being capable of capturing contextual information across input and output tokens. T5 leverages an encoder–decoder architecture, where the encoder processes the input text, and the decoder generates the output text. Moreover, T5 uses positional encodings to preserve spatial information and token positions. The authors used SentencePiece [22] to encode text as WordPiece [23] tokens, resulting in a vocabulary of 32,000-word pieces. The model was pre-trained on a set of languages, including Romanian. There are several model versions released of different sizes—e.g., T5-Small (60 M parameters), T5-Base (220 M parameters), T5-Large (770 M parameters), T5-3B (3 B parameters), and T5-11B (11 B parameters). The models achieved state-of-the-art results on various tasks such as machine translation, abstractive summarization, question answering, and text classification.

Finetuned Language Net (FLAN) [24] emphasized fine-tuning as a key component in its design for achieving high performance across various tasks and improving zero-shot learning ability for language models. FLAN was fine-tuned on a large set of instructions—more than 470 NLP datasets and 1800 tasks, which makes the model suitable to follow instructions, even for unseen tasks. The model was evaluated against language inference, reading comprehension, question answering, machine translation, common sense reasoning, coreference resolution, and additional tasks like sentiment analysis, paraphrase detection, and struct-to-text. FLAN-T5 outperformed GPT3 on zero-shot prompting on 20 out of 25 tasks.

Bidirectional Auto-Regressive Transformers (BART) [25] is a denoising auto-encoder, representing a combination between BERT and GPT architectures, by using a seq2seq machine translation with bidirectional encoder and left to right decoder. The pre-training shuffled the order of sentences, and chunks of text were replaced with masked tokens using an infilling scheme. BART is efficient in text generation and comprehension tasks, achieving good results on various NLP tasks like machine translation, abstractive dialogue, question answering, or summarization. The model was released in two standard versions: BART-base (6 encoder and decoder layers and 140 M parameters) and BART-large (12 encoder and decoder layers and 400 M parameters), and three fine-tuned versions of the BART-large model on MNLI [26] which is a bitext classification that predicts if one sentence entails another (BART-large-mnli), CNN/DM [27] which is a news summarization dataset (BART-large-cnn), and Xsum [28] which is also a news summarization dataset with highly abstractive summaries (BART-large-xsum).

### *2.2. Generative Models*

#### 2.2.1. Romanian Specific Language Models

Most of the previous models were released for high-resourced languages, such as English or multilingual models, with lower performance than the language-dedicated models—these multilingual versions are presented in detail in the following sub-section. With the development of language models and the increasing need for dedicated resources, Romanian-dedicated language models emerged.

With the introduction of RoGPT2 [29], state-of-the-art performance was achieved for Romanian text generation by leveraging a Romanian version of GPT2 [2]. RoGPT2 was trained using the largest available corpus for Romanian and was evaluated against six tasks from the LiRo benchmark [30], specifically (1) text categorization and dialect classification, (2) sentiment analysis, (3) semantic textual similarity, (4) machine translation, (5) QA and zero-shot cross-lingual learning, and (6) language modeling. For all the targeted tasks except zero-shot cross-lingual learning, RoGPT2 outperformed other BERT-based models for Romanian, such as RoBERT or BERT-ro-base. Competitive results were achieved for another task, namely grammar error correction (RoGEC), leveraging the RONACC corpus,

and demonstrating its capability of generating grammatically correct text. RoGPT2 was released in three versions: base (124 M parameters), medium (354 M parameters), and large (774 M parameters).

A similar model was developed for automatic text generation called MCBGPT-2 [31]. The model was trained exclusively on a corpus of 24,600 news articles collected between March and October 2021, manually labeled as true or fake news with different polarities (i.e., positive or negative). Following the same methodology as RoGPT2, the model was trained using 131 M parameters. Validation was performed via statistical analysis between the original and generated news items. Evaluation metrics such as tokens distribution, BERTScore [32], BLEU [33], and ROUGE [34] metrics were considered. Results show that RoGPT2 [29] exceeded the performance of the MCBGPT-2 model in terms of length and metrics scores of the generated sentences. However, MCBGPT-2 achieved better results for BERTScore, a metric that uses a pre-trained model to understand the generated text and the text of reference for comparison.

RoSummary [35] is a generative language model for abstractive summarization. The model is based on the RoGPT2 architecture, trained to predict the next token using the previous sequence. Four control tokens were used to indicate the characteristics of generated text, namely NoSentences (i.e., number of sentences that the summary should have), NoWords (i.e., number of words generated in the summary), RatioTokens (i.e., proportion in which the sequence of words in the summary must be longer than the input), and LexOverlap (i.e., ratio of 4 grams in the summary that appear in the reference text). The model generated grammatically correct texts and was evaluated using ROUGE and BERTScore. Experiments were conducted with three versions of the model with a context size of 724: base (12 layers, batch size of 128), medium (24 layers, batch size of 24), and large (36 layers, batch size of 16). The medium version achieved the best results, combined with beam search decoding (ROUGE 34.67% and BERTScore 74.34%). However, using control tokens improved BERTScore by up to 2%. Higher scores were achieved when only one control token was used.

A more recent model based on GPT-3 architecture [16] is GPT-NEO-RO (https://huggingface.co/dumitrescustefan/gpt-neo-romanian-780m, accessed on 8 January 2024). The model has 780 M parameters, making it one of the largest language models available for Romanian. The model's architecture is composed of multiple Transformer blocks interconnected through multi-head self-attention mechanisms, allowing the model to capture long-range dependencies and dependencies between different parts of the input sequence, which is critical for generating coherent and fluent text. GPT-NEO-RO was pre-trained on a 40 GB corpus of Romanian text collected from various sources (e.g., Oscar, Wikipedia, Romanian literature), being fine-tuned on various NLP tasks, such as language modeling, text classification, and sentiment analysis.

Romanian versions of T5 [21] and Flan-T5 [36] were recently published by the research community and fine-tuned for specific tasks, such as paraphrasing. Flan-T5-paraphrase-ro (https://huggingface.co/BlackKakapo, accessed on 8 January 2024) is built on the T5 architecture and was fine-tuned for the paraphrasing task. The model generates different versions of the same sentence while preserving its meaning. It was pre-trained on 60,000 Romanian paraphrasing documents (https://huggingface.co/datasets/BlackKakapo/paraphrase-ro, accessed on 8 January 2024), and it was released in three sizes: small (77 M parameters), base (220 M parameters), and large (783 M parameters).

#### 2.2.2. Multilingual Generative Language Models

Although dedicated Romanian language models achieved superior performance over the multilingual models, few multilingual alternatives with support for the Romanian language exhibit good results. The OpenAI text-davinci-003 is a multilingual InstructGPT model [37] part of the GPT-3.5 series, with a size of 1.5 B parameters with 12 Transformer layers. The model was trained on data up to June 2021 using a supervised fine-tuning technique by distilling the best completions from all models and has a max token capacity

of 4097. Newer models based on GPT 3.5 are the turbo models, i.e., gpt-3.5-turbo-1106, trained on data up to September 2021 and having a larger context window (16,385 tokens) compared to its precursors. This version provides improved capabilities of instruction following, embedding JSON mode, and parallel function calling, returning a maximum of 4096 output tokens. The next GPT generation, GPT-4 and GPT-4 turbo, have superior context windows, up to 128,000 tokens, and have been trained on data up to April 2023. Compared to the previous generation, GPT-4 accepts images as input and has advanced reasoning capabilities.

mBart [38] introduced pre-training on Romanian–English corpus (WMT16 [39]) and argued that multilingual denoising improved MT at both sentence level and word level.

Other multilingual language models that leverage the text-to-text transfer Transformer (T5) architecture are mT5 [40] and Flan-T5 [36]. mT5 is a massively multilingual text-to-text Transformer model pre-trained on 101 languages, containing between 300 M and 13 B parameters. The model achieved strong performance on a diverse set of benchmarks. Flan-T5 is an enhanced, fine-tuned version of T5, showing promising results in improving performance and generalization for unseen tasks. Flan-T5 was released in five versions: small (80 M parameters), base (250 M parameters), large (780 M parameters), XL (3 B parameters), and XXL (11 B parameters).

For translation tasks, open-source multilingual projects like Opus-MT [41] emerged. The repository provides over 1000 pre-trained translation models, including for the Romanian language. The authors applied Marian-NMT [42] which is a machine translation framework with efficient training and decoding capabilities.

### *2.3. Detection Mechanisms*

Due to the outstanding performance of the generative language models in producing qualitative texts extremely similar to human writings, machine-generated detection algorithms became a necessity in the research field. Despite the release of various detection systems, none of them have proven to be foolproof. This subsection explores the recent state-of-the-art detection methods and algorithms.

According to Chakraborty [11], recent research in MGT detection can be split into different categories, namely, statistical approaches, classification-based detection models, zero-shot detection, LLM fine-tuning-based detection methods, and watermark-based identification.

Statistical approaches use metrics like entropy, perplexity, or n-gram frequency to distinguish machine-generated and human content [43,44]. More recent studies proposed DetectGPT [45], which states the artificially generated text tends to lie in the negative curvature of the log-likelihood. The model uses Gradient Boosting [46], a Machine Learning (ML) technique that trains multiple models sequentially and combines them to produce a more accurate model, which outperforms other zero-shot methods with high AUC [47] scores. The algorithm extracts a set of 35 features from the input text, such as sentence length, punctuation usage, and the frequency of particular words and phrases. These features are fed to the Gradient Boosting model that predicts if the text is GPT-generated. The model achieved an F1-score of 98.6%. Even though the DetectGPT algorithm is detailed in the Mitchell [45] paper, the implementation was not open sourced nor publicly available until recently. For this reason, the research community created a public implementation (https://github.com/BurhanUlTayyab/DetectGPT, accessed on 8 January 2024) of the algorithm. The approach preserved the idea from the original paper; however, there are a few differences in the feature extraction and the model training process. In the public implementation, 20 features are extracted from the input text, which includes the frequency of particular words and phrases, sentence length, punctuation usage, and the presence of certain characters. These features are then fed to the Gradient Boosting model, which is trained using cross-validation to ensure that the model is generalizable to new data. The model achieved an F1-score of 96.3%, slightly lower than the original paper implementation.

Another statistical approach for MGT detection is GPTZero [48] that uses perplexity [49] and burstiness [50] to classify texts. Perplexity is a measure of text randomness vastly used in NLP. The human-written text is considered less structured and more unpredictable; therefore, its perplexity value should be higher. In contrast, a text generated by AI should have a lower perplexity score. Burstiness considers other variables not accounted for in perplexity to improve text analysis. The term refers to the appearance of non-common tokens in random clusters. The artificially generated text tends to have a more consistent structure than the human-written text. GPTZero uses these two measures to determine if a text is human or AI-generated: through perplexity, GPTZero evaluates how good a language model is at predicting the next token, while with burstiness, it assesses the distribution of sentences. Detection is performed based on the idea that humans tend to mix long and short sentences, while AI-produced sentences are more uniform. Recent research [51] presents encouraging findings in the identification of AI-generated texts using GPTZero. The study reveals an accuracy of 0.80 with a 95% confidence interval, a specificity of 0.90, and a sensitivity of 0.65. The conclusion drawn is that GPTZero exhibits a low false-positive rate (misclassifying human-written texts as machine-generated) and a high false-negative rate (misclassifying machine-generated texts as human-written).

Despite the performance exhibited by DetectGPT and GPTZeo on English corpora, both models had extremely poor results on our Romanian dataset, which is described in the following section. As per the results of our preliminary experiments, GPTZero misdetected all human texts as AI-generated, while DetectGPT misclassified most AI texts as being written by humans. As outlined in the beginning, pre-existing solutions are primarily designed for high-resourced languages like English and exhibit limited performance or complete inoperability when applied to Romanian. As such, there is a need for specialized solutions to deal with the specificity of the Romanian language.

Zero-shot detection is showcased by Gehrmann [44] via the Giant Language Model Test Room (GLTR) study. The idea of the research is that LLMs generate from a limited subset of the true distribution of natural language for which they have high confidence. To test whether a text is machine-generated, the authors use three approaches: (1) compute the probability of the word, (2) compute the absolute rank of a word, and (3) compute the entropy of the predicted distribution. The first two steps evaluate whether a word was sampled from a model similar to the detection model; in contrast, the last step verifies whether the previously generated context is well-known to the detection system, such that it is sure of its next prediction.

Classifier-based methods are widely spread in detection paradigms, while watermark-based methods represent an innovative alternative to the above-mentioned methods [52]. In the early days, watermark methods were used in computer vision and image processing to ensure copyright protection [53]. Recently, Kirchenbauer [54] proposed in their study the use of watermarks with LLMs, incorporating signals in generated text, which is undetectable to human observers and can be detected with open source algorithms without access to the language model API or parameters for detection. It works by selecting a randomized set of "green" tokens before a word is generated and then softly promoting the use of green tokens during sampling. It requires, however, access to the language model while generating the text.

