Detecting, Generating, and Evaluating in the Writing Style of Different Authors

Mosab Rezaei
Department of Computer Science
Northern Illinois University

Abstract

In recent years, stylometry has been investigated in many different fields. Hence, in this work, we are going to tackle this problem, detecting, generating, and evaluating textual documents according to the writing style by leveraging state-of-the-art models. In the first step, the sentences will be extracted from several different books, each belonging to a different author, to create a dataset. Then the selected models will be trained to detect the author of sentences in the dataset. After that, generator models are utilized to generate sentences based on the authors' writing styles with unpaired samples in the dataset. Finally, to evaluate the performance of the generators, the previously trained models will be used to assess the generated sentences and to compare the distribution of various syntactic features between the original and generated sentences. We hope the result shows that models can be achieved to detect and generate textual documents for the given authors according to their writing style.

1 Introduction

Stylometry is a linguistic discipline that applies statistical analysis to literature based on the assumption that each author has an unconscious aspect to their style (Yang et al., 2008). Generally and simply, stylometry is a field of study that statistically analyzes authorship attribution (Holmes, 1998). As the production of digital documents increases, the importance of stylometry grows as well. The increasing attention to stylometry has been reflected in Wayman et al. (2009): "As non-handwritten communications become more prevalent, such as blogging, text messaging, and emails, there is a growing need to identify writers not by their written script, but by analysis of the typed content".

In this work, after demonstrating the existence of distinguishable patterns between different authors' writing styles, we aim to train generator models without paired samples to generate and then evaluate the generated sentences in different writing styles. Leveraging these models opens new advances in generating stylistic text, further enriching applications such as authorship verification, creative writing, forensic linguistics, legal systems, and criminology.

It is important to note that one of the key issues in this work lies in evaluating the generated sentences. First, while well-known metrics like accuracy or F1 score are valuable, they cannot adequately reflect how accurately the model detects and mimics writing styles. Moreover, these metrics do not provide clear insights into the performance of the model in each of the writing style categories. On the other hand, relying on expert human evaluations presents significant challenges. For example, gathering experts who specialize in all five authors' writing styles in the dataset is nearly impossible. Hence, we are going to use an AI-evaluate-AI technique to assess the generated sentences. We will train a detector capable of classifying sentences with high performance and use it to evaluate the generated sentences. Furthermore, we will incorporate feature-based evaluation by comparing the features extracted from both the original sentences and the generated sentences to measure their alignment.

Given the importance of stylometry, and the challenges mentioned above, this study has been focused on answering three main research questions in this area:

RQ1 (Detection): Given the differences in authors' writing styles, can the proposed model extract related features and accurately detect the authors for a given sentence?

RQ2 (Generation): Is it possible to train a generator to produce sentences in the writing style of a specific author without using paired training data?

RQ3 (Evaluation): Can detector models be used to evaluate generated sentences by generator models?

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), pages 485–491
April 30 - May 1, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Since the use of machine learning for analysis alone is well understood, and evaluation is part of our future work, in this section, we describe only the systems used for style generation.

In Logeswaran et al. (2018), the authors propose a novel generative model for sentence style transfer that modifies the style of a given sentence based on categorical attributes. The architecture comprises an RNN-based encoder-decoder that generates sentences consistent with the input's content and specified attributes.

de Rivero et al. (2021) address style transfer in NLP by fine-tuning GPT-2 on Grammarly's Yahoo Answers Formality Corpus (GYAFC) to convert informal text into formal text while preserving meaning. Their model generates multiple formal sentence options, achieving a formality score above 0.7 in 61.36% of cases and a content preservation score above 0.8 in 71.33% of cases, demonstrating effective style transformation.

Also, in Tian et al. (2018), the researchers propose a text style transfer model using an attentional auto-encoder and a binary style classifier, ensuring content preservation by minimizing the distance between the POS-tagged structure of input and output sentences. The approach focuses on maintaining noun consistency, incorporating a language model for fluency and a style classifier to guide the generator in producing sentences with the desired style.

For the text style transferring task, other researchers in Lai et al. (2019) propose a GAN-based framework for non-parallel text style transfer that integrates a seq-to-seq encoder-decoder with attention, word-level conditional mechanisms, and dual discriminators (CNN and RNN) to balance content preservation and style transformation.

