Advancing Authorship Attribution: A Phonetic Encoding and Fusion Approach

Arjun Ramesh Kaushik, Sunil Rufus, Nalini Ratha

University at Buffalo, The State University of New York, Buffalo, USA
{kaushik3, sunilruf, nratha}@buffalo.edu

***Abstract*—With the advent of Generative AI, machines are now able to replicate/generate multi-modal data. Deepfake videos, fake news, and fake audio have become increasingly common today. This brings us to the question: how do we distinguish genuine information/data from the fake ones? To this end, in this paper, we propose a combination of character encoding scheme and embedding fusion focused on discerning AI-generated text and human-authored text. Inspired by Chaos Game Representation (CGR), we propose an encoding scheme based on the phonetic sounds of each character. With either CGR or Phonetic Encoding scheme, we can represent text as a sequence of 0's and 1's. We reshape the linear representation as a 2D representation, before processing them as images. Additionally, we use the embedding fusion method to aid the authorship attribution classification task on texts of variable length. Through extensive experimentation, we show that using the Phonetic Encoding scheme along with embedding fusion achieves better results than CGR on the authorship attribution task on two publicly available datasets.**

***Index Terms*—Generative AI, Large Language Models, Trustworthy AI**

I. Introduction

In recent years, advancements in Generative AI have transformed the world, enhancing the diversity, control, and quality of data produced by these models. Large Language Models (LLMs) are at the forefront of this change. Notable examples such as OpenAI's ChatGPT, Google's Gemini, and Meta's Llama showcase exceptional performance across various tasks, including answering questions, composing emails, essays, and code snippets. However, while these advancements usher in a new era of human-like text generation, they also raise significant concerns regarding the detection and mitigation of potential misuse of LLMs.

LLMs have infiltrated every corner of society with students, developers, researchers, and reporters, all turning towards them [1] [2] [3]. Instances have arisen where educational institutions have banned ChatGPT due to fears of its potential to facilitate cheating in assignments [1] [2]. Additionally, media reports have drawn attention to the proliferation of fake news generated by LLMs [3]. Of late, there have been growing concerns that researchers are frequently using LLMs to write research papers. These concerns have understandably cast a shadow over the application of LLMs, particularly in critical domains like media and education.

Technology to discern LLM-generated texts is the need of the hour to mitigate potential consequences associated with the misuse of LLMs. With the ability to distinguish between human-authored and LLM-generated text, there is transparency in an author's work. In addition, it can greatly enhance trust in LLMs and encourage wider adoption. Similarly, effective text detection mechanisms can assist developers and researchers in tracking generated texts and preventing unauthorized usage [1] [2] [3].

Given the critical significance of accurate LLM-generated text detection, we propose a novel encoding technique to represent text as images through a Phonetic Encoding scheme. In each case, the text is converted to a numerical form through a unique representation mapping and split into chunks of size 32 x 32. We follow the CGR paper [4] and stick with the recommended size of 32 x 32, as shown in Fig. 2. The chunks are accumulated as channels of the image representing the text. Binary classification is performed as shown in Fig. 1. In addition to Phonetic Encoding, we also utilize embedding fusion during testing, making our approach feasible on texts of variable length. Features extracted from the channels of the image (representative of the text) are averaged before performing binary classification using a simple neural network. Results show that Phonetic Encoding and embedding fusion, together, yield a superior performance than CGR.

The rest of the paper is structured in the following way. Section II provides a basis to help understand CGR and Phonetic Encoding schemes. We dive into the past and recent research contributions in solving the authorship attribution task in Section III. A detailed overview of the dataset is shown in Section V. Sections VI and VII provide methodology and results in support of our approach.

