Securing Author Privacy Using Large Language Models

Anonymous EMNLP submission

Abstract

Sophisticated machine learning models can determine the author of a given document using stylometric features or contextualized word embeddings. In response, researchers have developed Authorship Obfuscation methods to disguise these identifying characteristics. Despite the growing popularity of large language models like GPT-4, their utility for this purpose has not been previously studied. In this work, we explore the application of popular large language models to the task of author obfuscation, and show that they can outperform a state-of-the-art approach. We analyze their behavior and suggest a personalized prompting technique for improving performance on more difficult authors. Our code and experiments will be made publicly available.

1 Introduction

Author Attribution (AA) and Author Verification (AV) are two classic problems in Natural Language Processing. AA involves predicting the author of a text T from a set of users. AV is a specific case of AA where we verify whether an author u_i is the writer of a given T. With the abundance of online data and advancements in transformer-based language models, AA and AV have become easier tasks than ever. The emergent power of LLMs poses significant privacy threats (Staab et al., 2023), particularly to journalists and human rights activists working under authoritarian regimes who could be affected by successful AA and AV attacks.

To defend against these models, authors employ *Author obfuscation (AO)* approaches to anonymize their writing by altering their writing style while retaining the meaning of the text. With the rise of ChatGPT and similar models, the standard for fluency in algorithm-generated text has increased. These widely accessible models are likely to be used for AO by vulnerable authors, making it crucial to assess their effectiveness for this purpose.

In this study, we explore the abilities of three popular LLMs: GPT-3.5 (Brown et al., 2020), GPT-4 (Achiam et al., 2023), and Gemini (Team et al., 2023) for author obfuscation through different prompts. We compare their obfuscation performance with a state-of-the-art AO technique, Avengers (Haroon et al., 2021), and evaluate the methods based on the extent to which they preserve semantics, readability of the output text, and their success in fooling an external AV model.

2 Related Work

Early AO studies used rule-based methods for sentence transformations, such as contraction replacement or synonym substitution (Castro-Castro et al., 2017; Karadzhov et al., 2017; Potthast et al., 2016). These methods are simple and fast, but reduce fluency and semantic similarity. Many researchers treat AO as a an adversarial attack on AA/AV models, aiming to minimally perturb the input to ensure misclassification while maintaining semantic similarity (Gao et al., 2018; Ebrahimi et al., 2017). Adversarial perturbations are effective against transformer-based classifiers but often degrade text quality (Crothers et al., 2022).

Other studies address the more realistic scenario where the target classifier is unknown, using re-writing methods such as back translations (Keswani et al., 2016; Altakrori et al., 2022). Although effective, these approaches can produce unnatural phrasing and semantic loss. Variational auto-encoders and generative adversarial networks have also been explored for obfuscation (Shetty et al., 2018; Mireshghallah and Berg-Kirkpatrick, 2021). Mutant-X (Mahmood et al., 2019) and Avengers(Haroon et al., 2021) use a genetic algorithm to iteratively substitute words until the text fools the internal classifier. Alison (Xing et al., 2024) is a faster syntactical AO method which replaces multi-token phrases to fool an internal classifier trained on character and POS n-grams.

3 Dataset

The dataset that we work with in this study is IMDb62 (Seroussi et al., 2014) which consists of 62,000 posts by 62 of the most prolific IMDb users. It contains reviews posted on IMDb about different movies and shows. We perform no pre-processing as the nature of the task requires to work with the raw text containing all stop-words and punctuation. We randomly select 9 users from all the users. There are 1000 posts for each user, of which we randomly select 900 as the training data and withhold the remaining 100 reviews as the test set.

4 Method & Evaluation

To change the writing style we use three large language models: GPT-3.5, GPT-4 and Gemini. For each review, we pass it to the models and prompt them to paraphrase the text. We use two different prompts to change the writing style and we aim to compare performance differences between the prompts. In the first prompt, P_1, we ask the models to paraphrase the review ("Rephrase the text below."), whereas in the second prompt, P_2, we explicitly mention in the prompt to paraphrase the review such that it seems like it was written by someone else ("Change the writing style of the text below so it seems like someone else wrote it."). We hypothesize that prompting the model to conceal identifying characteristics in the text will direct its attention to specific features. We evaluate our experiments with three evaluation metrics:

Semantic Similarity. To evaluate semantic preservation in our experiments, we use SBERT (Reimers and Gurevych, 2019) to get semantic embeddings of the reviews and compute the cosine similarity between the reviews. We do not use the common n-gram based metrics such METEOR or BLUE (Banerjee and Lavie, 2005) as they often fail to robustly match paraphrased sentences.