Authors in Hu et al. (2017) propose a deep generative model that enhances Variational Autoencoders (VAEs) with structured latent variables and holistic discriminators to generate text with specified attributes while ensuring disentanglement. Their approach, which incorporates a wake-sleep algorithm for collaborative optimization, effectively learns interpretable latent representations from minimal supervision, enabling controlled text generation with potential applications in NLP and content creation.

In Du et al. (2020), researchers introduced Schema-Guided Natural Language Generation (SG-NLG), a task that generates natural language prompts based on rich schemata, repurposing a dataset from dialog state tracking to train Seq2Seq, CVAE, and GPT-2 models. Their findings show that leveraging schema information enhances semantic quality and diversity, with Seq2Seq and CVAE excelling in reference similarity and GPT-2 performing best in diversity and human evaluation.

The Stable Style Transformer (SST) presented in Lee (2020), introduces a model-agnostic text style transfer approach using the Delete and Generate framework, where a pre-trained classifier extracts attribute markers without relying on a dictionary or attention scores, and a Transformer-based encoder-decoder generates the transferred sentence while preserving content. This method, trained on non-parallel datasets, demonstrates robust performance in handling long dependencies and offers a stable, effective solution for text style transfer.

CTERM-GAN (Betti et al., 2020) addresses the common limitation of NLG models that focus solely on syntax by incorporating both syntactic and semantic aspects through a relational memory-based generator and dual discriminators. Experimental results show that it maintains or improves syntactic accuracy while significantly enhancing semantic coherence, demonstrating its potential for generating text conditioned on various inputs, including writing styles.

Authors in Li et al. (2022), developed Diffusion-LM, a non-autoregressive language model leveraging continuous diffusion processes for controllable text generation, where gradient-based manipulation of latent variables during denoising enables fine-grained style control, outperforming prior plug-and-play models and achieving competitive results against fine-tuned autoregressive baselines.

The paper by Lyu et al. (2023) explores the application of diffusion models for fine-grained text style transfer, demonstrating that their approach, trained solely on the StylePTB dataset without external resources, achieves state-of-the-art performance across 13 tasks, including compositional style transfers. Their results highlight the potential of diffusion-based models for controllable text generation in low-resource settings, while also suggesting future directions such as integrating pre-trained embeddings and exploring alternative architectures.

3 Procedure

Based on our research questions, we divided the proposal into three phases: detection, generation, and evaluation. The outline of the procedure has been shown in Figure 1, which illustrates each phase and the data flow using different colors: Blue for phase 1 (detection), green for phase 2 (generation), and purple for phase 3 (evaluation). Also, this shows that three different subsets of the dataset were extracted and flowed in different paths, one for the detection phase and two for the text generator model.

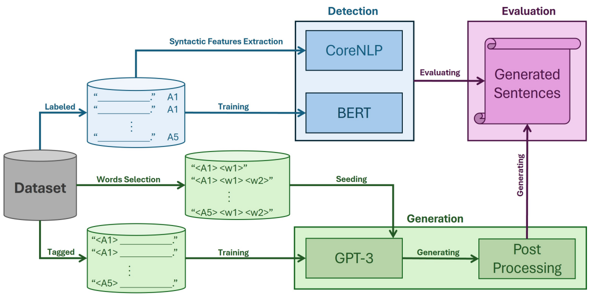

Figure 1: The outline of the procedure for detecting, generating, and evaluating text in various writing styles.

In order to create the dataset, we used Project Gutenberg to collect books by different authors. We picked five authors, Charles Dickens, Mark Twain, Herman Melville, Jane Austen, and Louisa May Alcott. They all belong to the 18th century and provide a good balance of male and female authors as well as British and American authors in our analysis. We also aim to cover a broad range of topics, as Twain and Melville generally wrote for men, whereas Austen and Alcott typically wrote about women. Dickens, meanwhile, mostly addressed social conditions, such as poverty and wealth, rather than focusing specifically on men or women. The total number of extracted sentences is 115,471. In datasets with paired samples, there exist at least two different styles for the same content, like formal and informal datasets. As mentioned before, in our dataset, there are no identified sentences from different authors expressing the same content, which makes it much more challenging for the model to understand the differences or, in the next steps, to transfer one sentence from an author to another author's writing style.