II. Background details

A. Chaos Game Representation (CGR)

The concept of Chaos Game Representation (CGR) was introduced by Jeffrey in 1990 and further elaborated upon in 1992 [5]. This technique offers a visualization method for the structural analysis of DNA sequences. It begins with a square with corners labeled A, C, G, and T, and the starting point is the center of this square. Subsequently, each nucleotide in the sequence is represented by plotting a point equidistant between the current nucleotide's corner and the starting point, thereby forming a sequence of points. When the resulting image is formatted as a square with dimensions of $2^k \times 2^k$ pixels, it's been shown that each pixel represents a unique k-mer.

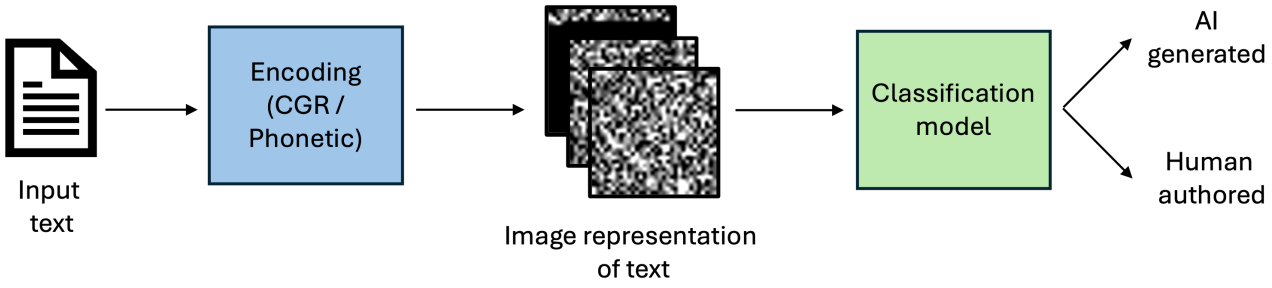

Fig. 1. An overview of the proposed authorship attribution framework.

The grayscale intensity of each pixel reflects the frequency of occurrence of its corresponding k-mer within the DNA sequence relative to the total count of k-mers. Different species' DNA sequences produce distinctive patterns in CGR images, ranging from simple shapes like triangles or rectangles to more intricate fractal structures [6].

An alternative method known as Frequency Chaos Game Representation (FCGR) [7] [8], offers a modification of the CGR approach. FCGR retains equivalence with CGR when pixelation levels are consistent, but it offers improved computational processing ease. In FCGR, a k-th order representation of a sequence is achieved through a $2^k \times 2^k$ matrix, where each element represents the number of points located within the corresponding grid square.

$\text{FCGR}_{k+1}(s)$ is defined by updating each element in the previous $\text{FCGR}_k(s)(s)$ with specific values. In FCGR images, words with higher frequencies are depicted with darker pixels, reflecting their higher intensity, making it easier to identify patterns and significant features within the DNA sequence [7].

B. Phonetic Encoding

Name Matching [9] is a commonly employed technique to check if multiple name strings refer to the same entity. It employs various techniques such as Phonetic Encoding and Pattern Matching. Phonetic Encoding aims to represent names based on their pronunciation, grouping names with similar sounds. On the other hand, Pattern Matching focuses on comparing the arrangement of characters in names, typically used in fuzzy string matching. Another method for Name Matching is the lexicographic technique, which considers all possible orthographic variations, alternative forms of actual names, and initials to identify matches.

In this paper, we employ Phonetic Encoding, which converts names into codes based on their pronunciation. Despite differences in spelling, names with similar sounds are assigned the same code. This approach is effective for handling spelling variations in names. Soundex is a commonly used algorithm for this purpose, with later algorithms like DoubleMetaphone, Phonex, Phonix, and NYSIIS building upon its principles [10] [11]. The English language can be categorized into 44 Phonemes, Underhill 2008. We merge the 44 categories based on overlap and reduce the categorization to 10. The non-alphabets and punctuations are categorized into 2 separate categories, thus, making it a total of 12 as shown in Table II.

