A generalized solution to verify authorship and detect style change in multi-authored documents

Rohan Leekha
MIT Lincoln Laboratory
Lexington, Massachusetts, USA
Rohan.Leekha@ll.mit.edu

Courtland VanDam
MIT Lincoln Laboratory
Lexington, Massachusetts, USA
courtland.vandam@ll.mit.edu

***Abstract*—Identifying changes in style can be used to detect multi-authored social media accounts, plagiarism, compromised accounts, and author contributions in long documents. We propose an approach to recognize changes in authorship using large language models. Our approach leverages sentence-level contextual embeddings and semantic relationships. First we expand the training set by adding adversarial examples to the minority class [5], [13], [17]. Then we fine-tune a sequence classification transformer model to detect style change. Our approach outperforms all baselines of PAN21 with macro F1-scores of 0.80, 0.74, and 0.70 for detecting style changepoint between paragraphs, closed-set author ID per paragraph, and style changepoint between sentences, respectively. Our approach also performs better than the leading competitors in PAN22. Also, we achieved a five percent improvement in macro F1-score (0.78) on the newly introduced DarkReddit+ dataset for authorship verification.**

I. Introduction

Style change detection (SCD) has become of utmost importance. By detecting style change, we can find instances of plagiarism [4] and link multiple social media accounts related to trolls and hackers [7]. Style change detection is defined as follows: given a document, determine the number of authors and at which positions the author changes. In this research, we detect style change between paragraphs and between sentences. We also identify the authors of the respective paragraphs based on their writing styles.

Previous research predominantly focused on linguistic features, text-based features, and token-based embedders (e.g. BERT) with mixing contextual (language model based), and semantic style features for classification. These features and approaches perform well on large datasets when long passages from each author are available. However, texts today are shorter and more informal, e.g. social media and blog posts. The traditional SCD approaches have several limitations when applied to social media text.

1) Short sentences are often sparse and use informal language (For example, varying word order, incorrect usage of punctuation and spelling errors) making linguistic features unable to accurately characterize writing style of documents on a sentence level [3], [15].
2) Training Siamese networks and graphical convolutional neural networks [12] for style change detection can be computationally expensive to train from scratch (increase in training time).
3) Previous research combined large language models for embeddings with traditional machine learning models, e.g. logistic regression or random forest. Such an approach results in a loss of important information [20], [6]. The language models used only capture individual token-based information and do not fully capture the contextual meaning of whole sentences [20], [6].
4) SCD research that used language models for classification primarily used masked language modeling (MLM). MLM (e.g. BERT, RoBERTa) does not learn semantic relationships of words within a sentence well [16].

We overcome the aforementioned limitations as follows. We address the first limitation by expanding the training set with adversarial examples. For texts from the minority class, we added examples with 20% of the words swapped. This made the model more robust and generalizable [5]. To address the second limitation, we fine-tune a pretrained large language model on examples of style change. Fine-tuning a pretrained model can achieve higher accuracy with less data than training a model from scratch. Addressing the third limitation, we use a neural network classification layer to avoid suffering the same loss of information that ML models experience. To overcome the last limitation, we use MPNet, a language model that unifies both Masked Language Modeling (MLM) and Permuted Language Modeling (PLM) for modeling and classification tasks [19].

In our approach, we use a sentence-based tokenizer combined with a sentence-based transformer model to detect author change. A sentence tokenizer encodes a sentence as a

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Air Force.

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey

ACM ISBN 979-8-4007-0409-3/23/11…$15.00
http://dx.doi.org/10.1145/3625007.3627589

whole, preserving the contextual meaning of the text. Next, a large language model (LLM) with a classification head embeds the sentences and predicts whether an author change occurred. Our approach out-performs existing methods on the PAN CLEF 2022 style change detection challenge [4] and the newly released DarkReddit+ dataset for author verification [11].

Section II highlights the datasets and tasks used in this study. We present related work in Section III. Section IV describes the methodologies we developed for style change detection and authorship verification. Section V compares our results with other approaches. We discuss limitations of our proposed method in Section VI. We conclude and talk about the future scope of this research in Section VII.

II. Dataset

We explore author style change detection on two forum datasets: Stack Exchange and Reddit.