Sentences from the dataset along with their labels are used to train the BERT classification model. Here, BERT functions as a classifier to determine the author of the given sentences. The expectation is that a highly accurate classifier not only demonstrates that there are distinguishable features among authors, making the Generation phase possible but also will provide a reliable model for evaluating newly generated sentences, verifying whether they align with the writing style of the intended author. On the other hand, as illustrated in the Detection phase, there is also a syntactic feature extraction path. This path aims to perform a similar function as BERT but relies on syntactic features. We expect these syntactic features, comprising both low-level and high-level features, to clearly demonstrate differences between various writing styles.

The generator model will be trained on sentences concatenated with their labels at the beginning. The main idea behind this approach is that, since we don't have paired samples, we explicitly add the label of each sentence to help the model understand patterns shared by sentences with similar labels. After training the model, new sentences are generated by providing seeds with different labels and randomly extracted words. The generator then completes these sentences based on the initial labels. Finally, the generated sentences must be preprocessed to evaluate their quality and remove tags.

As mentioned in the introduction, evaluating generated sentences with common techniques has several challenges. Hence, to make the evaluation more systematic and practical, we will use the AI-evaluate-AI technique. In addition to using a large language model like BERT for evaluation, we employ a feature-based evaluation to further assess the quality of the generated text. As demonstrated in the next section, we will show that extracting syntactic features can help highlight stylistic differences between authors. For example, in prior studies (Rajaei Moghadam et al., 2024a,b), we showed how the syntax in speeches by U.S. presidents differed from the syntax in their written works. Similarly, we will extract high-level and low-level syntactic features using Stanford CoreNLP (Manning et al., 2014) to compare the generated sentences with the original dataset.

In summary, our proposed workflow combines detection, generation, and evaluation techniques to accurately create sentences in different writing styles and assess them in a meaningful way. The goal is to ensure that the generated sentences not only reflect the stylistic features of the target authors but also maintain the consistency and fluency of any generated sentences.

3.1 Detection

Using our previous work (Rajaei Moghadam et al., 2024a,b), we are going to analyze and evaluate the generated sentences and compare them with the original sentences by extracting the low-level and high-level syntactic features of each sentence. The dataset used in the above papers contained sentences of transcribed speeches and written books by United States presidents. For sentence extraction, the *nltk* library (Bird et al., 2009) was used, while CoreNLP (version 4.5.7) was employed for tokenization and word counting.

Low-level features (Rajaei Moghadam et al., 2024b), are categorized into three different aspects: morphological aspects, which include average syllables per word, average words per sentence, and average characters per word; lexical aspects, which include the number of words in a sentence, percentage of different POS, and percentage of personal pronouns; and syntactical aspects, which include percentage of subordinate clauses, depth of parse tree, percentage of noun phrases, the average length of noun phrases, percentage of yes/no questions, and percentage of direct wh-questions.

High-level syntactic features that have been introduced in Rajaei Moghadam et al. (2024a) contain: Pronoun and noun phrases in the subject, passive and active sentences, comparative and superlative, imperative structures, conjunction phrases, and prepositional phrases.

The number of words in a sentence was used as an aid to understanding syntactic complexity since longer sentences often indicate more complex ideas or more detailed information. Also, the height of the parse tree can be considered as an indicator of sentence complexity.

The analysis includes part-of-speech (POS) tags, which reveal structural, stylistic, and functional aspects of the text. This parsing model employs context-free grammar, along with associated probabilities for each rule, to generate a parse tree for each sentence. The token and sentence boundaries and other features provided by CoreNLP help in the analysis process. We rolled up the multiple types of nouns and verbs provided by CoreNLP into one type for each.

One important issue is the identification of passive sentences. According to Aygen (2016), the active voice is the typical form in which the subject of the sentence is the agent. To do this, the PassivePy package (Sepehri et al., 2023) in the SpaCy library (Honnibal et al., 2020) enables us to compute active, agentless passive, and agentive passive forms.