Another detection approach based on text classification was proposed by OpenAI (https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text, accessed on 8 January 2024) and consisted in fine-tuning a GPT model with data from Wikipedia, WebText [55], and human input data to create an interface for a discrimination task using outputs produced by 34 language models. Their approach combined the classifier-based method with human evaluation to determine if a text was artificially generated. Nonetheless, this approach has some limitations. The text must have at least 1000 characters, and it was primarily trained on English corpora, making it inappropriate for multilingual use cases. Its authors recommend using the classifier only for English text since it performs significantly worse in other languages. Based on the preliminary evaluations of a set of

English texts, the model correctly identifies 26% of AI-written text (true positives) and incorrectly labels human-authored text as AI text for 9% of the texts (false positives).

Simpler classifier methods involve ML models such as XGBoost [56]. In their approach, the input features are based on the TF-IDF score and hand-crafted linguistic features based on characters and punctuation. The authors achieved an F1-score of 99% for detecting ChatGPT text. However, as a limitation, it can easily perform overfitting due to sample bias, requiring a large training dataset to overcome this drawback.

Among the various detection methods, we can also include fine-tuning language models for binary classification [2]. Solaiman et al. [2] used a sequence classifier model based on RoBERTa-base and RoBERTa-large, which achieved 95% accuracy on a GPT2 dataset detection. The advantage of this method is the bidirectionality, which allows discriminative classification models to be more powerful for detection than generative classification models.

However, despite the numerous studies targeting MGT detection, Krishna et al. [57] highlight that paraphrased text escapes the existent detectors, including watermarking, DetectGPT, or GPTZero, with an important drop in performance. Therefore, text perturbations may affect detectors' accuracy, hence the increasing need for more reliable and robust detection systems.

## 3. Method

Our method comprises two modules, namely the artificial corpus generation and the detection models (see Figure 1). Both detection models are formulated as a multiclass text classification task, leveraging a Transformer architecture and a Romanian pre-trained encoder model versus an XGBoost model fed with a selection of linguistic features.

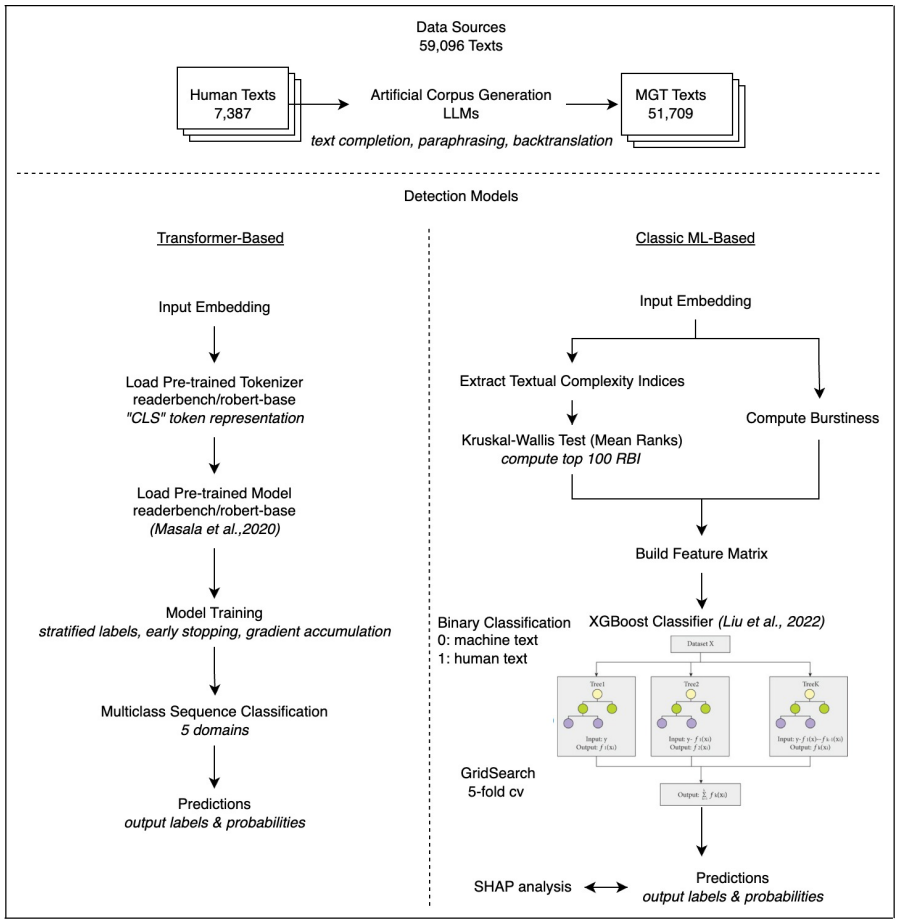


**Figure 1.** Detection methods using RoBERT [58] and XGBoost [59] models.

### *3.1. Corpus*

The corpus for this study consists of multiple datasets of comparable text lengths, both machine-generated and human-written. Experiments were performed against all datasets iteratively for a comprehensive overview. For a detector that generalizes well on several domains and writing styles, the human dataset comprises texts acquired from different domains such as Romanian literature (i.e., books), news, medical, legal, and scientific articles, obtaining a human corpus of 7387 documents across five domains, from which,

1. A total of 1401 books: 841 manually written abstracts provided by the Central University Library of Bucharest, representing descriptions of Romanian old documents (literary magazines and books dated between the 19th century and the present), and 560 books descriptions (https://cartigratis.com/, accessed on 8 January 2024);
2. A total of 4320 news articles crawled from DigiNews (https://www.digi24.ro/, accessed on 8 January 2024);
3. A total of 557 medical texts acquired from several specialized publications: 71 texts from medical scientific journals (https://srumb.ro/index.php?page=revista&spage=numere&id=9, accessed on 8 January 2024), 372 texts from scientific magazines (https://www.medichub.ro, accessed on 8 January 2024), and 114 texts from glossary of diseases (https://www.sfatulmedicului.ro/boli-si-afectiuni, accessed on 8 January 2024);
4. A total of 1000 juridical/legal texts representing Romanian law texts from Monitorul Oficial (https://monitoruloficial.ro/, accessed on 8 January 2024);
5. A total of 109 scientific articles from the Romanian Journal of Human-Computer Interaction (RoCHI) (http://rochi.utcluj.ro/, accessed on 8 January 2024).

Our primary focus when generating the artificial corpus was to create human–machine text pairs for analyzing the linguistic similarity and comparing the employed LLMs. Specifically, we were interested in capturing model-specific patterns, which resulted in the global dataset imbalance having a ratio of 1:7 for human–machine texts.

With respect to enhancing readability of the conducted experiments, we refer to the readerbench/RoGPT2-medium model from HuggingFace as RoGPT2, dumitrescustefan/gpt-neo-romanian-780m is denoted as GPT-Neo-Ro, flan-t5-small-paraphrase-ro is referred to as Flan-T5, while OpenAI's text-davinci-003 accessible via their API is referred to as davinci-003. We maintain the original names for Opus-MT and mBART.

#### 3.1.1. Text Generation Strategies

Starting from the human-authored texts, we expanded the dataset artificially via text generation, leveraging different generation techniques and state-of-the-art language models. An overview of the utilized methods and models is presented in Table 1.

**Table 1.** Methods overview for artificial text generation.

| Text Generation Method | Language Model |
|---|---|
| 1. Text Completion | RoGPT2 |
| 2. Text Completion | GPT-Neo-Ro |
| 3. Text Completion | davinci-003 |
| 4. Paraphrasing | Flan-T5 |
| 5. Backtranslation (Ro-Ru-Tr-Ro) | davinci-003 |
| 6. Backtranslation (Ro-Fr-Es-Ro) | Opus-MT |
| 7. Backtranslation (Ro-En-Ru-Ro) | mBART |

Several techniques were leveraged for text generation, such as text completion, paraphrasing, or backtranslation, to have a greater diversity of the artificially generated texts to feed our detection system.

For each generation method, a different input was provided to the model. The text completion method takes as input the first 10 words from each human text and asks the

model to continue the text with coherent paragraphs, while for paraphrasing and back-translation methods, the input is represented by the entire human written text fragments. The motivation behind choosing 10-word prompts for the text completion task is to avoid artificially inflating similarity scores in human–machine text pairs due to n-gram overlap. Using longer prompts could lead to a higher number of shared words between the human reference and the generated texts, skewing the similarity metrics by introducing a higher degree of text overlap. By limiting the prompt length to 10 words, we aimed to strike a balance where the generated texts could still be contextually meaningful, yet not excessively overlap with the reference paragraphs, ensuring a more accurate evaluation of text similarity without introducing unintended biases into the analysis.

The language models used for the text completion tasks are RoGPT2 (based on GPT2 architecture), GPT-Neo-Ro (based on GPT3 architecture), and Open AI's davinci-003 (based on GPT3.5 architecture). For the paraphrasing task, Flan-T5 (based on FLAN architecture) was leveraged. The backtranslation task applied three multilingual models, namely davinci-003 (based on GPT3.5 architecture), mBART (based on BART architecture), and Opus-MT (based on a standard Transformer architecture with six attention layers in both encoder and decoder network, having eight attention heads in each layer). The backtranslation models used three iterative translations for each model.

The choice of employing iterative translations, involving translations from the original language (Romanian) to multiple target languages and subsequently backtranslating to Romanian, is motivated by several key advantages in the context of generating an artificial corpus. First, this process induces a higher diversity of language patterns since iterative translations introduce linguistic variations, thus making the artificial corpus more representative of the linguistic variability encountered in real-case scenarios. Second, the translations may yield different word choices and contextual interpretations by involving multiple languages, contributing to richer representations in the artificial corpus. Third, iterative translations introduce controlled semantic drift and ambiguity into the text. As the text navigates multiple languages back and forth, subtle changes in meaning and context may occur. This controlled variation is valuable for simulating ambiguities present in real-world language. Fourth, the iterative translation strategy enhances generalization as the generated corpus becomes more adaptable to a range of language variations, ensuring better performance and robustness when applied to tasks involving varied linguistic inputs.

In the context of machine-generated text detection, the decision to use three iterative translations instead of just two aligns with the goal of creating a more diverse artificial corpus that further amplifies linguistic variations and introduces additional layers of text transformation. While two iterations (forward translation and backtranslation) already bring diversity, incorporating a third iteration enhances the previously introduced benefits. This additional step ultimately contributes to the model's capability to generalize effectively across a wide range of linguistic variations.

#### 3.1.2. Overview of the Generated Corpus

The AI-generated dataset was built starting from the human set using the previously introduced strategies applied to the texts from each of the five domains. The machine-generated set contains 51,709 documents across the five domains. The entire corpus reached almost 60 k documents (59,096), both human and machine-generated—the overview is presented in Table 2.

Lexical diversity was computed for each generated set to have a comparison baseline with human-authored sets. Lexical diversity represents the variety of words used in a given text and is commonly computed via the Type-Token Ratio (TTR) score, which is the ratio of the number of unique words (types) to the total number of words (tokens) in a text. A high TTR score indicates that a text has a wide range of different words (a higher diversity); in contrast, a low TTR score means the text has a limited vocabulary and may be repetitive (have a lower diversity). Lexical diversity and TTR score are important measures in linguistics in general and for our experiment, as they facilitate analysis and comparison

of the language use and generation style across different language generation models. Based on the average TTR scores for our generated datasets, Flan-T5 and davinci-003 (on back-translation task) produce the most diversified text, superior in diversity compared to human writings.

For a comparison baseline, both models leverage the same data split for train-test partitions and use stratified labels.

**Table 2.** MGT dataset: human-written and machine-generated texts.

| Domain | Method | Model | Avg TTR | Doc Count | Aggregate |
|---|---|---|---|---|---|
| Books | Human | Human | 0.7447 | 1401 | 11,208 |
| | Completion | RoGPT2 | 0.6615 | 1401 | |
| | Completion | GPT-Neo-Ro | 0.7011 | 1401 | |
| | Completion | davinci-003 | 0.6125 | 1401 | |
| | Backtranslation | davinci-003 | 0.7652 | 1401 | |
| | Paraphrasing | Flan-T5 | 0.8708 | 1401 | |
| | Backtranslation | Opus-MT | 0.7581 | 1401 | |
| | Backtranslation | mBART | 0.7379 | 1401 | |
| News | Human | Human | 0.6510 | 4320 | 34,560 |
| | Completion | RoGPT2 | 0.6762 | 4320 | |
| | Completion | GPT-Neo-Ro | 0.6867 | 4320 | |
| | Completion | davinci-003 | 0.6508 | 4320 | |
| | Backtranslation | davinci-003 | 0.7798 | 4320 | |
| | Paraphrasing | Flan-T5 | 0.8389 | 4320 | |
| | Backtranslation | Opus-MT | 0.6589 | 4320 | |
| | Backtranslation | mBART | 0.7024 | 4320 | |
| Medical | Human | Human | 0.6911 | 557 | 4456 |
| | Completion | RoGPT2 | 0.6795 | 557 | |
| | Completion | GPT-Neo-Ro | 0.6893 | 557 | |
| | Completion | davinci-003 | 0.6262 | 557 | |
| | Backtranslation | davinci-003 | 0.7510 | 557 | |
| | Paraphrasing | Flan-T5 | 0.8503 | 557 | |
| | Backtranslation | Opus-MT | 0.7490 | 557 | |
| | Backtranslation | mBART | 0.7618 | 557 | |
| Legal | Human | Human | 0.7264 | 1000 | 8000 |
| | Completion | RoGPT2 | 0.6542 | 1000 | |
| | Completion | GPT-Neo-Ro | 0.6880 | 1000 | |
| | Completion | davinci-003 | 0.5828 | 1000 | |
| | Backtranslation | davinci-003 | 0.7987 | 1000 | |
| | Paraphrasing | Flan-T5 | 0.8418 | 1000 | |
| | Backtranslation | Opus-MT | 0.7231 | 1000 | |
| | Backtranslation | mBART | 0.7514 | 1000 | |
| RoCHI | Human | Human | 0.6234 | 109 | 872 |
| | Completion | RoGPT2 | 0.6901 | 109 | |
| | Completion | GPT-Neo-Ro | 0.5460 | 109 | |
| | Completion | davinci-003 | 0.5810 | 109 | |
| | Backtranslation | davinci-003 | 0.7514 | 109 | |
| | Paraphrasing | Flan-T5 | 0.8356 | 109 | |
| | Backtranslation | Opus-MT | 0.6032 | 109 | |
| | Backtranslation | mBART | 0.7477 | 109 | |
| Total | | | | | 59,096 |

### *3.2. Detection Models*

We built a comprehensive pipeline dedicated to the training and assessment of a text classifier tasked with predicting the text generation model associated with a given text fragment. This involves two distinct classification techniques, each based on different

architectures. The primary objective is to evaluate and compare their efficiency, particularly when incorporating additional features such as linguistic complexity into the analysis. The comprehensive approach taken in this study aims to provide insights into the effectiveness of diverse classification methodologies while considering nuanced linguistic attributes, thus contributing to a stronger understanding of AI-text detection.