A. PAN22 Authorship Analysis: Style Change Detection

PAN is a series of tasks in natural language processing. The style change detection tasks in 2022 focused on Stack Exchange forums.

The goal of the style change detection task is to identify, based on an intrinsic style analysis, the text positions within a given multi-author document at which the author changes. Multi-author documents have been largely understudied. The shared PAN 2022 style change detection task focuses on three sub-tasks [4]. Table 1 describes the PAN2022 datasets.

1) Style Change Basic: given a text with exactly two authors and one change, identify the position of change between two paragraphs.
2) Style Change Advanced: given a text with one or more authors, identify the location of style change at the paragraph level and assign paragraphs to the appropriate author.
3) Style Change Real-World: given a text written by one or more authors, detect all positions of writing style change. The style change can now occur between sentences within a paragraph.

B. VeriDark: A Large-Scale Benchmark for Authorship Verification on the Dark Web

To check the generalizability of our classification model, we make use of a large-scale authorship verification Reddit dataset called DarkReddit+. DarkReddit+ is collected from a Reddit forum called /r/darknetmarkets, which features discussions related to trading on DarkNet marketplaces [11]. It contains same author (SA) and different author (DA) pairs of comments from the /r/darknetmarkets subreddit. The dataset, referred to as Task 4, consists of 120k pairs, which are divided into 106,252 training pairs, 6,124 validation pairs and 6,623 test pairs.

4) Multiauthor: Given two short texts, identify the number of authors

PAN Dataset	Description
1. Style Change Basic	2 Authors/doc 1 Change/doc 7.8 Avg Paragraphs/doc 1581.0 Avg Characters/doc
2. Style Change Advanced	1-5 Authors/doc 4 Avg Changes/doc 7.5 Avg Paragraphs/doc 1880.1 Avg Characters/doc
3. Style Change Real-World	1-5 Authors/doc 8 Avg Changes/doc 16.0 Avg Sentences/doc 1834.3 Avg Characters/doc

TABLE I
DESCRIPTION OF THE THREE DATASETS USED IN THE THREE TASKS FOR THE PAN'22 CHALLENGE

III. Related Work

Previous work in style change detection deploys multiple linguistic features including syntactic, lexical, or semantic features. Zlatkova et al. [21] developed the winning solution for PAN CLEF 2018, where they used Term Frequency-Inverse Document Frequency (TF-IDF) features combined with other linguistic features to classify style change using machine learning algorithms. The best approach was achieved by combining LightGBM with TF-IDF and linguistic features. Singh et al. [18] extracted linguistic features from each paragraph in each document and used the absolute differences between the feature vectors corresponding to pairs of paragraphs as input to a Logistic Regression classifier. Linguistic features[1], e.g. special character frequency or punctuation frequency, provide good discriminating power for style change detection. Al-Shamasi and Menai [1] proposed an ensemble-based authorship clustering method to detect style change in multi-authored documents for PAN CLEF 2022 by clustering documents into stylistically homogeneous groups based on sentence similarity. For PAN CLEF 2022 Alvi et al. [2] extracted hand-crafted linguistic features from multi-authored documents. These features were then used to detect style changes using machine learning algorithms (Random Forest, Logistic Regression, Nave Bayes etc.) for Task 1 and Task 2. They then apply K-Means Clustering to partition the paragraphs of a given document into n clusters, where n is the number of authors to detect style change at a sentence level for Task 3.

More recently, researchers shifted to using deep learning models trained from scratch to detect style change in text-based documents. Nath [14] converted numbers to their textual interpretations and used a Siamese neural network composed of Bidirectional Long Short Term Memory (BiLSTM) with a distance layer and an activation layer for classification.

With the development of large-scale transformer-based language models in 2018, PAN data challenges saw an influx of approaches that used such models. For example, Iyer and

[1]https://pypi.org/project/writeprints-static/

Vosoughi [8] used a BERT pretrained bidirectional model to tokenize and generate embeddings for the sentences in each document and used the embeddings to train a Random Forest classifier. The winning solution of PAN CLEF 2021 by Strom [20] extracted text level contextual embedding features using language models and linguistic features. This was followed by a boosting ensemble for classification and demonstrated that transformer-based models boost performance.