The results showed that the most significant features identified are sentence length, verb percentages, noun percentages, and prepositional phrases. Also, despite having fewer samples for long sentences, using long sentences improves the accuracy across all models. Increasing the sentence length also raised the importance ranking of prepositional phrases. Also, combining both sets of features improves the model's performance. Finally, based on our analysis, U.S. presidents are more likely to use prepositional phrases and longer sentences in their speeches than in their books.

Based on the results of the previous works mentioned above, we expect that there will be distinguishable features among the different authors in our dataset. We will examine whether BERT-based detection methods and detections based on syntactic features can identify differences between the five authors. This analysis will not only enhance our understanding of the linguistic characteristics specific to each author but also allow us to compare real sentences with sentences generated by our model. In other words, our evaluation technique involves calculating the similarity between the patterns found in real and generated data. Our preliminary results using BERT to detect the writing styles of sentences from five different authors show 84 percent in both accuracy and F1 score metrics.

At the same time, we are working on Graph Neural Networks (GNNs) (Zhou et al., 2020), which are deep neural networks that have attracted the attention of researchers across various fields. Five different Graph Neural Network (GNN) models were applied to understand and classify each sentence based on the author's writing style. We utilized a message-passing spatial method (GraphSAGE), an attention-based method (GAT), spectral methods (GCN, ChebNet), and a highly expressive GNN model (GINConv). In our preliminary results, we demonstrated the power of GNNs in extracting patterns behind the different writing styles of authors by using only dependencies between words in each sentence.

In that study, we extracted only the dependencies between words in different sentences, which represent a minimal set of information that can be derived from a sentence. We processed the sentences using the CoreNLP parser to extract word dependency information. In these graphs, each node represented a word in the sentence, while edges captured the grammatical dependencies between the words.

Table 1: The primary results of the generated sentences using the seeding technique for the expected writing styles.

Target	Seed	Generated Sentences
Charles Dickens	<0> When	**When** I had got my breath, I said, "I am going to London.
Jane Austen	<1> When	**When** they were gone, she sat down again, ...
Mark Twain	<2> When	**When** the sun went down, we had a grand supper, ...
Louisa May Alcott	<3> He	**He** was a man of great courage, and a man of great resolution.
Herman Melville	<4> He	**He** had been a very good-looking young man, ...
Charles Dickens	<0> He	**He** was a man of great talent, and his music was considered ...
Jane Austen	<1> A	**A** very few minutes more, however, and she was in the street, ...
Mark Twain	<2> A	**A** few of the boys had gone to the river, ...
Louisa May Alcott	<3> A	**A** few words of explanation will make it clear.

3.2 Generation

The most important and challenging part of the pipeline is generating texts, particularly when considering the challenges of working with style and the lack of paired datasets for training the models. On the other hand, based on the related work and the identified gaps that align with our main goal, in this section we aim to generate different writing styles by utilizing GPT models.

In order to address the challenges with training generators without parallel data, we add identifier tags at the beginning of each sentence as an indicator of each of the five different authors, to force the models to learn and capture the patterns of the writing style of each author. For example, <A0> in "<A0> Why, I have been ashamed of your moroseness there! <end>" indicates that the sentence belongs to Charles Dickens.

As explained, we will train the GPT-3 models using author tag identifiers for each sentence in the dataset. This involves using the seeding technique to prompt the model to generate the rest of the words in a sentence. For example, by adding an author identifier, the expectation is that the model will generate sentences similar to the writing style of that author. The seeding process can start with only a tag or with a tag that is followed by one or more words. For instance, a seed could be "<A0>", "<A0> hello", or "<A0> today is". Hence, the model generates sentences in different writing styles, rather than transferring the writing style.

We use the GPT-3 (Brown, 2020) structure for sentence generation, as it is one of the publicly available state-of-the-art models, known for its remarkable ability to produce coherent and contextually appropriate text from given prompts. Specifically, we train GPT-Neo 1.3B (Black et al., 2021), an open-source autoregressive language model developed by EleutherAI, which contains 1.3 billion parameters. After generating the sentences, we apply post-processing techniques to improve their quality. For instance, we remove the tags from the beginning and the ending of sentences and check for issues like repeated words or incomplete sentences. At the end of this step, we aim to have a collection of polished, high-quality generated sentences.