Obfuscation. To evaluate the extent of attribution evasion, we measure the performance drop of an external AV model that we train for each author separately. The bigger $Score_{AV}(Original) - Score_{AV}(Obfuscated)$, the more successful is the Obfuscation.

Fluency. To evaluate fluency, we use the perplexity score calculated as negative log-likelihood by GPT-2 (Radford et al., 2019).

5 Experiments

To evaluate how well the LLMs obfuscate each author, we first train an AV model on the authors' training dataset and test it on the modified reviews. The greater the drop in the performance of the AV model, the more successful author obfuscation is evading detection. We train two models as our AV models: BERT (Devlin et al., 2018) and a logistic regression trained on write-print features (Abbasi and Chen, 2008), a set of linguistic and syntactic features used to identify individuals in cyberspace. The results for both models are presented in Table 1. We find that, as expected, both models achieve high accuracy on the AV task. While the average BERT performance is higher, the logistic regression model with write-print features is more interpretable and allows us to inspect which features are most characteristic of each user (Section 6.3). We will also see that it is more robust to obfuscation.

5.1 Test on Rephrased Reviews

To discover how well the three models obfuscate each author, we prompt the models to paraphrase the reviews using the two prompts described above, and then we pass the modified reviews to the AV model for each user. The results are in Table 1.

User	Original	Gemini P_1	Gemini P_2	GPT 3.5 P_1	GPT 3.5 P_2	GPT 4 P_1	GPT 4 P_2
User 562732	0.99	**0.05**	**0.05**	0.16	0.36	0.31	0.24
User 342623	0.96	0.98	**0.94**	0.98	0.96	0.99	**0.94**
User 306861	1.00	0.85	0.88	**0.76**	0.85	0.89	0.77
User 453228	1.00	0.11	**0.01**	0.41	0.51	0.22	0.20
User 819382	1.00	0.06	0.19	**0.0**	0.11	0.03	0.14
User 4445210	0.96	0.1	**0.05**	0.15	0.46	0.25	0.21
User 1406078	1.00	0.53	**0.52**	0.78	0.84	0.66	0.55
User 1416505	1.00	0.04	**0.0**	0.27	0.29	0.06	0.02
User 2020269	0.97	0.91	0.96	0.54	**0.53**	0.87	0.84
Average	0.98	0.40	**0.40**	0.45	0.54	0.47	0.43

(a) Accuracy scores of BERT on transformed reviews.

User	Original	Gemini P_1	Gemini P_2	GPT 3.5 P_1	GPT 3.5 P_2	GPT 4 P_1	GPT 4 P_2
User 562732	0.96	0.03	**0.0**	0.66	0.69	0.58	0.34
User 342623	**0.90**	0.96	0.96	0.98	0.98	0.98	0.97
User 306861	0.99	**0.94**	0.95	1.0	0.99	1.0	1.0
User 453228	0.99	0.61	**0.30**	0.95	0.94	0.79	0.65
User 819382	0.96	0.10	**0.07**	0.77	0.80	0.90	0.78
User 4445210	0.94	0.14	**0.02**	0.64	0.81	0.66	0.56
User 1406078	0.97	0.96	0.96	**0.92**	0.94	0.93	0.96
User 1416505	0.97	0.39	0.21	0.70	0.81	0.17	**0.11**
User 2020269	0.94	0.99	1.0	0.86	**0.85**	0.95	0.98
Average	0.95	0.56	**0.49**	0.83	0.86	0.77	0.70

(b) Logistic regression's accuracy score on transformed reviews.

Table 1: Accuracy Scores on Transformed Reviews. P_1 is the prompt just asking to rephrase and P_2 is the prompt which we ask the model to conceal the author.

We find that the average BERT AV performance of 0.98 drops very significantly after obfuscation by each model and prompt, to an average accuracy of 0.40, indicating that, in general, commercial LLMs can successfully perform author obfuscation. However, the average obscures the strong bimodal distribution of the AV performance for the nine users in our dataset. For some, the obfuscation works almost perfectly, bringing the AV performance down to 0.0-0.11. Other authors are barely obfuscated, with an AV performance of 0.76-0.99. Gemini performs obfuscation the best against BERT, with the lowest AV accuracy for most of the users.