Building on Strom [20], PAN CLEF 2022 researchers used transformers-based approaches. Jiang et al. [9] used a transformer model, ELECTRA, to detect style change in multi-authored documents. The best results were achieved on a max sentence length of 64 tokens for Task 1 and 128 characters for Task 2 and Task 3. ELECTRA changes the generated Masked Language Model (MLM) pre-trained task into the discriminative Replaced Token Detection (RTD) task, which allows for higher classification accuracy than BERT based models. Lin et al. [10] proposed an ensemble neural network to detect style change in multi-authored documents by taking predictions from the fine-tuned BERT, RoBERTa, and ALBERT transformers and use a majority voting scheme for classification.

IV. Methodology

We build on how previously proposed transformer language models can learn semantic relationships between texts [16] by using MPNet, a large language model proposed by Microsoft[2] in our research. While the majority of language models are pretrained using either masked language modeling (MLM) or permuted language modeling (PLM) objectives, MPNet takes the best of both language models [19]. The first step of our algorithm consisted of adding adversarial perturbations to the training data in the form of swapping or deleting words from a sentence. This was followed by tokenizing the perturbated sentences using a sentence tokenizer. These tokenized sentences were used to fine-tune the MPNet for the task of sentence classification. Our framework is shown in Figure 1. Our algorithm is also described in Algorithm 1.

A. Adversarial Perturbation for Style Change Detection

We add an additional step to improve class balance in the PAN CLEF dataset by adding adversarial perturbations to the minority class before tokenization and sentence embedding generation. By training large language models like MPNET on perturbated data we widely increase their ability to generalize over testing data and the overall model robustness [13], [17].

We use the following two techniques to increase the size of the minority class.

[2]https://github.com/microsoft/MPNet

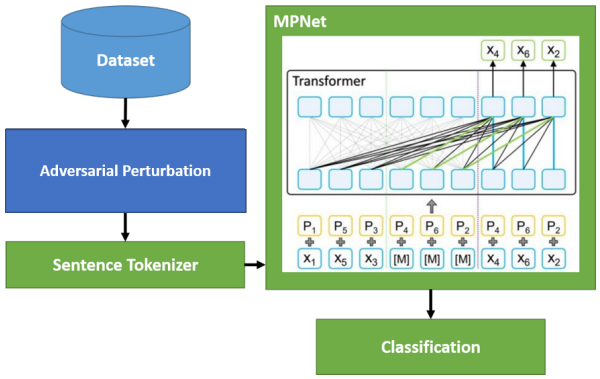

Fig. 1. Our proposed framework. We tokenize into sentences, and pass sentences to MPNet with a classification head to identify author and change points.

Algorithm 1 MPNet for Style Change Detection. $*$ indicates both sentences s_{train} and labels y_{train} are arguments. This is the outermost algorithm. For details on PerturbSentence, see Algorithm 2. For details on fine-tuning MPNet, see Algorithm 3

Input: Training sentences s_{train}, Training labels y_{train}, Test sentences s_{test}
Output: Test labels y_{pred}
$s_{minority} \leftarrow$ SELECTMINORITYCLASS(s_{train})
$s_{perturbed} \leftarrow$ PERTURBSENTENCE($s_{minority}$)
$s_{train} \leftarrow$ CONCATENATE($s_{train}, s_{perturbed}$)
$s_{train}*, s_{val}* \leftarrow$ TRAINTESTSPLIT($s_{train}*$)
$model \leftarrow$ FINE-TUNE MPNET($s_{train}*$)
$y_{pred} \leftarrow$ EVALUATE MPNET(s_{test})
return y_{pred}

For each sentence in the minority class, do either:

1) Random word swap: Randomly choose 20% words in the sentence and swap their positions.
2) Random word deletion: Randomly choose and delete 20% words in a sentence.

By training a classifier on adversarial examples [13], it can learn more robust features that are less susceptible to small perturbations.