Our goal is to achieve the final result with the highest possible accuracy within the limitations of data and resources. It is worth mentioning that preliminary results have been obtained. The trained model, after 3 epochs, achieved 86% on both accuracy and F1 score metrics, which seems acceptable. The results showed that the introduced model is capable of generating sentences based on arbitrary seeding prompts. Table 1 reports some of the generated results, which need improvement in terms of both their assigned classes and their fluency and clarity.

Although the generated sentences, such as those reported in Table 1, represent our primary results, initial evaluations by a human expert provide evidence that it seems the model has learned distinct authorial patterns. For example, in the first sentence attributed to Charles Dickens, we observe British English usage such as "had got", since American English typically uses "had gotten." In the second sentence, attributed to Jane Austen, the reference to parties and social behavior clearly aligns with themes frequently explored in her stories. Regarding the fourth sentence by Alcott, words like "courage" and "resolution" reflect the language commonly found in novels from her period. The fifth sentence, attributed to Melville, interestingly focuses on men and boys, a theme prevalent throughout his works. In the seventh sentence, attributed to Austen, it is not surprising to encounter a depiction of a woman busy shopping in the street, a typical scenario in Austen's novels. For the eighth sentence, attributed to Mark Twain, the importance of boys and references to Mississippi strongly reflect his characteristic themes. Lastly, the sentence attributed to Alcott resembles a direct note to the reader, a common stylistic feature in 19th-century literature. Future evaluation by human experts and AI-Evaluate-AI can potentially clarify the accuracy of patterns learned by the generator model.

3.3 Evaluation

The final experiment involves evaluating the generated sentences. As shown in Figure 1, we plan to use the AI-evaluate-AI techniques. One of the main reasons behind this approach is the inherent ambiguity in evaluating an author's writing style.

All generator models, like other models, provide metrics such as accuracy or F1 score for evaluation. However, achieving high values for these metrics does not necessarily reflect true accuracy in generating distinct writing styles, so these metrics in the generative model can not reflect the performance of the model in different writing styles. Alternatively, involving human evaluation adds further complexity. Imagine a scenario where a generator produces a sentence, and we ask a group of humans to identify the writing style from among five 19th-century authors. How reliable would their evaluation be? The complexity of this task presents significant challenges.

Another method to improve the reliability of human evaluators in such a process is to use a preliminary test. For instance, we could test participants on the training data and only involve those who achieve high accuracy in the evaluation process. However, this approach significantly increases both the time and cost of evaluation.

Therefore, our proposal for evaluating generated sentences involves using a detector model that has demonstrated high accuracy in training and test data. For example, if we have a BERT model with high performance in author detection, we can use it for quick and cost-effective evaluation of generated text, while factoring in the reliability of the model. Also, we are going to use the feature-based approaches, outlined in the detection section, to compare both the original and generated sentences and determine how closely high-level and low-level syntactic features exhibit similar patterns for each author. Furthermore, a comparison between generated sentences and original sentences allows us to determine whether a model has memorized each author's sentences or not; in other words, we can check for overfitting in the model.

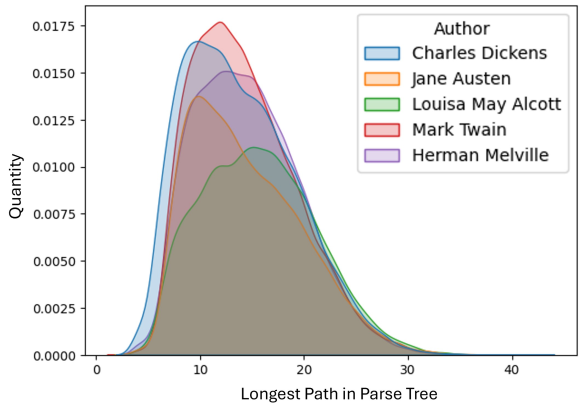

Figure 2: Histogram of the longest path in the parse trees of sentences.