III. Related Work

The concept of authorship attribution using BERT embeddings has been effectively demonstrated by many researchers. PART [12] and BertAA [13] shows how BERT embeddings can be used to grasp authors' writing styles and generate stylometric representations. [14] explores Siamese Networks for authorship attribution (AA), comparing their effectiveness with BERT fine-tuning.

Previous works such as [15] [4] [16] use traditional machine learning algorithms for classification on datasets of limited scope - comparison between human-authors or consider just a single LLM. [17] discusses the application of Convolutional Neural Networks (CNNs) in character-level signal processing, serving as a motivation for our work.

The studies [18] and [19] explore the detection and regulation of AI-generated text, particularly in the context of scientific writing. In [18], a dual approach is introduced, employing feature-based methods to categorize aspects like Writing Style and Coherence, alongside neural network-based fine-tuning of a GPT-2 output detector model using RoBERTa, achieving a 94.6% F1 score. Conversely, [19] provides a broader overview of detection techniques, encompassing black-box methods relying on API-level access and deep learning approaches involving fine-tuning LLMs like RoBERTa. It also discusses white box methods, including post-hoc rule-based and neural-based approaches, as well as inference-time watermarking techniques for modifying word selection during text generation. Together, these studies contribute to understanding and regulating AI-generated text in scientific writing, offering insights into both specific detection methodologies and broader frameworks for control and regulation.

IV. Dataset

In this research, we consider two datasets - Human vs LLM text [20] and the dataset from Authorship Attribution for Neural Text Generation (AANTG) [21]. The AANTG dataset is balanced as shown in Table IV, whereas the Human vs LLM dataset is highly imbalanced. **Hence, we balance the dataset by randomly sampling n texts from human-authored texts,**

TABLE I
ENCODING TECHNIQUE ADAPTED FROM CGR AS BASE4 REPRESENTATION

Characters	Base4 Representation	Binary Representation
'h', 'j', 'g'	00	0000
'i', 'y'	01	0001
't'	02	0010
'm'	03	0011
'l', 'r'	10	0100
'a'	11	0101
' '	12	0110
's'	13	0111
'e'	20	1000
'n'	21	1001
'0','1','2',.....'*', '/', etc	22	1010
'u'	23	1011
'b', 'd', 'p'	30	1100
'f', 'v', 'w'	31	1101
'c', 'k', 'q', 'x', 'z'	32	1110
'o'	33	1111

TABLE II
OUR PROPOSED PHONETIC ENCODING TECHNIQUE ADAPTED FROM NAME MATCHING

Characters	Decimal Representation	Binary Representation
'b'	1	0001
'd'	2	0010
'f', 'v'	3	0011
'g', 'j', 'h', 'w', 'u', 'o', 'y', 'i', 'a', 'e'	4	0100
'k', 'q', 'x', 's', 'c', 'z'	5	0101
'l'	6	0110
'm', 'n'	7	0111
'p'	8	1000
'r'	9	1001
't'	10	1010
' '	11	1011
'0','1','2',.....'*', '/', etc	12	1100

where n is equal to the number of LLM-generated (GPT / Llama / Flan / Mistral / OPT) texts in consideration.

The Human vs LLM text dataset is a compilation of texts generated from 63 different LLMs. From the 63 LLMs, we pick texts generated from 4 different LLMs for our experiments. We balance the dataset before performing any experiment. The distribution of the dataset is shown in Table III.

V. PROPOSED METHODOLOGY

A. CGR encoding

The Chaos Game Representation (CGR) technique [4] categorizes all alphanumeric characters into 16 distinct groups. These groups are indexed in base4, as presented in Table I. Representing text as an image using CGR encoding can be achieved in 4 simple steps -

1) Substitute the characters with their numerical representation from Table I
2) Substitute the numbers with their 2-bit binary representation, i.e., 'a' is represented as '0101'.
3) Split the binary representation into equal chunks of size 2^k, and pad with '0' if necessary.
4) Now, we have a 2^k x 2^k image of n channels, where n varies based on the length of input text

The characters of the text are split into chunks of equal size with padding, if necessary. There is no restriction on the chunk size, we have chosen 1024 to facilitate the generation of 32 x 32 images. The number of channels for each image will vary depending on the length of the text and the size of the chunk.