#### 3.2.1. Transformer-Based Model

We model the task as a multiclass classification problem that leverages a Romanian pre-trained encoder model, readerbench/RoBERT-base (https://huggingface.co/readerbench/RoBERT-base, accessed on 8 January 2024). The model is trained for four epochs using a data split of 80-20 ratio for the train-validation sets, using the maximum length of the model of 512 tokens, with a batch size of 32 and a learning rate of $1 \times 10^{-5}$. We use specific parametrization during the model's training to overcome the challenges of overfitting and of the unbalanced dataset. As such, the model uses L2 regularization with a weight decay of 0.1, adding a penalty term to the loss function that discourages large weights. Moreover, we use early stopping to monitor the model's performance on the validation set during the training, stopping the training when the validation starts to degrade, which indicates the model begins to overfit. This helps denoise the model learning and helps generalize better on unseen data. We also leverage gradient accumulation. The weight updates become more stable by accumulating gradients during several mini-batches (i.e., two in our case). This technique is helpful when dealing with large batch sizes that do not fit into the available memory.

#### 3.2.2. Classic ML-Based Model

This method leverages the ReaderBench textual complexity indices (RBIs) (https://github.com/readerbench/ReaderBench/wiki/Textual-Complexity-Indices, accessed on 8 January 2024), available for Romanian language. The first step in our classic ML-based approach is to compute and extract the most relevant textual complexity indices from the dataset containing both human and machine-generated texts. Since most RBIs are non-normally distributed on the train partition, we employ the non-parametric Kruskal–Wallis test with mean ranks to determine the most relevant indices in predicting significant differences between human-written and machine-generated texts. From the RBI collection, the top 100 indices with the most significant differences in mean ranks are selected. These indices are considered the most relevant to distinguish between human and machine-generated texts and are further used in the detection algorithm to build the feature selection given as input to the classification model.

Additionally, we compute the burstiness measure, which shows the variation in word lengths within a text and may be used to identify irregularities or changes in word lengths. As such, we compute the mean word length as the average length of all words in the text. Further, we compute the standard deviation of word lengths to measure the extent to which word lengths deviate from the mean word length. The standard deviation measures the dispersion of word lengths in the text. Finally, the burstiness is determined by dividing the standard deviation by the mean word length. This ratio represents how much the word lengths vary relative to their average length. Higher burstiness values indicate greater variability in word lengths within the text, while lower values indicate more uniform word lengths. Further, the feature selection is built by combining the bustiness score with the top 100 textual complexity indices, which serve as input to the XGBoost classification model. This enhances the model's capability to discriminate between human and machine-generated texts.

The selection of XGBoost as our classic ML model for the binary classification task of detecting machine-generated texts was based on its superior performance and versatility. XGBoost, an ensemble learning algorithm, has achieved noteworthy success in various domains [60,61], particularly excelling in binary classification tasks. Its ability to handle complex relationships within the data, manage large datasets efficiently, and mitigate

overfitting makes it a robust choice for the given task. Furthermore, XGBoost is known for its flexibility in handling diverse feature types and managing imbalanced datasets, which is crucial when dealing with text data. The decision to opt for XGBoost is supported by its proven track record in achieving high accuracy, precision, and recall in comparable applications [60]. While other ML classifiers could be employed, the selection of XGBoost is justified by its empirical effectiveness, which aligns with the objectives of this study, ensuring a robust and reliable approach to text classification in the context of Romanian binary text classification.

Due to the high degree of imbalance between human-written and artificially generated texts, we leverage a weighted XGBoost classifier to mitigate this issue by incorporating weighting and regularization parameters. Moreover, we perform hyperparameter tuning by defining a grid search with five-fold cross-validation to find the best hyperparameters. We select the best classifier with the optimal parameters from grid search (see Table 3), and the classifier is further trained on the entire training dataset. Next, the classifier is evaluated on the test dataset, and the class probabilities are predicted.

**Table 3.** Grid Search best parameters for ML-model.

| Domain | learning_rate | max_depth | n_estimators | reg_alpha | reg_lambda | scale_pos_weight |
|---|---|---|---|---|---|---|
| Books | 0.1 | 5 | 100 | 0.001 | 0.01 | 1 |
| Legal | 0.1 | 3 | 100 | 0.001 | 0.1 | 3 |
| Medical | 0.1 | 4 | 50 | 0.1 | 0.1 | 3 |
| News | 0.1 | 5 | 200 | 0.001 | 0.1 | 3 |
| RoCHI | 0.1 | 3 | 200 | 0.1 | 0.1 | 5 |

## 4. Results

Given the results presented in Table 4, both detection methods are effective instruments for discerning human-authored from AI-generated texts across different domains. The F1 scores are generally good, showing minor differences between the two methods in terms of performance. However, the number of misclassified texts, specifically for the human category, varies across domains, behavior that is motivated by class imbalance. The random chance for predicting each class, given the class imbalance of 1:7 (machine-generated to human-generated texts), is 12.25%. Table 4 presents the cumulative results, based on which the Transformer-based method outperformed the classic ML-based method on two domains out of five, being better in distinguishing AI texts from human texts for legal and news domains, while the linguistic features extracted in the classic ML-based approach contributed to better results for books, medical, and scientific (RoCHI) domains.

**Table 4.** Detection results (bold marks the best scores per domain).

| | Transformer-Based | | | Classic ML-Based | | |
|---|---|---|---|---|---|---|
| **Domain** | **P** | **R** | **F1** | **P** | **R** | **F1** |
| Books | 0.90 | 0.89 | 0.89 | **0.92** | **0.92** | **0.90** |
| Legal | **0.91** | **0.90** | **0.90** | 0.85 | 0.85 | 0.85 |
| Medical | 0.92 | 0.92 | 0.91 | **0.96** | **0.96** | **0.96** |
| News | **0.96** | **0.96** | **0.96** | 0.88 | 0.88 | 0.88 |
| RoCHI | 0.88 | 0.85 | 0.85 | **0.92** | **0.93** | **0.91** |
| Micro | | | 0.60 | | | 0.59 |
| Macro | | | 0.90 | | | 0.91 |

Transformer-based model. The multiclass classifier performs well across different categories with high precision (P), recall (R), and F1-scores as highlighted by confusion matrices (see Figure 2 for the books domain, while the matrices for other domains can be consulted in Appendix A, Figure A1). While evaluating the model's performance, it is important to consider the dataset split, the sampling technique, and the class imbalance.

The classification is less performant on the subset with fewer samples (i.e., RoCHI), causing more confusion and an F1-score of 0.85, while the classification performs best for the subset with the highest number of samples (i.e., news), achieving an F1-score of 0.96.

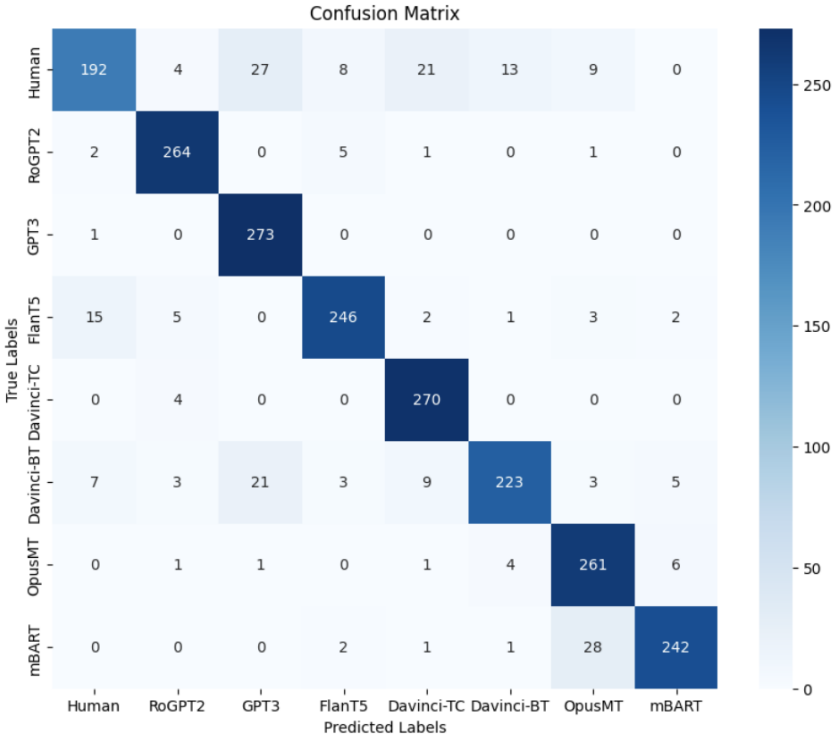


**Figure 2.** Transformer-based model: confusion matrix for the books domain.

Classic ML-based model. The human texts provide less accurate results because the number of human samples in the dataset is seven times less than the total number of AI texts. Analyzing the results by domains, the classic ML-based algorithm achieves the highest F1 score for the medical domain (0.96), followed by the RoCHI domain, which exhibits an F1 of 0.92. The precision for AI texts is particularly high (0.93), with only a few misclassified AI texts (9 out of 144). Books domain follows with an F1 of 0.90. It correctly identifies both human and machine-generated texts with high accuracy (0.92). The legal domain exhibits reasonable performance as well, with an F1 of 0.85. We observe the same pattern regarding human misclassified texts compared to AI texts motivated by class imbalance. The news domain achieves an F1 score of 0.87.

Furthermore, we introduce SHAP (Shapley Additive exPlanations) [62] to examine the contribution of individual features (RBI indices and burstiness) to the model's predictions. SHAP provides insights into why a particular prediction was made by attributing a portion of the prediction to each feature. SHAP values are based on cooperative game theory to allocate contributions among features, helping us understand the impact of each feature on the model's decision. The SHAP summary plots (see Figure 3 for the books domain, while SHAP for other domains can be consulted in Appendix B, Figure A2) show the impact of each feature on the model's predictions for the respective domain. The features are ranked by their importance, with the most important features at the top. The color of the dots indicates the direction of the impact, with red indicating a positive impact and blue indicating a negative impact. The size of the dots indicates the magnitude of the impact. To determine which class is influenced most by a particular feature, we can look at the color of the dots. For example, if the dots for a particular feature are mostly red, then that feature is likely to be associated with the positive class (human text). On the contrary, if the dots are mostly blue, then that feature is likely to be associated with the negative class (machine text).