Preliminary analysis of syntactic features in the original sentences reveals distinct patterns that merit deeper investigation in comparison with the generated sentences. For instance, Figure 2 presents a histogram of the longest path in the parse tree. Notably, Alcott (green) exhibits a distribution pattern distinct from Twain (red). The diagram indicates that most sentences by Twain have shorter paths in their parse trees. Conversely, Alcott's sentences show a more uniform distribution across various path lengths. This suggests that Mark Twain tends to write simpler sentences than Louisa May Alcott.

3.4 Conclusion and Future Work

This study contains three main sections: detection, generation, and evaluation, each focusing on different authors' writing styles. In the first section, using the established framework from our previous work, we analyzed writing styles based on their unique syntactic characteristics and classified them using machine learning models, as well as LLMs and GNNs. In the second section, we trained a GPT-3 model on a dataset containing unpaired sentences from five different authors. Preliminary results indicated that the generated sentences reflect meaningful stylistic differences among the authors. The final section focuses on evaluation, where we compare generated sentences with real sentences using both feature-based and LLM-based approaches.

References

Gulsat Aygen. 2016. *English Grammar: A Descriptive Linguistic Approach*, third edition. Kendall Hunt.

Federico Betti, Giorgia Ramponi, and Massimo Piccardi. 2020. Controlled text generation with adversarial learning. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 29–34.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large scale autoregressive language modeling with Mesh-Tensorflow. Retrieved from https://doi.org/10.5281/zenodo.5297715.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Mariano de Rivero, Cristhiam Tirado, and Willy Ugarte. 2021. Formalstyler: GPT based model for formal style transfer based on formality and meaning preservation. *In Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, 1:48–56.

Yuheng Du, Shereen Oraby, Vittorio Perera, Minmin Shen, Anjali Narayan-Chen, Tagyoung Chung, Anushree Venkatesh, and Dilek Hakkani-Tur. 2020. Schema-guided natural language generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 283–295.

David I Holmes. 1998. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3):111–117.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. doi: 10.5281/zenodo.1212303.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1587–1596.

Chih-Te Lai, Yi-Te Hong, Hong-You Chen, Chi-Jen Lu, and Shou-De Lin. 2019. Multiple text style transfer by using word-level conditional generative adversarial network with two-phase training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3579–3584.

Joosung Lee. 2020. Stable style transformer: Delete and generate approach with encoder-decoder for text style transfer. *arXiv preprint arXiv:2005.12086*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-LM improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 5108–5118.

Yiwei Lyu, Tiange Luo, Jiacheng Shi, Todd C Hollon, and Honglak Lee. 2023. Fine-grained text style transfer with diffusion-based language models. *arXiv preprint arXiv:2305.19512*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Mina Rajaei Moghadam, Mosab Rezaei, Gülşat Aygen, and Reva Freedman. 2024a. Text vs. transcription: A study of differences between the writing and speeches of Us presidents. In *the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 352–361.

Mina Rajaei Moghadam, Mosab Rezaei, Miguel Williams, Gülşat Aygen, and Reva Freedman. 2024b. Investigating lexical and syntactic differences in written and spoken English corpora. In *the 37th International FLAIRS Conference Proceedings*.

Amir Sepehri, Mitra Sadat Mirshafiee, and David M Markowitz. 2023. PassivePy: A tool to automatically identify passive voice in big text data. *Journal of Consumer Psychology*, 33(4):714–727.

Youzhi Tian, Zhiting Hu, and Zhou Yu. 2018. Structured content preservation for unsupervised text style transfer. *arXiv preprint arXiv:1810.06526*.

James Wayman, Nicholas Orlans, Qian Hu, Fred Goodman, Azar Ulrich, and Valorie Valencia. 2009. Technology assessment for the state of the art biometrics excellence roadmap. *Mitre Technical Report*, Volume 2. FBI.

Christopher C Yang, Hsinchun Chen, Michael Chau, Kuiyu Chang, Sheau-Dong Lang, Patrick Chen, Raymond Hsieh, Daniel Zeng, Fei-Yue Wang, and Kathleen M Carley. 2008. Intelligence and security informatics. *IEEE ISI 2008 International Workshops: PAISI, PACCF, and SOCO*.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.