B. Phonetic Encoding

Our proposal, Phonetic Encoding categorizes all alphanumeric characters into 12 distinct groups, with decimal representations for each group as in Table II. Representing text as an image using Phonetic Encoding can be achieved in 4 simple steps -

1) Substitute the characters with their numerical representation from Table II
2) Substitute the numbers with their 4-bit binary representation, i.e., 'a' is represented as '0100'.
3) Split the binary representation into equal chunks of size 2^k, and pad with '0' if necessary.
4) We now have a 2^k x 2^k image of n channels, where n varies based on the length of input text

Like CGR encoding, the characters of the text are split into chunks of equal size with padding, if necessary. The channel count for each image will vary depending on the input text length and chunk size.

TABLE III
DISTRIBUTION OF KAGGLE'S HUMAN VS LLM DATASET.

Source	No. of Samples	Min. word count per sample	Max. word count per sample
Human	347,692	25	71,543
Flan	45,608	25	905
GPT	75,599	25	3,565
Llama	42,623	25	1770
OPT	80151	25	1044

TABLE IV
DISTRIBUTION OF AANTG DATASET.

Source	No. of Samples
Human	1066
CTRL	1066
FAIR	1066
GPT	1066
GPT2	1066
GPT3	1066
GROVER	1066
InstructGPT	1066
PPLM	1066
XLM	1066
XLNet	1066

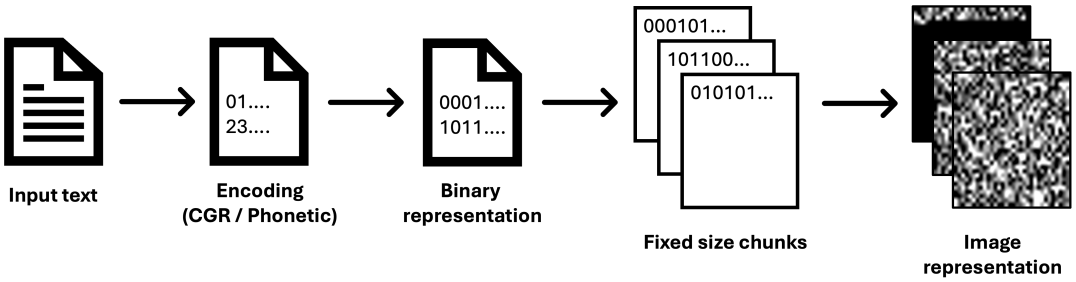

Fig. 2. Steps to achieve image representation of a given text using the encoding schemes - CGR and Phonetic.

C. Embedding Fusion

With texts of varying lengths, their encoding lengths will vary as well. To address this we take an innovative approach - embedding fusion. We perform the following steps -

- We reshape an encoding of size n into 32 x 32 2D arrays, yielding a shape of (m, 32, 32) for each encoding, where m can be considered as the number of channels.
- We train a classification model with 1-channel inputs, i.e., each embedding of shape (m, 32, 32) will be treated as m inputs of shape (1, 32, 32) carrying the same label as shown in Fig. 3.
- During testing as shown in Fig. 4, we average the feature vector (embedding) obtained from each of these m channels and perform the binary classification using a simple multilayer perceptron.