When pitted against the Logistic Regression (LR) AV model, the commercial LLMs are less successful at obfuscation. The lowest average AV performance of 0.49 is achieved by Gemini under P_2, which explicitly asks the model to conceal the author identity, while GPT-3.5 and 4 have unacceptable average accuracies of 0.70 and up. As with BERT, we observe a bimodal performance distribution, with some users successfully obfuscated and others barely obfuscated at all. Unlike BERT, the LR write-print model is sensitive to the differences between P_1 and P_2. Explicitly asking the models to conceal the identity of the author (P_2), performs better than mere paraphrasing (P_1).

It is interesting to note that despite the variation in performance across AV models, obfuscation models, and prompts, individual users seem consistently either easy or hard to obfuscate. It is possible that there is some consistency in which features are changed by the LLM rephrasing process, and that obfuscation will be successful when the features that are characteristic of a particular user align with that set. In Section 6.3, we analyze what features are being changed when the LLMs rephrase, and how this relates to the characteristics of individual users, and the likelihood that a review will be successfully obfuscated.

5.2 Comparison with Avengers

We compare the obfuscation performance of the commercial LLMs with a state-of-the-art method, Avengers (Haroon et al., 2021). We run the comparison on a random four users out of the original set, as Avengers takes a longer time to generate output for each review. We first train the model for each user in the AV setting. Then we run the model on each user's test set with the parameters set to their default values. The algorithm runs for 25 iterations on each input and we report the fluency and semantic preservation scores on the output of the last iteration. Next, we run the AV models we trained for each user on the obfuscated text generated by Avengers. The scores are in Table 2.

The commercial LLMs produce output that is significantly more fluent. This is to be expected,

Models	**Perplexity Score**	**Semantic Similarity**	**Avg Score on BERT**	**Avg Score on LR**
Avengers	153.4	0.839	**0.57**	0.92
GPT-3.5 - P_1	27.3	0.834	0.61	0.85
GPT-3.5 - P_2	28.0	0.852	0.67	0.86
GPT-4 - P_1	34.4	**0.871**	0.70	0.86
GPT-4 - P_2	32.2	0.853	0.64	0.81
Gemini - P_1	25.8	0.837	0.61	**0.73**
Gemini - P_2	**23.8**	0.799	0.61	**0.73**

Table 2: Comparison of AO methods based on Perplexity Score and cosine similarity score. Lower perplexity scores indicate higher fluency.

as the Avengers algorithm uses a genetic algorithm to iteratively substitute words, which can result in infelicitous phrasings. The commercial LLMs also generally preserve semantic similarity better, though the differences are not as large, and Gemini is significantly worse under P_2.

Avengers obfuscation is comparable with the commercial LLMs. It exhibits similar patterns of a bimodal distribution over users, and more difficulty fooling the LR writeprint model. Overall, our experiments show that LLM-based obfuscation has competitive performance with a SOTA technique, Avengers, outperforming it for some users, while generating text with higher quality and fluency.

6 Analysis

The results in Section 5 show that commercial LLMs can obfuscate authorship with high fluency and semantic preservation, and good average performance. However, their performance is only successful for some users, and does not work at all for others. In this section, we explore their performance against the write-print based Logistic Regression (LR) model, which is easier to interpret than BERT, in order to try to understand what the LLMs are changing about the text when they are prompted to rephrase or obscure authorship, and how this relates to their ability to fool an AV model.

6.1 Features Affected by LLM Rephrasing

We note in Section 5 that per-user performance is quite consistent across the three LLMs and two prompts. We hypothesize that all six approaches are making similar changes to the original text, which may or may not be aligned with the features that make a particular user recognizable.

Table 4 in Appendix A.1 lists the number of features affected by each model and prompt. We see a rough correspondence between these numbers and the average performance of each experiment. Gemini+P_2 has the highest number of features changed, and the highest average obfuscation performance (lowest average AV performance; see

Table 1). GPT-3.5 has the lowest number of features changed, and the lowest average performance.

When we examine the overlaps between the sets of features changed by each model+prompt, our hypothesis regarding consistency is confirmed. Of the 170 write-print features, 71 are changed by all models, and 21 are changed by zero models, meaning that for over half the features, there is no difference between any of the models or prompts.

When GPT-4 was prompted to modify specific stylometric features, while it did increase and decrease two features, *upper case* and *question mark* frequencies, for others, it would only increase a feature and ignore prompts to decrease, or vice versa. (See Appendix A.2.) If a user is characterized by features that an LLM does not "know how to" modify, their authorship will not be obfuscated.