B. Fine-tuning Vanilla MPNET

For our method, paragraphs longer than 512 tokens were truncated, while paragraphs whose length were less than 512 were padded to the length of 512. We experimented with different sequence lengths and identified the sequence length of 512 to provide the most accurate results. These sequences are used to fine-tune a pretrained MPNet sequence classification model with a classification head on top[3]. We set the learning rate as 2e-5 and a weight decay of 0.01 for all classification tasks discussed above. The batch size for train and test was set to 16.

[3]See Algorithm 3 for pseudocode

Algorithm 2 PerturbSentence: This algorithm shows how we perturb sentences in the minority class. We either delete $n\%$ of the words or swap $n\%$ of the words for each sentence. We then add these perturbed sentences to the training set.

Input: Sentences from minority class
Output: Perturbed sentences
for $sentence \in$ minority class **do**
  $s_{perturbed} \leftarrow$ RANDOMWORDSWAP
  or
  $s_{perturbed} \leftarrow$ RANDOMWORDDELETE
end for
return $s_{perturbed}$

We used 90% to train the models and the remaining 10% for parameter tuning. All of the above parameters were obtained by fine-tuning the model on the 10% holdout validation data.

Algorithm 3 Fine-tune MPNet: This is how we fine-tune MPNet. For ModelForSequenceClassify, we call transformers AutoModelForSequenceClassification.

Input: Training dataset X_{train} and validation dataset X_{val}
Output: Fine-tuned MPNet classifier
$model \leftarrow$ MODELFORSEQUENCECLASSIFY
$trainer \leftarrow$ TRAINER($model$, X_{train}, X_{val})
$trainer$.TRAIN
return $trainer$

V. RESULTS

We compare MPNet against the top competitors of PAN 2021 and PAN 2022. For PAN 2021 competitors, we re-trained their models on the PAN 2022 dataset, using the same train/validation splits described above. Specifically we compare against Strom [20], Nath [14] and Singh et al. [18]. Most PAN 2022 competitors presented the macro-F1 on the development set in their papers. Those results are reported in Table II.

For Task 1, MPNet and Adversarial MPNet outperformed all other methods. MPNet Adversarial outperformed all other methods on Task 2. For Task 3, MPNet and Adversarial MPNet performed well, but an ensemble of BERT-based models Lin et al. [10] achieved the highest macro-F1.

For the DarkReddit+ dataset, we again compare against Strom [20] and Jiang et al. [9] and add also the method proposed by Manolache et al. [11]. We show the macro F1-scores in Table III. MPNet and MPNet adversarial outperformed all other methods.

We observe that all of the methods that use transformers achieved higher macro-F1 on the PAN 2022 data than those that do not. We hypothesize that this is due to the complicated semantic relationships between features and the contextual dependencies among large sentences. Further, it is believed that linguistic features [18] captures less stylistic information

Method	Task 1	Task 2	Task 3
Strom [20]	0.656	0.658	0.609
Nath [14]	0.462	0.364	0.518
Singh et al. (2021) [18]	0.549	0.441	0.659
Alvi et al. (2022) [2]	0.710	0.330	0.580
Jiang et al. [9]	0.670	0.730	0.690
Lin et al. [10]	0.740	0.540	**0.730**
BERT (MLM)	0.738	0.714	0.672
XLNET (PLM)	0.783	0.723	0.674
MPNet (MLM + PLM)	**0.806**	0.732	0.690
MPNet Adversarial (MLM + PLM)	0.796	**0.741**	0.696

TABLE II
RESULTS ON THE PAN 2022 STYLE CHANGE DETECTION TASK. METRIC IS MACRO-F1 SCORE.

Method	DarkReddit+ Dataset
Manoache et al. [11]	0.721
Strom [20]	0.656
Jiang et al. [9]	0.726
MPNet	0.771
MPNet Adversarial	**0.779**

TABLE III
RESULTS ON TASK 4, AUTHOR VERIFICATION ON /R/DARKNETMARKETS REDDIT POSTS. METRIC IS MACRO-F1 SCORE.

of the author than transformer-based models.While LSTM-based models [14] are less able to capture subtle instances of style change in multi-authored documents.