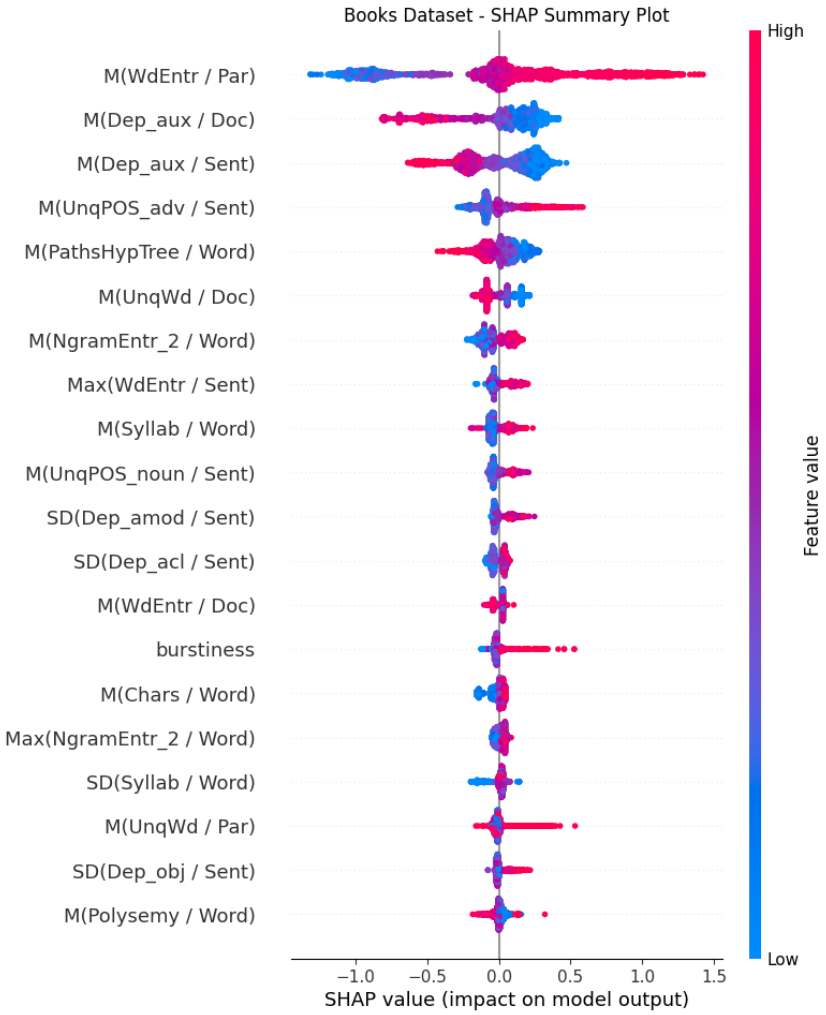


**Figure 3.** Classic ML-based model: SHAP summary plot for the books domain.

Based on the charts, in the news domain, the most important features are the number of nouns per paragraph, the number of unique words in the document, the word entropy per paragraph, and the number of words per paragraph. The prediction is less influenced by the number of punctuation marks per paragraph or by the number of words in the document. This means the model predictions for the news domain are the most influenced by morphology and surface indices. The books domain is most influenced by surface indices like word entropy per paragraph and syntax indices like dependencies, but also by morphology features like unique words for adverbs per sentence. The predictions are less influenced by word polysemy or number of syllables in words. For the legal domain, the model output is most influenced by the number of sentences in the document, the number of syllables in words, and the number of connectors in the sentences. This reveals that the syntax, word, and discourse elements are the differentiators, while the cohesion between the first and last text element has a lower impact on the prediction. The main feature of the medical domain is represented by a syntactic measure, namely case dependency per paragraph, followed by a surface index, namely word entropy per paragraph. This validates the assumption that more complex texts contain more information and more diverse concepts. The third most significant index is burstiness, which shows the variation in word lengths within a text and may be used to identify irregularities or changes in word lengths. The scientific domain is most influenced by a syntactic index, namely case dependency per paragraph, followed by word entropy per sentence, and a morphology index, namely unique nouns in the paragraph.

The SHAP analysis provides insights into the top linguistic features contributing the most to the model's predictions. As a general observation, we may notice among all domains that the most significant indices were the surface indices, followed by morphology and syntax, and then discourse structure.

## 5. Discussion

Our results highlight the effectiveness of both the Transformer-based and classic ML-based methods in discriminating between human-authored and AI-generated texts across diverse domains. The performance differences observed in the individual domains provide valuable insights into the strengths and limitations of each approach, further pinpointing potential paths for further refinement and application.

One key observation is the variance in misclassified texts, particularly from the human category, across different domains. This behavior is attributed to class imbalance, where the number of human samples is significantly lower than that of AI texts. The classic ML-based model, in particular, shows sensitivity to this imbalance, as evidenced by the notable improvement in results when experiments were conducted with an equal number of documents for both classes. This underscores the importance of addressing the class distribution challenge in training datasets to enhance the model's accuracy, especially in scenarios where human-authored texts are underrepresented.

Moreover, examining SHAP values for the classic ML-based model offers valuable insights into the features contributing significantly to classification decisions. The consistency of the average word entropy per paragraph as a strong indicator across multiple domains suggests its importance in machine-generated text detection.

In addition to evaluating the performance of the Transformer-based and classic ML-based models in discerning human-authored from AI-generated texts, we further research two methods to gain a comprehensive understanding of the classification outcomes. First, we conduct a human–machine textual similarity analysis to assess the degree of resemblance between human and machine-generated texts. This analysis involves leveraging similarity metrics to quantify the textual proximity between the human texts and their actual counterparts. By exploring the textual distinctions, we aim to discern any patterns or divergences in similarity across different domains, providing insights into the models' interpretability and their ability to capture subtle variations in writing styles.

Furthermore, we extend our investigation to analyze the impact of textual complexity indices for both datasets. Textual complexity metrics, such as syntactic and semantic features, are key in characterizing the complexities of written content. By examining indices such as sentence length, vocabulary richness, and syntactic structures, we seek to clarify the distinct patterns exhibited by human and machine-generated texts in each domain. This dual analysis offers a multi-faceted perspective, allowing us to not only quantify the degree of similarity but also investigate the inherent complexities present in the textual content, thus enriching our understanding of the models' performance across diverse domains.

### *5.1. Human–Machine Textual Similarity*

For a better understanding of the similarity between human-written and artificially generated texts, a series of statistical similarity metrics were computed, such as cosine similarity [63], BLEU scores [33], and ROUGE scores [34], followed by non-parametric statistical tests to determine the differences or patterns in the similarity scores, while also extracting and comparing various textual complexity indices between human and machine-generated sets of documents.

Cosine similarity measures the relatedness of documents using a bag-of-words representation that considers term frequency. The main advantages of this approach are its efficiency for high-dimensional data, such as text, and its insensitivity to document length, while as for disadvantages, it ignores semantic meaning and word order. BLEU is a standard machine translation metric that measures the precision of n-grams (word sequences) in the generated text. The main benefit is the penalization of overusing common words,

while the shortcoming is represented by its sensitivity to the reference length, as well as its disregard for recall and semantic meaning. The ROUGE scores are a set of metrics used for evaluating text quality by comparing it to a reference text by measuring n-gram overlap (ROUGE-N) and the longest common subsequence (ROUGE-L). As an advantage, ROUGE is sensitive to content overlap and partial matches, while as a disadvantage, it may not capture the semantic meaning well due to its limitation to the recall and precision of n-grams. Therefore, analyzing the three scores together provides a more grounded view of text similarity, covering different aspects, such as incorporating structural and distributional aspects of the text (cosine), adding a layer of specificity via analyzing the presence of key phrases and content overlap (ROUGE), and offering a view on how well machine-generated texts align with human references in terms of specific word sequences (BLEU).

The previous three similarity scores (i.e., cosine similarity, BLEU score, and ROUGE score) were computed between human-written documents and artificially generated texts. Cosine similarity measures the cosine of the angle between two vectors and denotes the similarity between a pair of texts, while BLEU and ROUGE scores evaluate the quality of machine-generated texts by comparing them to reference (human) texts.

The scores distribution for each domain is illustrated in Appendix C, Figures A3–A7 for each dataset (e.g., books, news, medical, legal, and scientific/RoCHI), and are further correlated with mean and standard deviation presented in Table 5 for a comprehensive similarity overview. Models with higher mean scores are generally better at capturing textual similarity according to the respective metrics. Nevertheless, we should also analyze the standard deviations to understand the consistency of each model's performance. Lower standard deviations suggest more consistent performance, while higher standard deviations indicate greater score variability.

**Table 5.** Mean (M) and standard deviation (SD) values for similarities per model.

| Domain | Model | Cosine M (SD) | BLEU M (SD) | ROUGE M (SD) |
|---|---|---|---|---|
| Books | RoGPT2 | 0.359 (0.118) | 0.138 (0.124) | 0.251 (0.120) |
| | GPT-Neo-Ro | 0.358 (0.076) | 0.138 (0.018) | 0.148 (0.029) |
| | Flan-T5 | 0.545 (0.239) | 0.296 (0.237) | 0.395 (0.243) |
| | Davinci-TC | 0.375 (0.082) | 0.164 (0.048) | 0.202 (0.049) |
| | Davinci-BT | 0.410 (0.131) | 0.079 (0.063) | 0.133 (0.075) |
| | Opus-MT | 0.557 (0.147) | 0.237 (0.098) | 0.314 (0.099) |
| | mBART | 0.440 (0.141) | 0.118 (0.073) | 0.178 (0.081) |
| Legal | RoGPT2 | 0.467 (0.127) | 0.196 (0.131) | 0.313 (0.125) |
| | GPT-Neo-Ro | 0.417 (0.088) | 0.151 (0.037) | 0.187 (0.051) |
| | Flan-T5 | 0.680 (0.160) | 0.405 (0.194) | 0.519 (0.170) |
| | Davinci-TC | 0.418 (0.160) | 0.139 (0.019) | 0.186 (0.031) |
| | Davinci-BT | 0.356 (0.121) | 0.054 (0.062) | 0.112 (0.078) |
| | Opus-MT | 0.530 (0.167) | 0.218 (0.124) | 0.294 (0.128) |
| | mBART | 0.348 (0.160) | 0.065 (0.073) | 0.105 (0.097) |
| Medical | RoGPT2 | 0.375 (0.107) | 0.077 (0.072) | 0.212 (0.064) |
| | GPT-Neo-Ro | 0.387 (0.082) | 0.140 (0.013) | 0.137 (0.018) |
| | Flan-T5 | 0.471 (0.234) | 0.219 (0.199) | 0.298 (0.226) |
| | Davinci-TC | 0.403 (0.077) | 0.132 (0.034) | 0.182 (0.032) |
| | Davinci-BT | 0.421 (0.121) | 0.075 (0.055) | 0.139 (0.068) |
| | Opus-MT | 0.273 (0.127) | 0.019 (0.064) | 0.026 (0.081) |
| | mBART | 0.392 (0.154) | 0.098 (0.064) | 0.143 (0.075) |

**Table 5.** *Cont.*

| Domain | Model | Cosine M (SD) | BLEU M (SD) | ROUGE M (SD) |
|---|---|---|---|---|
| News | RoGPT2 | 0.414 (0.102) | 0.075 (0.066) | 0.208 (0.056) |
| | GPT-Neo-Ro | 0.421 (0.081) | 0.145 (0.012) | 0.141 (0.019) |
| | Flan-T5 | 0.530 (0.238) | 0.224 (0.193) | 0.316 (0.221) |
| | Davinci-TC | 0.424 (0.082) | 0.133 (0.032) | 0.186 (0.030) |
| | Davinci-BT | 0.525 (0.104) | 0.106 (0.065) | 0.240 (0.084) |
| | Opus-MT | 0.671 (0.112) | 0.285 (0.092) | 0.360 (0.089) |
| | mBART | 0.688 (0.098) | 0.310 (0.093) | 0.384 (0.091) |
| RoCHI | RoGPT2 | 0.415 (0.082) | 0.050 (0.044) | 0.198 (0.025) |
| | GPT-Neo-Ro | 0.402 (0.069) | 0.139 (0.009) | 0.132 (0.012) |
| | Flan-T5 | 0.471 (0.208) | 0.165 (0.165) | 0.243 (0.203) |
| | Davinci-TC | 0.414 (0.079) | 0.121 (0.020) | 0.171 (0.017) |
| | Davinci-BT | 0.446 (0.105) | 0.059 (0.036) | 0.130 (0.051) |
| | Opus-MT | 0.661 (0.101) | 0.281 (0.076) | 0.353 (0.068) |
| | mBART | 0.451 (0.139) | 0.113 (0.063) | 0.162 (0.071) |

Summarizing these results, we notice that Flan-T5 produces texts that are similar to human-produced ones while also having the highest variability (standard deviation), thus denoting potential inconsistencies.

Since our data do not follow a normal distribution across the five domains, non-parametric statistical methods are required to compare and analyze the results of human and machine similarity. These tests are valuable when working with similarity scores, allowing us to make statistical inferences about the performance of different text generation models without making strong assumptions about the underlying data distribution.

The Friedman test is a non-parametric statistical test that represents an alternative to repeated measures analysis of variance (ANOVA) and represents a measure of the variability between multiple models. Table 6 shows the Friedman test results for the five domains applied to cosine similarity scores.

**Table 6.** Friedman statistics across domains for cosine similarity scores.

| Domain | Friedman Statistic | *p*-Value |
|---|---|---|
| Books | 2275.96 | <0.001 |
| Legal | 2378.24 | <0.001 |
| Medical | 545.72 | <0.001 |
| News | 13,458.79 | <0.001 |
| RoCHI | 213.61 | <0.001 |

In hypothesis testing, a low *p*-value (typically less than 0.05) means the null hypothesis is rejected, and significant differences exist among the tested models. According to the results across all five domains, we have high Friedman scores and low *p*-values, which suggests the compared language models perform differently in each domain, and further analysis or pairwise comparisons can be conducted to understand the nature of these differences.