6.2 Predicting Whether a Review Will Evade Author Verification

We hypothesize that the probability that a review will be successfully obfuscated increases linearly with its difference from the original review. We calculate the distance, $D(R, R')$, between the obfuscated review (R') and the original review (R), over the set $\mathcal{F}$ of write-print features:

$D(R, R') = \sum_{i=1}^{|\mathcal{F}|} |f_i - f'_i|$

We measure the Pearson correlation between the predictions made by the LR model and the distance between the reviews. We find that the correlation is moderate and significant: $r(5352) = -0.389, p < 0.0001$, confirming our hypothesis. This points to a potential strategy for an author who wants to know whether a text obfuscated by an LLM is likely to evade author verification.

6.3 P_3: Directly Targeting Important Features

Having found significant between-author variation in obfuscation performance, we formulate a third prompt, P_3, which targets specific features in an attempt at personalization. E.g., "Rephrase the text below and increase the average word length."

We focus on four users who experience consistent obfuscation failure. We identify features that are important for identifying each author using Shapley values, which are commonly used to explain machine learning models (Hart, 1989). We select each user's top two features with highest SHAP values and prompt GPT-4 to rephrase the text and specifically change those features (P_3). We see significant improvements over GPT-4+P_1 and GPT-4+P_2 with regard to the LR AV.

This confirms that P_3 can be a viable strategy for author obfuscation even for authors who are most difficult for the commercial LLMs to obscure. However, this prompting technique based on SHAP values from the LR AV does not robustly improve performance on BERT, limiting its utility to cases in which the author has access to the target AV.

USER	BERT AV	Logistic Regression AV
User 342623 GPT-4 P3	**0.87**	**0.50**
User 342623 GPT-4 P2	0.94	0.97
User 2020269 GPT-4 P3	0.87	**0.62**
User 2020269 GPT-4 P2	**0.84**	0.98
User 1406078 GPT-4 P3	0.73	**0.48**
User 1406078 GPT-4 P2	**0.55**	0.96
User 306861 GPT-4 P3	**0.57**	**0.87**
User 306861 GPT-4 P2	0.77	1.0

Table 3: GPT-4 + P_3 obfuscation performance.

7 Conclusion

In this paper we present a study of the use of LLMs for authorship obfuscation. We analyze the performance of 3 commercial LLMs and demonstrate that LLM-based obfuscation has competitive performance with a SOTA technique, Avengers, outperforming it for some authors while generating text with higher quality and fluency.

Our analysis yields several key insights. We observe that there is significant consistency in per-user performance and feature across all three models, suggesting that these findings are reasonably robust to details of implementation and training, and to the updates that make it difficult to draw concrete conclusions based on commercial LLMs.

To address our finding that there is significant between-user variation in obfuscation performance, we propose a heuristic that can indicate whether a text is likely to evade author verification, and a prompting technique that personalizes the rephrasing to improve performance on "difficult" users.

It has become common to employ commercial LLMs for numerous NLP tasks, with varying results. We find that these models are well-suited to the task of author obfuscation, outperforming a SOTA approach. We also note that due to their popularity and accessibility, they are quite likely to be used for this purpose by vulnerable authors. It is therefore important to understand their performance on this task.

Limitation

Our work has several limitations. Firstly, we are limited by our budget for accessing Open AI's API. For that reason, we only focus on the IMDB62 dataset and only 9 users. It would be beneficial to also assess the model's performance in other datasets like the blog authorship (Schler et al., 2006) and the Extended Brennan Greenstadt Corpus (Brennan et al., 2012).

Secondly, we only focused on simple prompts to ask the models to paraphrase the texts, while there is a huge possible prompt set to select from, each focused on a different stylometric feature. We encourage future work to explore the potential of in context learning for author obfuscation purposes.

Thirdly, while the first two prompts we propose are agnostic to which AV model is opposed, the third prompt relies on SHAP values from a specific model and does not generalize well to a different model. This is a common issue in adversarial machine learning. Future work can explore other approaches to personalization that build on this one.

Fourthly, all research involving commercial LLMs is limited in the sense that the models are to a large extent black boxes, business logic plays an unknown role in their responses, and they are subject to updates and modifications at any point. However, we feel that it is worthwhile to investigate their performance for this task, since they are very likely to be used in the wild for this purpose, and do in fact perform very well.