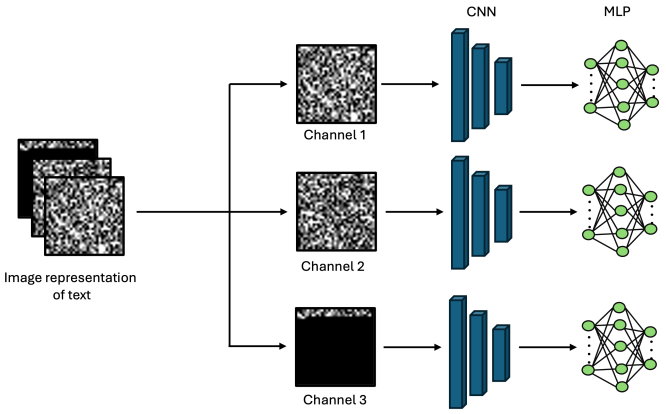

Fig. 3. During training, each channel of the image (representation of text) is treated as an independent input.

VI. RESULTS

From Tables V and VI, we can see that the Phonetic Encoding scheme outperforms the CGR encoding scheme. Since our problem statement corresponds to a binary classification task, we provide True Positive Rate (TPR), False Positive Rate (FPR), and Matthews Correlation Coefficient (MCC) as additional metrics to better understand the performance of our system. A higher TPR and lower FPR are the desired metrics for our system. MCC quantifies the correlation between actual and predicted binary classifications, with values falling within

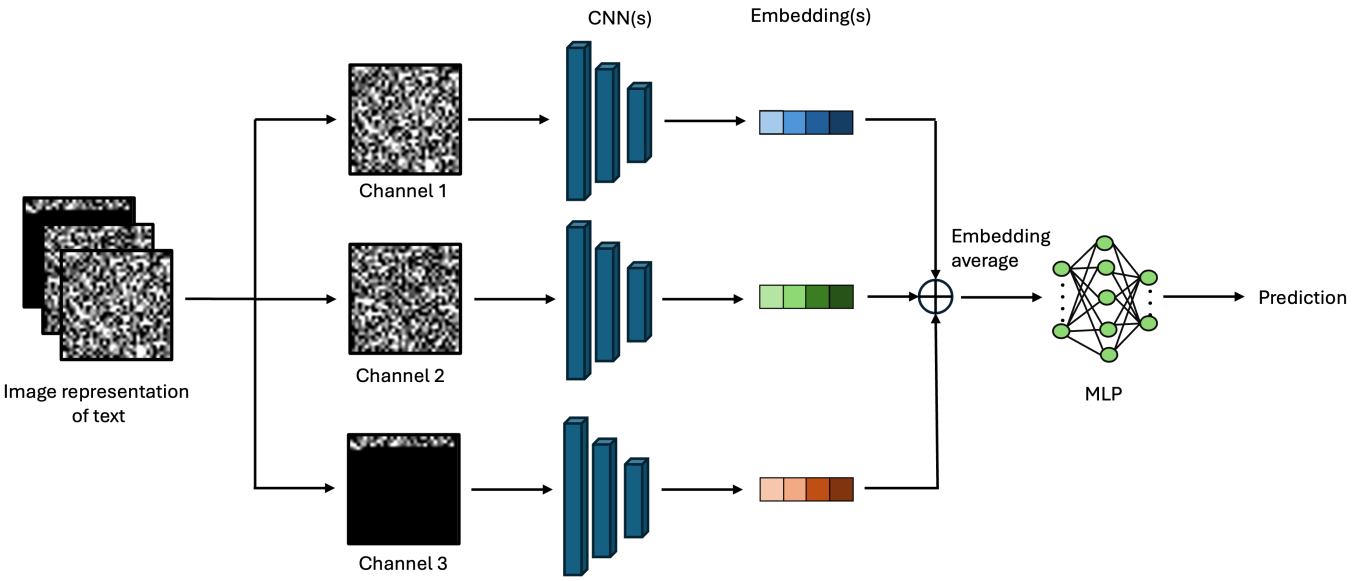

Fig. 4. During testing, we extract the feature vector (embedding) for each channel of the image (representation of text) and take their mean. Using the averaged embedding, we perform prediction.

the range of -1 to $+1$. Positive values highlight high correlation and negative values indicate low correlation between predictions and actual labels. For binary classification tasks, MCC serves as a better metric compared to accuracy.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