We further perform the Wilcoxon test, targeting pairwise comparisons. The Wilcoxon signed-rank test is used to compare two paired groups and check if there are significant differences between them. Similarly to the Friedman test, the Wilcoxon test is used in cases where data does not follow a normal distribution, and it denotes the degree of differences between the two pairs analyzed. The Wilcoxon test results for our dataset across the five domains for cosine similarity scores are presented in Appendix D, Figure A8. The high Wilcoxon scores in the matrix (see Figure A8) argue substantial differences in cosine similarity scores between pairs of models. There is strong evidence to suggest that the variations in performance are statistically significant.

### *5.2. Textual Complexity Indices*

To explore the linguistic differences between human-written and machine-generated texts, we additionally leverage the ReaderBench textual complexity indices (RBI) for Romanian (https://github.com/readerbench/ReaderBench/wiki/Textual-Complexity-Indices, accessed on 8 January 2024). The selection of the top three textual complexity indices (see Table 7) was determined through the Kruskal–Wallis statistical test. The indices that exhibited the most notable variations among the domains were selected, providing valuable insights into the distinctive linguistic features within each domain.

**Table 7.** Top indices per domain, where I, II, and III represents the rank.

| Domain | I | II | III |
|---|---|---|---|
| Books | M(WdEntr/Par) | Max(WdEntr/Par) | Max(WdEntr/Sent) |
| Legal | M(Dep_case/Sent) | M(Connector_conj/Sent) | M(Dep_nmod/Sent) |
| Medical | Max(Dep_case/Par) | M(Dep_case/Par) | M(UnqWd/Par) |
| News | M(Wd/Par) | Max(Wd/Par) | M(UnqWd/Par) |
| RoCHI | M(Dep_case/Par) | Max(Dep_case/Par) | M(Connector_conj/Par) |

The analysis of the top three ranking textual complexity indices across different domains yields valuable insights into the key features influencing the prediction process of the classifier. While the specific indices vary between domains, a consistent pattern emerges regarding the importance of certain features across diverse datasets. For the books domain, the model deems word entropy per paragraph and per sentence as the most significant factors. In legal texts, the focus shifts to dependencies per sentence, with a notable emphasis on the number of conjunctions per sentence. The medical domain distinguishes itself through lexical diversity, particularly in terms of specific dependencies and the number of unique words at the paragraph level. Similarly, the news domain emphasizes lexical diversity, as evidenced by the number of words and unique words per paragraph. Scientific publications, on the other hand, prioritize specific dependencies and the number of conjunctions per paragraph as the most influential indices. It is noteworthy that despite the divergence in the top indices, the underlying consistency in the importance of certain features across domains underscores the robustness of these linguistic patterns in shaping the classifier's predictions. The provided table summarizes the top indices for each domain, offering a clear reference for understanding the observed linguistic distinctions.

The distribution of values for these textual complexity indices within each dataset and domain is depicted in Figures A9–A13 in Appendix E. This provides a comprehensive overview of the variations in the most critical statistical features present in both human-written and machine-generated texts.

## 6. Conclusions

To conclude, our study introduced two distinct detection models designed for identifying machine-generated text in the Romanian language, employing different architectures—the first based on Transformers and the second based on decision trees with linguistic feature extraction. Leveraging a substantial corpus of 60,000 Romanian texts generated through diverse methods such as text completion, paraphrasing, and backtranslation, utilizing state-of-the-art large language models, we conducted extensive experimental studies. Additional investigations aimed to evaluate text similarity between human-authored and machine-generated texts, considering both statistical and linguistic perspectives. The findings underscored that artificially generated texts tend to exhibit higher complexity, while human-written texts showcase greater diversity from a lexical standpoint.

The comparative analysis revealed the superiority of the classic ML-based detection model over the Transformer model in three out of the five domains, as indicated by its highest F1 score (0.96 for detecting texts in the medical domain). The XGBoost model that considered linguistic features exhibited a macro F1 of 0.91, while the Transformer model

reached an F1 of 0.90. Despite these advancements, it is crucial to acknowledge that no existing detection system is foolproof, emphasizing the ongoing need for continuous investment in developing reliable detection algorithms. The dynamic nature of text generation methods and evolving language models necessitates a proactive approach in adapting detection mechanisms to remain ahead of emerging challenges.

Moving forward, future research endeavors could explore innovative techniques for enhancing the robustness of detection models. One path is the integration of advanced NLP techniques to discern more subtle nuances in language use. Additionally, investigating the potential impact of domain-specific linguistic characteristics on detection accuracy could provide valuable insights. Moreover, the exploration of ensemble models combining the strengths of different architectures may contribute to even more effective and versatile detection systems. Also, investigating the generalization capabilities of the detection models across diverse linguistic styles and writing conventions can provide insights into their adaptability. Exploring the robustness of the models against adversarial attacks, where subtle manipulations are made to deceive the system, is another option for future research. Such explorations could lead to developing more resilient detection mechanisms capable of withstanding sophisticated adversarial attempts in the evolving landscape of machine-generated text. Sustained efforts in research and development are essential to meet the growing demand for reliable and adaptive detection algorithms.

**Author Contributions:** Conceptualization, M.N. and M.D.; methodology, M.N. and M.D.; software, M.N.; validation, M.N.; formal analysis, M.N.; investigation, M.N.; resources, M.N.; data curation, M.N.; writing—original draft preparation, M.N.; writing—review and editing, M.D.; visualization, M.N.; supervision, M.D.; project administration, M.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** We release as open-source the dataset (https://huggingface.co/datasets/readerbench/ro-human-machine-60k, accessed on 8 January 2024), and the codebase (https://github.com/readerbench/ro-mgt-detection, accessed on 8 January 2024). Additional information is available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| AI | Artificial Intelligence |
| LLM | Large Language Models |
| MGT | Machine-Generated Text |
| ML | Machine Learning |
| MLM | Masked Language Model |
| NLG | Natural Language Generation |
| NLP | Natural Language Processing |
| P | Precision |
| R | Recall |
| RBI | ReaderBench Indices |
| TF-IDF | Term-Frequency Inverse Document Frequency |
| TTR | Type-Token Ratio |