References

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, 26(2):1–29.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Malik Altakrori, Thomas Scialom, Benjamin CM Fung, and Jackie Chi Kit Cheung. 2022. A multifaceted framework to evaluate evasion, content preservation, and misattribution in authorship obfuscation techniques. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2391–2406.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Michael Brennan, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):1–22.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Castro-Castro, Reynier Ortega Bueno, and Rafael Munoz. 2017. Author masking by sentence transformation. *CLEF (Working Notes)*, 40.

Evan Crothers, Nathalie Japkowicz, Herna Viktor, and Paula Branco. 2022. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Muhammad Haroon, Fareed Zaffar, Padmini Srinivasan, and Zubair Shafiq. 2021. Avengers ensemble! improving transferability of authorship obfuscation. *arXiv preprint arXiv:2109.07028*.

Sergiu Hart. 1989. *Shapley Value*, pages 210–216. Palgrave Macmillan UK, London.

Georgi Karadzhov, Tsvetomila Mihaylova, Yasen Kiprov, Georgi Georgiev, Ivan Koychev, and Preslav Nakov. 2017. The case for being average: A mediocrity approach to style masking and author obfuscation: (best of the labs track at clef-2017). In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8*, pages 173–185. Springer.

Yashwant Keswani, Harsh Trivedi, Parth Mehta, and Prasenjit Majumder. 2016. Author masking through translation. *CLEF (Working Notes)*, 1609:890–894.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Asad Mahmood, Faizan Ahmad, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2019. A girl has no name: Automated authorship obfuscation using mutant-x. *Proceedings on Privacy Enhancing Technologies*.

Fatemehsadat Mireshghallah and Taylor Berg-Kirkpatrick. 2021. Style pooling: Automatic text style obfuscation for improved classification fairness. *arXiv preprint arXiv:2109.04624*.

Martin Potthast, Matthias Hagen, and Benno Stein. 2016. Author obfuscation: Attacking the state of the art in authorship verification. *CLEF (Working Notes)*, pages 716–749.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. Authorship attribution with topic models. *Computational Linguistics*, 40(2):269–310.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. {A4NT}: Author attribute anonymity by adversarial training of neural machine translation. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1633–1650.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2023. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Eric Xing, Saranya Venkatraman, Thai Le, and Dongwon Lee. 2024. Alison: Fast and effective stylometric authorship obfuscation. *arXiv preprint arXiv:2402.00835*.

A Additional Experimental Results

A.1 Significantly Different Features

For each model, prompt, and feature, we run a two-sided t-test comparing the values of that feature between the original and obfuscated texts, to find which features are are significantly affected by that model/prompt rephrasing. We also compare which features are affected by the difference in prompts P_1 and P_2. Due to the large number of statistical tests, we perform a Bonferroni correction and use a strict significance threshold of $p = \frac{0.05}{1700} = 0.294e - 05$.

We present the number of significantly different features between the original set and the obfuscated set in Table 4.

Model	Between original & P_1	Between original & P_2	Between P_1 & P_2
Gemini	116	129	83
GPT-3.5	109	95	54
GPT-4	119	119	42

Table 4: Significantly different features counts between different experiments.

A.2 Feature Alteration Through Prompting

Our experiments with GPT-4 led us to observe that many stylometric features could be changed through prompting, when asked to rephrase the text and change the specific feature in it. However, some features tend to be aligned with the model's behavior for rephrasing text and could not be increased or decreased through prompting.

Feature	Prompt to Increase	Prompt to Decrease
Average word length	✓	✗
Proper noun frequency	✗	✓
Dash frequency	✓	✗
'&' frequency	✓	✗
Upper case character frequency	✓	✓
Comma frequency	✓	✗
Question mark frequency	✓	✓
Period frequency	✗	✗
Dollar sign frequency	✓	✓
Short word frequency	✗	✓
Total characters	✓	✗
Coordinating conjunctions frequency	✗	✓

Table 5: Feature changes with regard to its average value in original test set vs obfuscated test set for different users.

B Training BERT

We train Bert (base-cased) for each user separately using 1 NVIDIA A100 GPU. For each user, we trained the model on 900 reviews (810 for train and 90 for evaluation) for 3 epochs. We use Adam optimizer (Kingma and Ba, 2014) for training and we set the batch size to 16. The learning rate was set to $1e-5$. Training time for all users was less than 10 minutes. We used the model with best performance on validation for the rest of our experiments.