Where,

- TP - True Positives
- TN - True Negatives
- FP - False Positives
- FN - False Negatives

VII. Conclusion and Future Work

This paper introduces a novel encoding scheme aimed at distinguishing AI-generated text from human-authored text. Inspired by the concept of CGR, we propose a character-level encoding scheme called Phonetic Encoding. Our methodology involves transforming the encoded text into 2D arrays, treating them as images, thereby framing the task as a binary classification problem on images. Furthermore, we incorporate embedding fusion, enabling our approach to handle texts of variable lengths. Through extensive experimentation across two datasets, we demonstrate that the performance of Phonetic Encoding, combined with embedding fusion, surpasses that of CGR. We envision that our framework could be enhanced by incorporating temporally-aware embedding fusion techniques and further improve its capability to handle texts of any length.

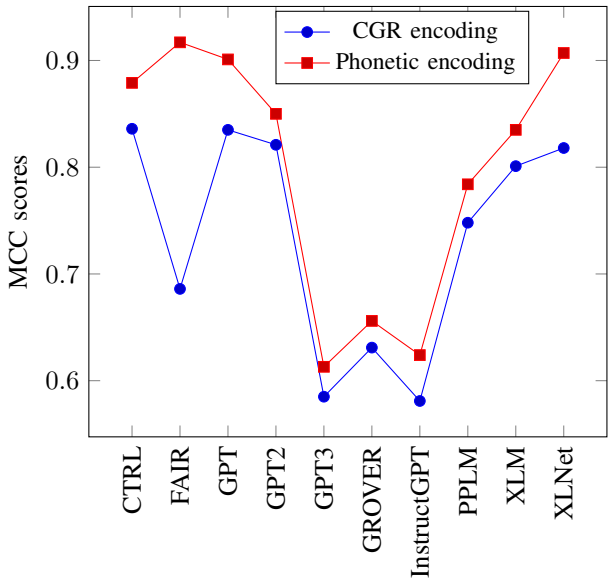

Fig. 5. Comparison of Matthews Correlation Coefficient (MCC) scores using CGR and Phonetic encoding on the AANTG dataset.

References

[1] Matt Dible. Schools ban chatgpt amid fears of artificial intelligence-assisted cheating. *Voice of America*, 2023.

[2] Maya Yang. New york city schools ban ai chatbot that writes essays and answers prompts. *The Guardian*, 2023.

[3] Pranshu Verma. The rise of ai fake news is creating a 'misinformation superspreader'. *The Washington Post*, 2023.

[4] Daniel Lichtblau and Catalin Stoean. Authorship attribution using the chaos game representation. 02 2018.

[5] H.Joel Jeffrey. Chaos game visualization of sequences. *Computers & Graphics*, 16(1):25–33, 1992.

[6] Rallis Karamichalis. Additive methods for genomic signatures. *BMC Bioinformatics*, 17:313.

[7] P Deschavanne, Alain Giron, J Vilain, Guillaume Fagot, and B Fertil. Deschavanne pj, giron a, vilain j, fagot g, fertil b.. genomic signature: characterization and classification of species assessed by chaos game

TABLE V
AUTHORSHIP ATTRIBUTION METRICS FOR THE AANTG DATASET.

LLM	Methodology	Encoding	Accuracy (%)	TPR	FPR	MCC
Flan	Embedding Average	CGR	94.54	0.809	0.038	0.735
		Phonetic	94.34	0.813	0.034	0.776
GPT	Embedding Average	CGR	76.50	0.869	0.322	0.539
		Phonetic	79.26	0.872	0.288	0.580
Llama	Embedding Average	CGR	80.37	0.499	0.044	0.540
		Phonetic	78.88	0.866	0.25	0.588
OPT	Embedding Average	CGR	82.03	0.304	0.014	0.445
		Phonetic	83.55	0.879	0.185	0.659
Mistral	Embedding Average	CGR	83.38	0.998	0.305	0.713
		Phonetic	86.80	0.998	0.282	0.757

TABLE VI
AUTHORSHIP ATTRIBUTION METRICS FOR THE KAGGLE HUMAN VS LLM DATASET.