## Appendix A. Transformer-Based Model: Confusion Matrices per Domain

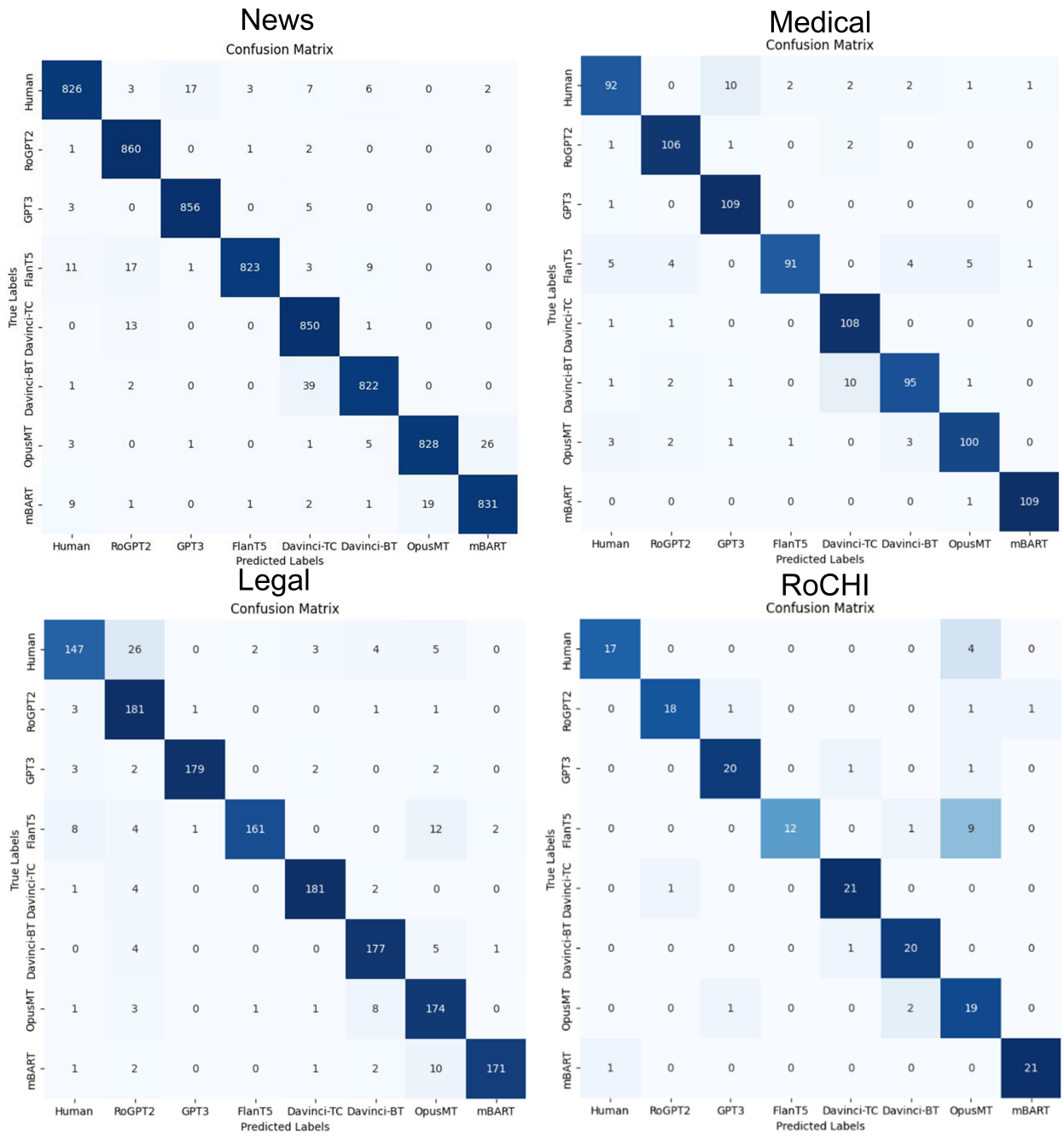


**Figure A1.** Transformer-based model: confusion matrices per domains.

## Appendix B. Classic ML-Based Model: SHAP Summary Plots per Domain

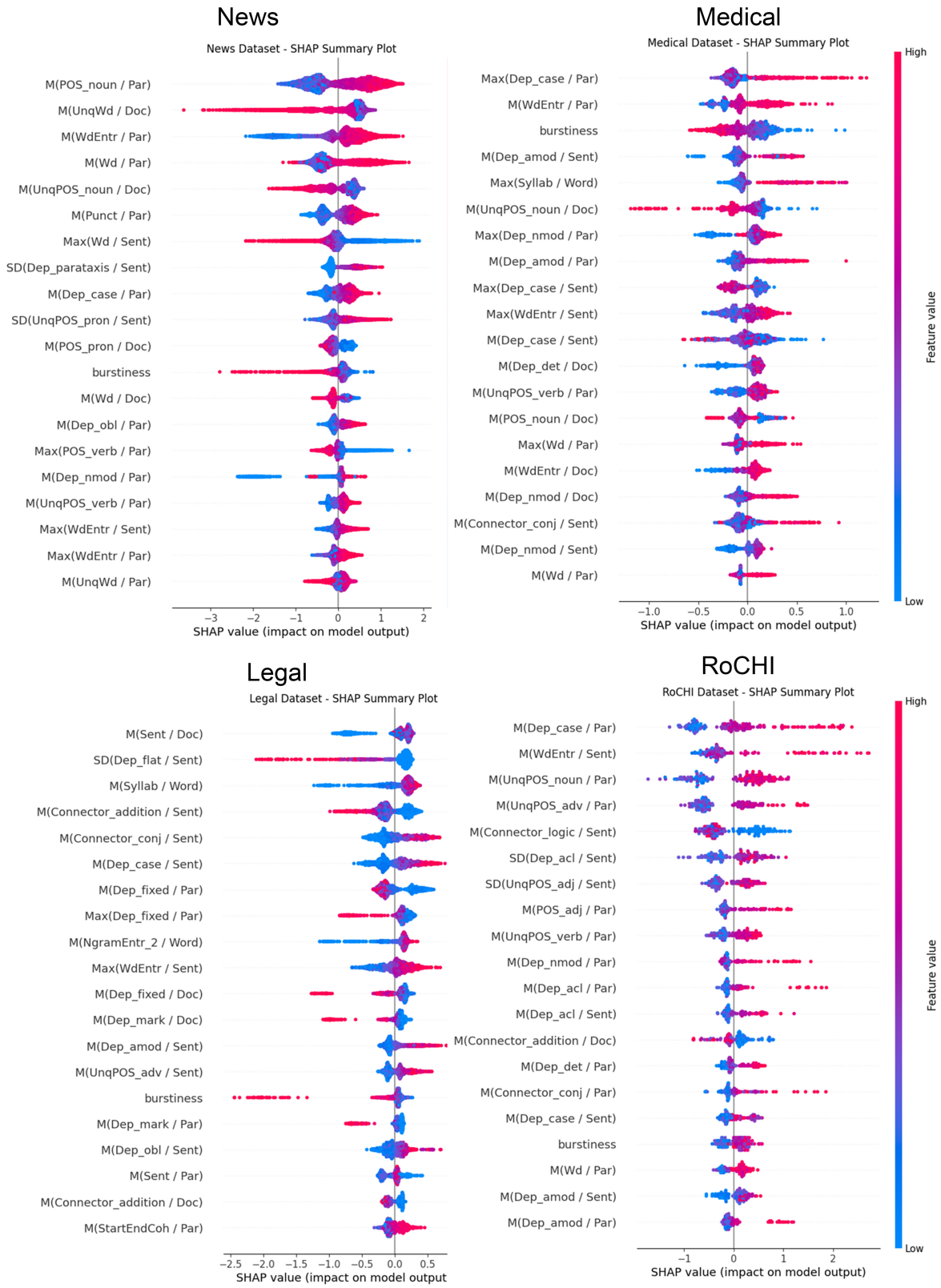


**Figure A2.** Classic ML-based model: SHAP summary plot per domain.

## Appendix C. Statistical Similarity for Artificially-Generated Texts

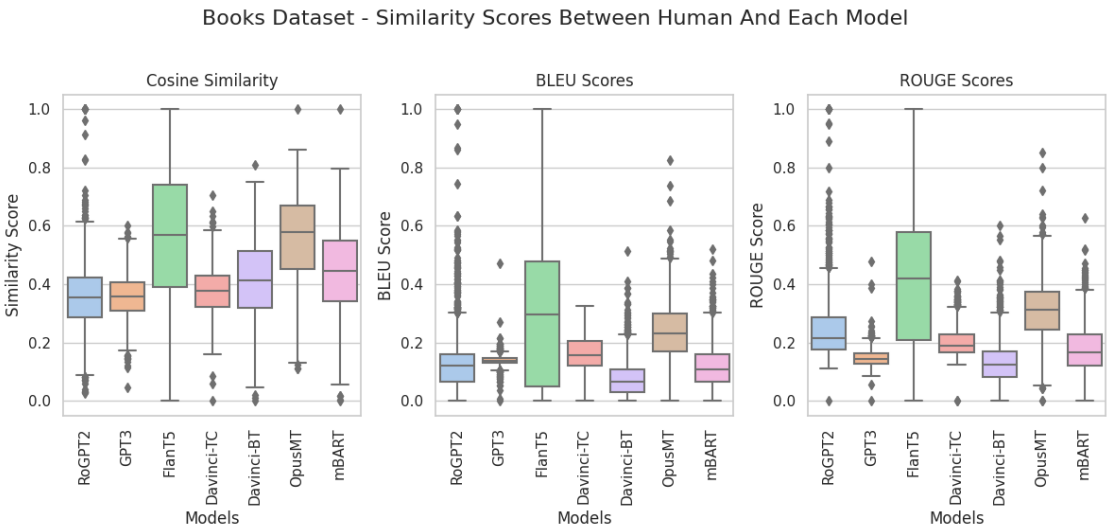


**Figure A3.** Text similarity metrics for books domain.

Discussion for books dataset textual similarity results:

- Cosine similarity—as revealed by the cosine similarity results, Flan-T5 has the highest mean cosine similarity score (0.5458), indicating it produces text that is more similar to human-written texts. Opus-MT also has a relatively high mean score (0.5578), followed closely by davinci-003 (on the backtranslation task, 0.4106). mBART and RoGPT2 have mean scores above 0.35, suggesting moderate performance, while GPT-Neo-Ro and davinci-003 (on text completion task) have lower mean scores. From a variability point of view, Flan-T5 has the highest standard deviation (0.2399), which denotes a wide range of scores and potential inconsistency. Opus-MT also has a relatively high standard deviation (0.1470), suggesting variability in its performance, while the most consistent models for the books dataset are considered GPT-Neo-Ro, davinci-003 (on text completion task) and RoGPT2, which expose lower standard deviations and variability in terms of performance.
- BLEU—BLEU scores reveal Flan-T5 with the highest mean BLEU score (0.2960), demonstrating better performance on average. Opus-MT has the second highest score (0.2374), followed by models with intermediate performance like davinci-003 (on text completion task). GPT-Neo-Ro and RoGPT2 show lower performance, while davinci-003 (on the backtranslation task) and mBART have the lowest scores. Concerning the variability of the models, Flan-T5 has the highest standard deviation (0.2372), illustrating variability in its BLEU scores. Lower variability and better consistency are shown by GPT-NEo-Ro, Opus-MT, RoGPT2, and davinci-003 (on text completion task).
- ROUGE—concerning ROUGE score results, Flan-T5 is also on top of the list with the highest mean score, followed by Opus-MT, while the lowest mean scores are registered for the mBART model. From a variability perspective, Flan-T5 shows the highest standard deviation (0.2437), hence the biggest variability in its ROUGE scores, while among the consistent models are GPT-Neo-Ro, Opus-MT, and RoGPT2.

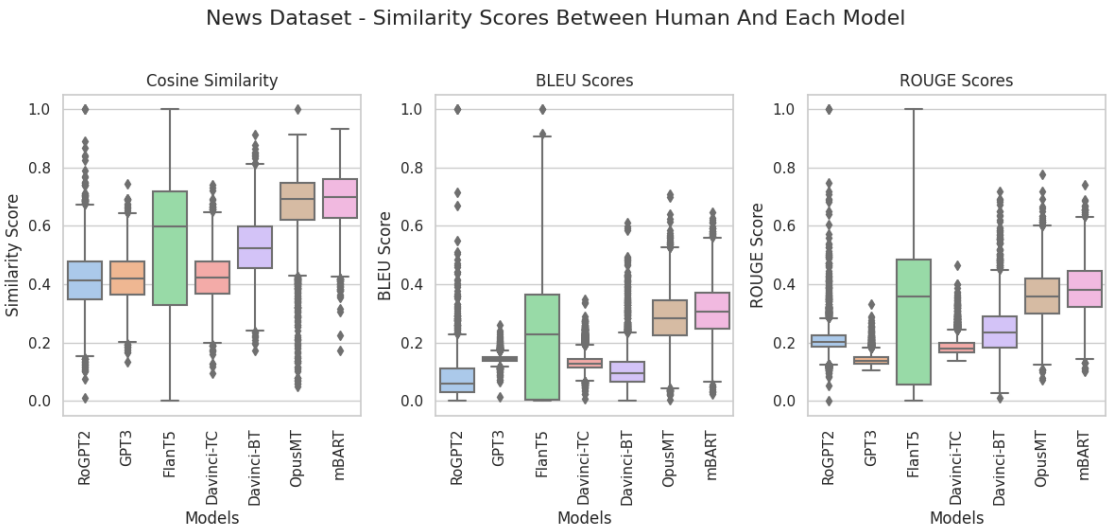


**Figure A4.** Text similarity metrics for news domain.

Discussion for news dataset textual similarity results:

- Cosine similarity—among the cosine scores, mBART has the highest mean score (0.6888), which indicates that, on average, it produces the most human-like texts. Opus-MT follows closely with a mean score of 0.6717. GPT-Neo-Ro, RoGPT2, and davinci-003 have mean scores above 0.4 but below the top-performing models. In terms of variability, Flan-T5 has the highest standard deviation (0.2389), which indicates a wide range of scores and, hence, higher variability. A more consistent performance is shown by GPT-NEo-Ro and RoGPT2.
- BLEU—similar to cosine scores result, mBART is at the top of the list with the highest mean BLEU score (0.3108), followed by Opus-MT (0.2854). The lowest BLEU mean scores are reached for RoGPT2 and davinci-003 (on the backtranslation task). Similar to the cosine analysis, variability scores (standard deviations) show Flan-T5 in the top list for inconsistent results, while the more consistent models were GPT-Neo-Ro, RoGPT2, and davinci-003.
- ROUGE—mBART shows the highest mean ROUGE score (0.3843) and a relatively high variability illustrated by its standard deviation score. On the contrary, GPT-Neo-Ro has the lowest mean score for ROUGE and is among the more stable models, having a lower standard deviation and a more consistent performance.


**Figure A5.** Text similarity metrics for medical domain.

Discussion for medical dataset textual similarity results:

- Cosine similarity—the top of the list model for cosine similarity scores mean is Flan-T5 (0.4716), followed by davinci-003 (on backtranslation task), which shows the most similar texts to human-produced texts. The least similar texts are produced by mBART and Opus-MT models, which exhibit mean scores above 0.27. In terms of variability, Flan-Ts is also the most variable model, as shown by its standard deviation (0.2347), while GPT-Neo-Ro and RoGPT2 show more consistent results.
- BLEU—Flan-T5 has the highest mean BLEU score (0.2193), illustrating the best performance, while Opus-MT exhibits the lowest mean BLEU score (0.0197). Moreover, Flan-T5 exhibits the greatest variability, as shown by its standard deviation score (0.1995), while RoGPT2 has lower standard deviations, thus causing more stability for the results.
- ROUGE—Flan-T5 also shows the highest mean ROUGE score (0.2983), which shows better performance on the one hand and the highest variability on the other hand, with the highest standard deviation as well (0.2263). On the contrary, GPT-Neo-Ro shows a lower mean ROUGE score (0.1374) but proves higher stability in terms of standard deviation.

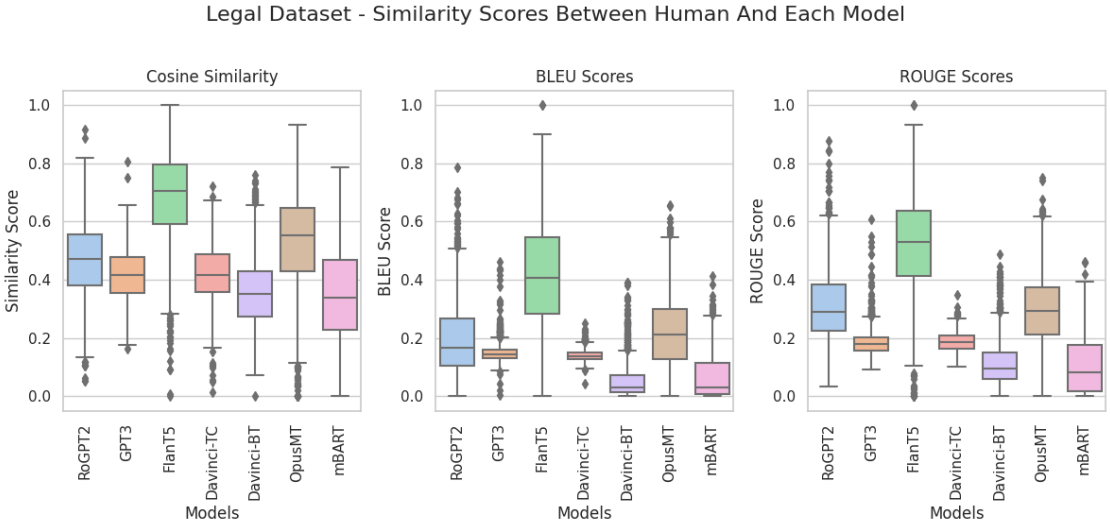


**Figure A6.** Text similarity metrics for Legal domain.

Discussion for legal dataset textual similarity results:

- Cosine similarity—among the models, Flan-T5 exhibits the highest mean for cosine similarity (0.6805), generating the most similar texts with the human set, nonetheless it produces the most variable scores as shown by the standard deviation (0.1607). On the contrary, the lowest mean is observed for davinci-003 (on backtranslation task) (0.3569), which suggests the lowest similarity with the human texts. A high standard deviation is noticed for mBART.
- BLEU—similarly, Flan-T5 exhibits the highest BLEU score (0.4057) and a moderate to high standard deviation (0.1948), indicating variability in its BLEU scores. The lowest mean BLEU score is observed for davinci-003 (on the backtranslation task) (0.0540), showing a moderate standard deviation.
- ROUGE—ROUGE scores also highlight Flan-T5 with the highest mean score (0.5191), indicating the highest similarity, and it has a moderate to high standard deviation (0.1705), which shows a high variability in its scores. GPT3 has a mean score of 0.1873 and a low standard deviation, which makes it more consistent, while mBART has the lowest mean score but a high variability in scores, shown by its high standard deviation.