Additionally, MPNet achieves higher macro-F1 than BERT-based models. We observe a 6% improvement over simply fine-tuning BERT. Among those who used BERT in their systems, Strom [20] achieved the lowest macro-F1 on Task 1 and Task 3. Their system used BERT only for feature generation and then used traditional machine learning models for classification. Lin et al. [10] achieved higher macro-F1 using BERT for classification. However, even an ensemble of BERT-based models could not match the performance of MPNet. We find that MPNet with its dual training objectives, captures more information about author style than BERT alone.

We also observe that XLNet (PLM) achieves higher macro-F1 compared to BERT (MLM), but MPNet achieves even higher macro-F1. It's the combination of the training objectives of MLM and PLM that results in the highest accuracy.

Furthermore, adding more training examples by adversarially changing training examples improves the generalizability of the model. We observe that randomly swapping words gave better results than randomly deleting words, shown in Table IV. Specifically for task 2 and task 3, swapping 20% of words in the new samples increase the performance over using the original dataset without added samples.

One reason why randomly swapping words was a better approach than deleting is that swapping words alters the

syntax of the sentence less than randomly deleting words. By preserving the overall syntax of the text, the classifier is better at detecting subtle changes in authorship style that are more difficult to detect when possible parts of speech are removed. Deletions also create gaps in the text that could make it more difficult for the classifier to understand the context and identify relevant features for detecting changes in authorship style.

Type	Percent	Task 1	Task 2	Task 3	Task 4
Baseline	0	**0.806**	0.732	0.690	0.771
Swap	10	0.791	0.66	0.665	0.774
Swap	20	0.796	**0.741**	**0.696**	0.779
Swap	30	0.795	0.739	0.685	0.781
Swap	40	0.788	0.691	0.665	0.774
Swap	50	0.794	0.686	0.671	0.763
Delete	10	0.781	0.660	0.642	0.767
Delete	20	0.767	0.651	0.626	0.777
Delete	30	0.799	0.685	0.676	0.759
Delete	40	0.771	0.681	0.655	0.741
Delete	50	0.787	0.676	0.629	0.743

TABLE IV
RANDOMLY SWAPPING WORDS ACHIEVED HIGHER MACRO-F1 COMPARED TO RANDOMLY DELETING WORDS. BEST PERCENT OF WORDS TO SWAPS OR DELETES IS AROUND 20% AND 30%.

We compared different percentages of words swapped or deleted and found that swapping 20% of the words or deleting 30% of the words have the highest F1-score. Lower percentages add samples that are too similar to existing samples, which does not provide additional entropy to the training algorithm. Deleting or swapping too many words adds too much entropy to the model. The words deleted or swapped may be the words the model appropriately uses to identify a style change or author change occurred. Future work will focus on more selective word swapping/deletion, e.g., swap only nouns or only adjectives.

VI. LIMITATIONS

Our approach, although shown to outperform the current state-of-the-art models on the PAN22 style change detection challenge, is not without limitations. A few potential limitations of our approach to detect change in authorship style, include but may not be limited to:

1) There are only a limited number of perturbation techniques available, and they may not be effective in all situations. For example, swapping words may be less effective at detecting changes in authorship style in documents with more complex language or structure.
2) Large language models like MPNET are complex and difficult to interpret and understand how a model is making its predictions. This can make it challenging to understand why the model is making certain mistakes or to identify areas for improvement.
3) Large language models require more compute power, e.g. GPUs, which means this approach is not effective in low resource environments.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a solution to the PAN 2022 shared task on style change detection, attempting to answer the questions: (1) Given a document, find the position of author change between paragraphs. (2) Assign all paragraphs of the text uniquely to the correct author. (3) Find all positions of writing style change at the sentence level. (4) Verify whether a paragraph has been written by single or multiple authors. We used a sentence-based tokenizer to encode whole sentences into a 512-dimension vector to conserve their contextual meaning before fine-tuning a pre-trained sentence-based language model (MPNet) for classification. Our approach achieved macro F1-scores of 0.80 on Task 1, 0.73 on Task 2, 0.69 on Task 3 and 0.77 on the DarkReddit++ dataset outperforming all other competing baselines on Task 4. This showcases the approaches ability to generalize well on language modeling tasks that involve contextual understanding of sentences like authorship verification and style change detection. For future work we aim to build stacking ensembles using predictions from various sentence-based transformers followed by a meta classification head.