LLM	Methodology	Encoding	Accuracy (%)	TPR	FPR	MCC
CTRL	Embedding Average	CGR	91.33	0.985	0.155	0.836
		Phonetic	94.91	0.990	0.082	0.879
FAIR	Embedding Average	CGR	83.61	0.719	0.055	0.686
		Phonetic	95.78	0.986	0.068	0.917
GPT	Embedding Average	CGR	91.33	0.985	0.158	0.835
		Phonetic	95.01	0.929	0.029	0.901
GPT2	Embedding Average	CGR	91.10	0.934	0.115	0.821
		Phonetic	92.27	0.987	0.150	0.850
GPT3	Embedding Average	CGR	79.16	0.844	0.264	0.585
		Phonetic	80.32	0.875	0.268	0.613
GROVER	Embedding Average	CGR	80.93	0.913	0.298	0.631
		Phonetic	82.11	0.921	0.276	0.656
InstructGPT	Embedding Average	CGR	78.22	0.879	0.302	0.581
		Phonetic	79.39	0.945	0.338	0.624
PPLM	Embedding Average	CGR	87.35	0.918	0.175	0.748
		Phonetic	89.23	0.891	0.107	0.784
XLM	Embedding Average	CGR	89.93	0.949	0.153	0.801
		Phonetic	91.33	0.986	0.163	0.835
XLNet	Embedding Average	CGR	90.87	0.925	0.106	0.818
		Phonetic	95.31	0.975	0.066	0.907

representation of sequences. mol biol evol 16: 1391-1399. *Molecular biology and evolution*, 16:1391–9, 11 1999.

[8] Yingwei Wang, Kathleen Hill, Shiva Singh, and Lila Kari. The spectrum of genomic signatures: From dinucleotides to chaos game representation. *Gene*, 346:173–85, 03 2005.

[9] Ankita Pilania and G. Mayil Muthu Kumaran. Comparative study of name matching algorithms. In *6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1174–1178, 2019.

[10] B. Lisbach and V. Meyer. *Linguistic Identity Matching*. SpringerLink : Bücher. Springer Fachmedien Wiesbaden, 2013.

[11] Peter Christen. A comparison of personal name matching: Techniques and practical issues. In *Sixth IEEE International Conference on Data Mining - Workshops*, pages 290–294, 2006.

[12] Javier Huertas-Tato, Alvaro Huertas-Garcia, Alejandro Martin, and David Camacho. Part: Pre-trained authorship representation transformer, 2022.

[13] Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. BertAA : BERT fine-tuning for authorship attribution. In Pushpak Bhattacharyya, Dipti Misra Sharma, and Rajeev Sangal, editors, *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, Indian Institute of Technology Patna, Patna, India, December 2020. NLP Association of India (NLPAI).

[14] Chakaveh Saedi and Mark Dras. Siamese networks for large-scale author identification, 2021.

[15] Niful Islam, Debopom Sutradhar, Humaira Noor, Jarin Tasnim Raya, Monowara Tabassum Maisha, and Dewan Md Farid. Distinguishing human generated text from chatgpt generated text using machine learning, 2023.

[16] Hosam Alamleh, Ali Abdullah S. AlQahtani, and AbdElRahman ElSaid. Distinguishing human-written and chatgpt-generated text using machine learning. In *Systems and Information Engineering Design Symposium (SIEDS)*, pages 154–158, 2023.

[17] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution, 2016.

[18] Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. Ai vs. human – differentiation analysis of scientific content generation, 2023.

[19] Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. The science of detecting llm-generated texts, 2023.

[20] Zachary Grinberg. Human vs. llm text corpus, 2024.

[21] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, 2020.