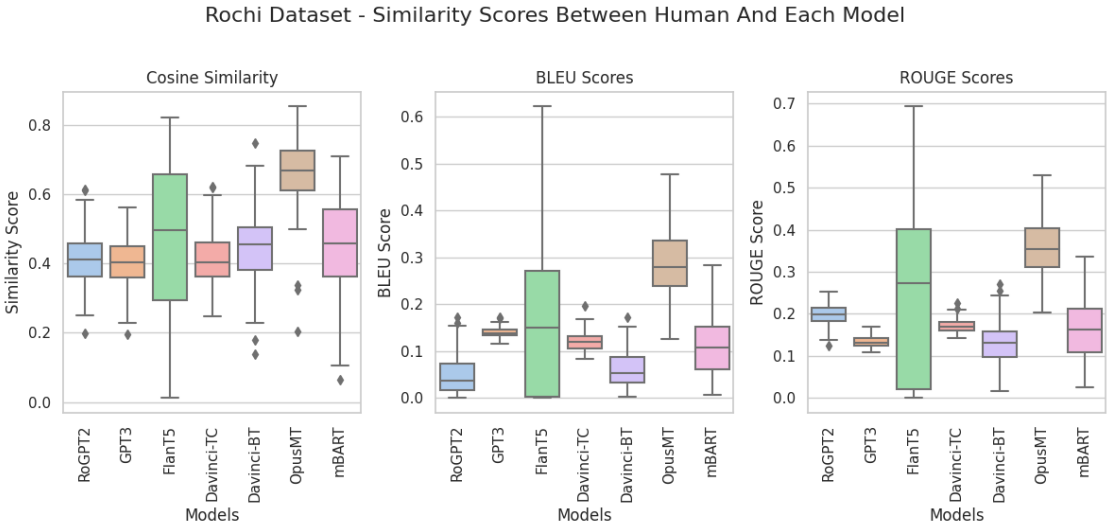


**Figure A7.** Text similarity metrics for RoCHI domain.

Discussion for scientific dataset (RoCHI) textual similarity results:

- Comparing the average (mean) cosine similarity scores among the models for the RoCHI dataset, the Opus-MT model has the highest mean score (see Table 5), indicating that on average, it produces text that is more similar to human-written text according to this metric. Flan-T5 also has a relatively high mean score (0,4719), while GPT-Neo-Ro has a slightly lower mean (0,4024), followed closely by RoGPT2 and davinci-003 (on text completion task). Intermediate mean scores are observed for davinci-003 (on backtranslation task) and mBART. Looking at the standard deviation scores (presented in Table 5), Flan-T5 has the highest standard deviation (0.2089), indicating a wide range of scores and potential inconsistency. Opus-MT has a relatively low standard deviation (0.1019), suggesting that its scores are more consistent across different text pairs, while other models have standard deviations that fall somewhere in between.
- BLEU scores demonstrate consistent results with cosine similarity scores, showing Opus-MT with the highest mean BLEU score (0.2812), indicating better performance according to BLEU. Flan-T5 follows closely with a mean score of 0.1652, GPT-NEo-Ro and davinci-003 (on text completion task) also have relatively high scores, while RoGPT2 and davinci-003 (on backtranslation task) have lower mean scores, mBART showing the lowest mean score (0.1136). The standard deviations for BLEU scores vary, with Flan-T5 having the highest variability (0.1658), which suggests inconsistency in its performance, while Opus-MT and RoGPT2 have relatively low standard deviations, indicating more stable performance.
- ROUGE scores expose Opus-MT with the highest score (0.3533), indicating better performance, followed by Flan-T5 and RoGPT2 with relatively high similarity. Davinci-003 and GPT-NEo-Ro have intermediate mean scores, meaning a moderate similarity, while mBART shows the lowest mean score (0.1624), signifying the least similarity with the human texts. In terms of variability, Flan-T5 has the highest standard deviation (0.2038), suggesting its ROUGE scores vary widely, while Opus-MT has a low standard deviation, indicating more consistent performance.