REFERENCES

[1] Shams Al-Shamasi and Mohamed Menai. Ensemble-based clustering for writing style change detection in multi-authored textual documents. In *CLEF 2022 Labs and Workshops, Notebook Papers*. CLEF, 2022.

[2] Faisal Alvi, Hasan Algafri, and Naif Alqahtani. Style Change Detection using Discourse Markers. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *CLEF 2022 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2022.

[3] Zaira Hassan Amur, Yew Kwang Hooi, Hina Bhanbhro, Kamran Dahri, and Gul Muhammad Soomro. Short-text semantic similarity (stss): Techniques, challenges and future perspectives. *Applied Sciences*, 13(6):3911, 2023.

[4] Janek Bevendorff, Berta Chulvi, Elisabetta Fersini, Annina Heini, Mike Kestemont, Krzysztof Kredens, Maximilian Mayerl, Reyner Ortega-Bueno, Piotr Pezik, Martin Potthast, et al. Overview of pan 2022: Authorship verification, profiling irony and stereotype spreaders, style change detection, and trigger detection. In *European Conference on Information Retrieval*, pages 331–338. Springer, 2022.

[5] Junfan Chen, Richong Zhang, Zheyan Luo, Chunming Hu, and Yongyi Mao. Adversarialword dilution as text data augmentation in low-resource regime. *arXiv preprint arXiv:2305.09287*, 2023.

[6] Maël Fabien, Esaú Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. Bertaa: Bert fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, 2020.

[7] Tommi Gröndahl and N Asokan. Text analysis in adversarial settings: Does deception leave a stylistic trace? *ACM Computing Surveys (CSUR)*, 52(3):1–36, 2019.

[8] Aarish Iyer and Soroush Vosoughi. Style change detection using bert. In *CLEF (Working Notes)*, 2020.

[9] Xinyin Jiang, Haoliang Qi, Zhijie Zhang, and Mingjie Huang. Style Change Detection: Method Based On Pre-trained Model And Similarity Recognition. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *CLEF 2022 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2022.

[10] Tzu-Mi Lin, Chao-Yi Chen, Yu-Wen Tzeng, and Lung-Hao Lee. Ensemble Pre-trained Transformer Models for Writing Style Change Detection. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *CLEF 2022 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2022.

[11] Andrei Manolache, Florin Brad, Antonio Barbalau, Radu Tudor Ionescu, and Marius Popescu. Veridark: A large-scale benchmark for authorship verification on the dark web. *arXiv preprint arXiv:2207.03477*, 2022.

[12] Jorge Alfonso Martinez-Galicia, Daniel Embarcadero-Ruiz, Alejandro Ríos-Orduña, and Helena Gómez-Adorno. Graph-based siamese network for authorship verification. *Mathematics*, 2022.

[13] John X Morris, Eli Lifland, Jin Yong Yoo, and Yanjun Qi. Textattack: A framework for adversarial attacks in natural language processing. *Proceedings of the 2020 EMNLP, Arvix*, 2020.

[14] Sukanya Nath. Style change detection using Siamese neural networks—Notebook for PAN at CLEF 2021. In Guglielmo Faggioli, Nicola Ferro, Alexis Joly, Maria Maistro, and Florina Piroi, editors, *CLEF 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2021.

[15] Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSuR)*, 50(6):1–36, 2017.

[16] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, abs/1908.10084, 2019.

[17] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917*, 2018.

[18] Rhia Singh, Janith Weerasinghe, and Rachel Greenstadt. Writing style change detection on multi-author documents. In *CLEF (Working Notes)*, pages 2137–2145, 2021.

[19] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020.

[20] Eivind Strøm. Multi-label style change detection by solving a binary classification problem. In *CLEF (Working Notes)*, pages 2146–2157, 2021.

[21] Dimitrina Zlatkova, Daniel Kopev, Kristiyan Mitov, Atanas Atanasov, Momchil Hardalov, Ivan Koychev, and Preslav Nakov. An ensemble-rich multi-aspect approach for robust style change detection. *CLEF 2018 Working Nots of CLEF*, 2018.