## Appendix D. Pairwise Wilcoxon Test

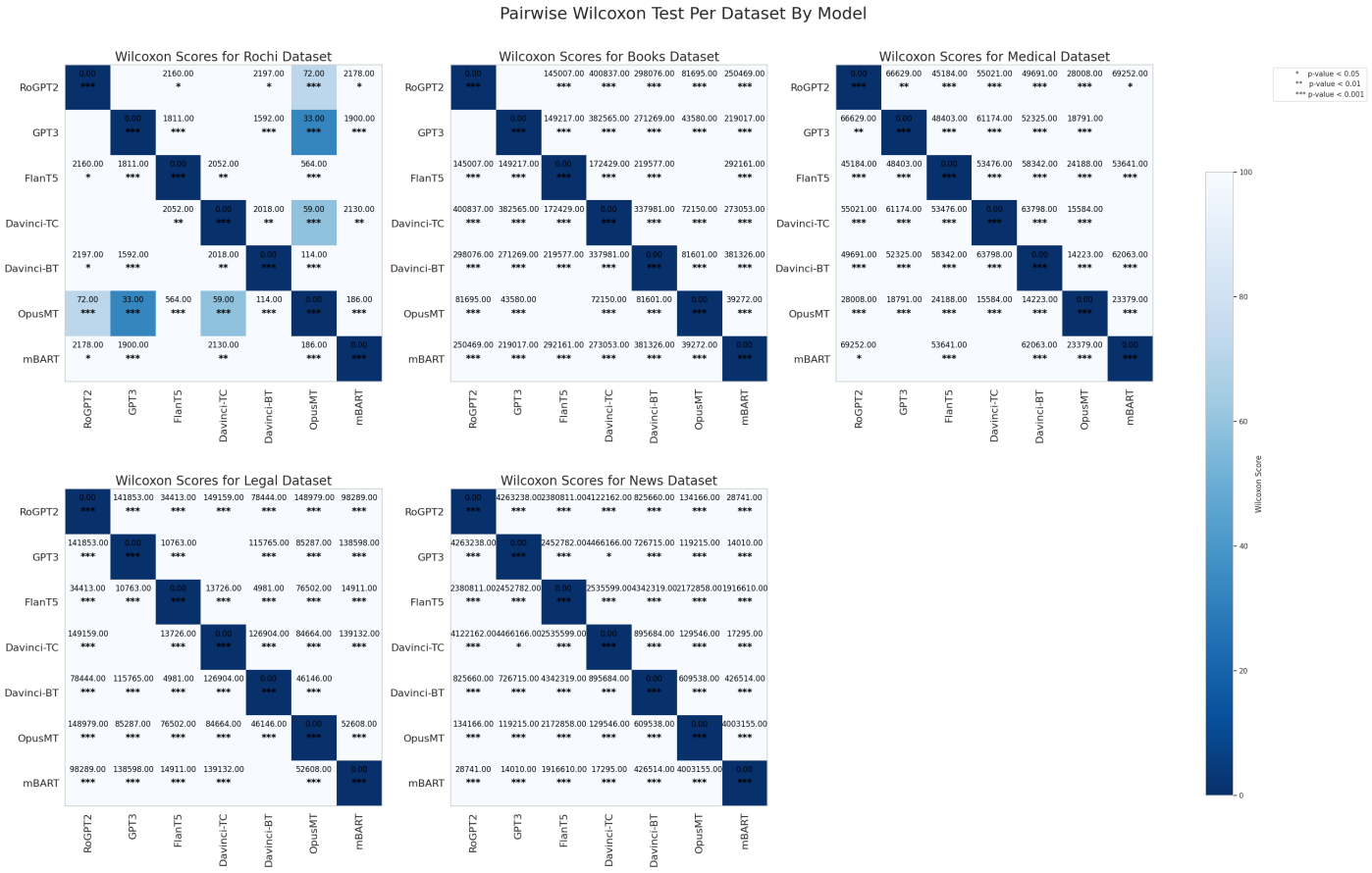


**Figure A8.** Pairwise Wilcoxon test by model per domain.

## Appendix E. Top ReaderBench Textual Complexity Indices

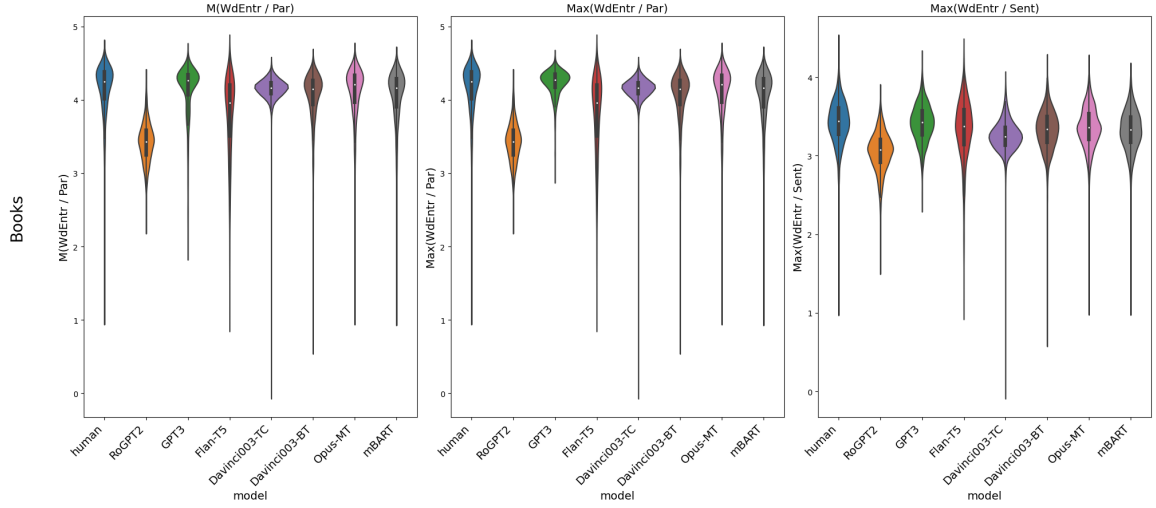


**Figure A9.** Top three RBIs for books domain.

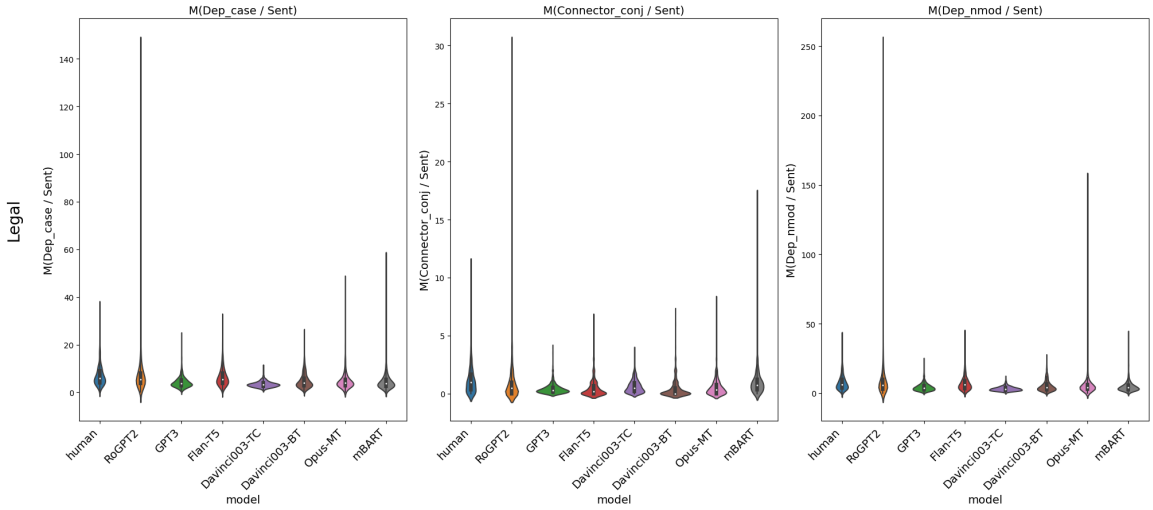


**Figure A10.** Top three RBIs for legal domain.

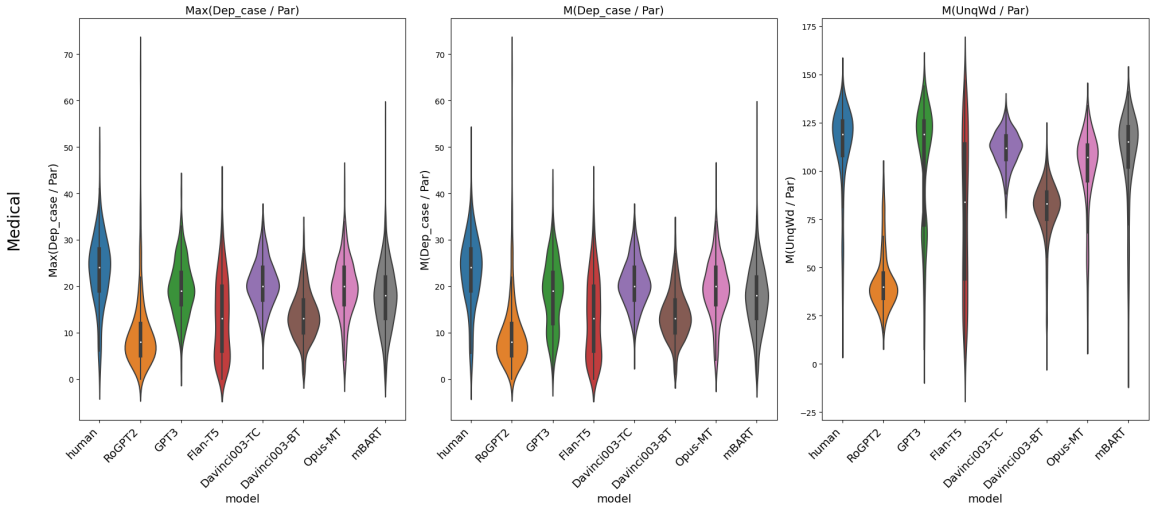


**Figure A11.** Top three RBIs for medical domain.

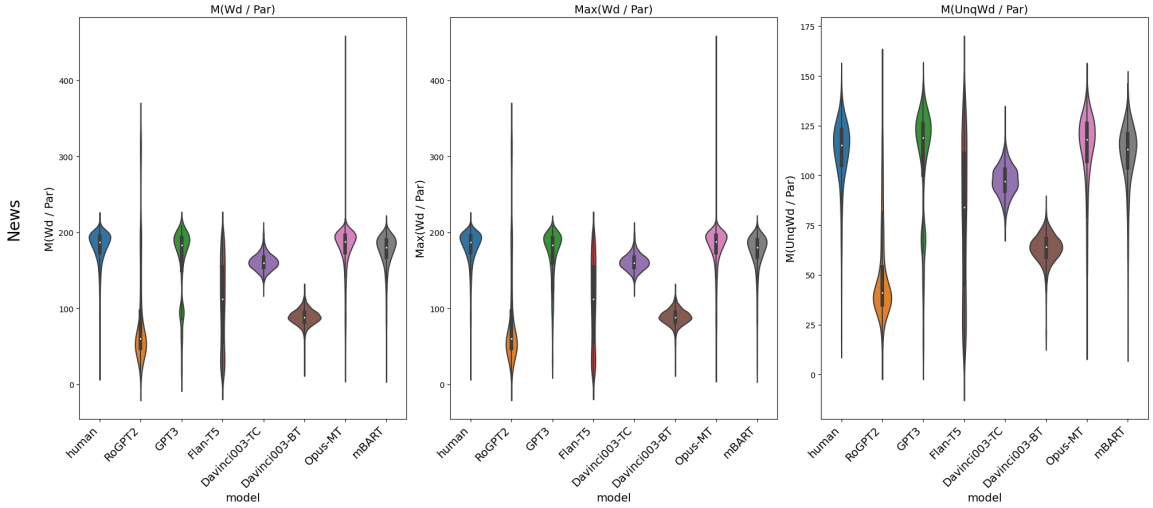


**Figure A12.** Top 3 RBIs for news domain.

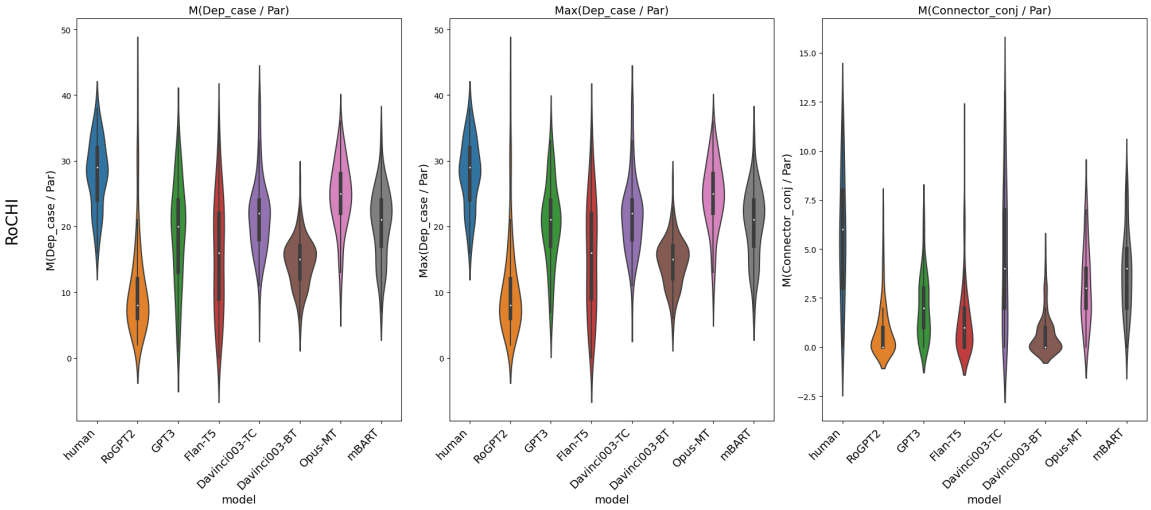


**Figure A13.** Top three RBIs for RoCHI domain.

## References


1. Weidinger, L.; Mellor, J.F.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. Ethical and social risks of harm from Language Models. *arXiv* **2021**, arXiv:2112.04359.
2. Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J.W.; Kreps, S.; et al. Release Strategies and the Social Impacts of Language Models. *arXiv* **2019**, arXiv:1908.09203.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates: San Jose, CA, USA, 2017; pp. 5998–6008.
4. Rush, A.M.; Chopra, S.; Weston, J. A Neural Attention Model for Abstractive Sentence Summarization. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015.
5. Serban, I.; Sordoni, A.; Bengio, Y.; Courville, A.C.; Pineau, J. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
6. Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.Y.; Edunov, S.; Chen, D.; Yih, W. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv* **2020**, arXiv:2004.04906.
7. Pandya, H.A.; Bhatt, B.S. Question Answering Survey: Directions, Challenges, Datasets, Evaluation Matrices. *arXiv* **2021**, arXiv:2112.03572.
8. Wang, Z. Modern Question Answering Datasets and Benchmarks: A Survey. *arXiv* **2022**, arXiv:2206.15030.
9. Chowdhury, T.; Rahimi, R.; Allan, J. Rank-LIME: Local Model-Agnostic Feature Attribution for Learning to Rank. In Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, Taipei, Taiwan, 23–27 July 2023.
10. Zheng, H.; Zhang, X.; Chi, Z.; Huang, H.; Yan, T.; Lan, T.; Wei, W.; Mao, X. Cross-Lingual Phrase Retrieval. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022.
11. Chakraborty, S.; Bedi, A.S.; Zhu, S.; An,B.; Manocha, D.; Huang, F. On the Possibilities of AI-Generated Text Detection. *arXiv* **2023**, arXiv:2304.04736.
12. Clark, E.; August, T.; Serrano, S.; Haduong, N.; Gururangan, S.; Smith, N.A. All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 1–6 August 2021.
13. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. *Preprint*, 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 6 November 2023).
14. Lamb, A. A Brief Introduction to Generative Models. *arXiv* **2021**, arXiv:2103.00265.
15. Bender, E.M.; Koller, A. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5185–5198.
16. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.
17. Michel-Villarreal, R.; Vilalta-Perdomo, E.; Salinas-Navarro, D.E.; Thierry-Aguilera, R.; Gerardou, F.S. Challenges and Opportunities of Generative AI for Higher Education as Explained by ChatGPT. *Educ. Sci.* **2023**, *13*, 856. [CrossRef]
18. Farrelly, T.; Baker, N. Generative Artificial Intelligence: Implications and Considerations for Higher Education Practice. *Educ. Sci.* **2023**, *13*, 1109. [CrossRef]
19. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
20. Christiano, P.; Leike, J.; Brown, T.B.; Martic, M.; Legg, S.; Amodei, D. Deep reinforcement learning from human preferences. *arXiv* **2023**, arXiv:1706.03741.

21. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1532–4435.
22. Kudo, T.; Richardson, J. SentencePiece: A simple and language-independent subword tokenizer and detokenizer for Neural Text Processing. *arXiv* **2018**, arXiv:1808.06226.
23. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. *arXiv* **2016**, arXiv:1508.07909.
24. Wei, J.; Bosma, M.; Zhao, V.Y.; Guu, K.; Yu, A.W.; Lester, B.; Du, N.; Dai, A.M.; Le Q.V. Finetuned Language Models Are Zero-Shot Learners. *arXiv* **2022**, arXiv:2109.01652.
25. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.
26. Williams, A.; Nangia, N.; Bowman, S.R. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *arXiv* **2018**, arXiv:1704.05426.
27. Hermann, K.M.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching Machines to Read and Comprehend. *arXiv* **2015**, arXiv:1506.03340.
28. Narayan, S.; Cohen, S.B.; Lapata, M. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. *arXiv* **2018**, arXiv:1808.08745.
29. Niculescu, M.A.; Ruseti, S.; Dascalu, M. RoGPT2: Romanian GPT2 for text generation. In Proceedings of the 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA, 1–3 November 2021; pp. 1154–1161.
30. Dumitrescu, S.; Rebeja, P.; Lorincz, B.; Gaman, M.; Avram, A.; Ilie, M.; Pruteanu, A.; Stan, A.; Rosia, L.; Iacobescu, C.; et al. LiRo: Benchmark and leaderboard for Romanian language tasks. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, Online, 6–14 December 2021.
31. Buzea, M.; Trausan-Matu, S.; Rebedea, T. Automatic Romanian Text Generation using GPT-2. *U.P.B. Sci. Bull. Ser. C Electr. Eng. Comput. Sci.* **2022**, *84*, 2286–3540.
32. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the 8th International Conference on Learning Representations, ICLR, Addis Ababa, Ethiopia, 26–30 April 2020.
33. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
34. Lin, C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
35. Niculescu, M.A.; Ruseti, S.; Dascalu, M. RoSummary: Control Tokens for Romanian News Summarization. *Algorithms* **2022**, *15*, 472. [CrossRef]
36. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. *arXiv* **2022**, arXiv:2210.11416.
37. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *arXiv* **2022**, arXiv:2203.02155.
38. Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual Denoising Pre-training for Neural Machine Translation. *arXiv* **2020**, arXiv:2001.08210.
39. Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B.; Huck, M.; Jimeno, Y.A.; Koehn, P.; Logacheva, V.; Monz, C.; et al. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*; Association for Computational Linguistics: Florence, Italy, 2016; pp. 131–198.
40. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 483–498.
41. Tiedemann, J.; Thottingal, S. OPUS-MT—Building open translation services for the World. In Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT), Lisbon, Portugal, 4–6 May 2020.
42. Junczys-Dowmunt, M.; Grundkiewicz, R.; Dwojak, T.; Hoang, H.; Heafield, K.; Neckermann, T.; Seide, F.; Germann, U.; Aji, A.; Bogoychev, N.; et al. Marian: Fast Neural Machine Translation in C++. In Proceedings of the ACL 2018, System Demonstrations, Melbourne, Australia, 15–20 July 2018; pp. 116–121.
43. Lavergne, T.; Urvoy, T.; Yvon, F. Detecting Fake Content with Relative Entropy Scoring. In Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse, Patras, Greece, 22 July 2008; Volume 377, pp. 27–31.
44. Gehrmann, S.; Strobelt, H.; Rush, A. GLTR: Statistical Detection and Visualization of Generated Text. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Florence, Italy, 28 July–2 August 2019; pp. 111–116.
45. Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C.D.; Finn, C. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. *arXiv* **2023**, arXiv:2301.11305.
46. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
47. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 11145–11159. [CrossRef]

48. Tian, E.; Cui, A. GPTZero: Towards Detection of AI-Generated Text Using Zero-Shot and Supervised Methods. Available online: https://gptzero.me (accessed on 8 January 2024).
49. Lee, N.; Bang, Y.; Madotto, A.; Khabsa, M.; Fung, P. Towards Few-Shot Fact-Checking via Perplexity. *arXiv* **2021**, arXiv:2103.09535.
50. Kleinberg, J. Bursty and Hierarchical Structure in Streams. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 91–101.
51. Habibzadeh, F. GPTZero Performance in Identifying Artificial Intelligence-Generated Medical Texts: A Preliminary Study. *J. Korean Med. Sci.* **2023**, *38*, e319. [CrossRef] [PubMed]
52. Verma, H.K.; Singh, A.N.; Kumar, R. Robustness of the Digital Image Watermarking Techniques against Brightness and Rotation Attack. *arXiv* **2009**, arXiv:0909.3554.
53. Langelaar, G.C.; Setyawan, I.; Lagendijk, R.L. Watermarking digital image and video data. A state-of-the-art overview. *IEEE Signal Process. Mag.* **2000**, *17*, 20–46. [CrossRef]
54. Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; Goldstein, T. A Watermark for Large Language Models. *arXiv* **2023**, arXiv:2301.10226.
55. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models Are Unsupervised Multitask Learners. 2019. Available online: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (accessed on 10 November 2023).
56. Shijaku, R.; Canhasi, E. *ChatGPT Generated Text Detection*; Technical Report; Unpublished; 2023. Available online: https://www.researchgate.net/profile/Ercan-Canhasi/publication/366898047_ChatGPT_Generated_Text_Detection/links/63b76718097c7832ca932473/ChatGPT-Generated-Text-Detection.pdf (accessed on 8 January 2024). [CrossRef]
57. Krishna, K.; Song, Y.; Karpinska, M.; Wieting, J.; Iyyer, M. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *arXiv* **2023**, arXiv:2303.13408.
58. Masala, M.; Ruseti, S.; Dascaku, M. RoBERT – A Romanian BERT Model. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), 8–13 December 2020; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 6626–6637.
59. Liu, J.-J.; Liu, J.-C. Permeability Predictions for Tight Sandstone Reservoir Using Explainable Machine Learning and Particle Swarm Optimization. *Geofluids*. **2022**, *2022*, 2263329. [CrossRef]
60. Chen, J.; Zhang, Y.; Hu, J. Synergistic effects of instruction and affect factors on high- and low-ability disparities in elementary students' reading literacy. *Read. Writ. J.* **2021**, *34*, 199–230. [CrossRef]
61. Vangala, S.R.; Bung, N.; Krishnan, S.R.; Roy, A. An interpretable machine learning model for selectivity of small-molecules against homologous protein family. *Future Med. Chem.* **2022**, *14*, 1441–1453. [CrossRef]
62. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 4765–4774.
63. Singhal, A. Modern Information Retrieval: A Brief Overview. *IEEE Data Eng. Bull.* **2001**, *24*, 35–43.