Proceedings of the Symposium

on

Deep Learning

University of Colorado, Colorado Springs

August 3, 2023

Editors: Jugal K. Kalita, Oluwatosin Oluwadare and
Adham Atyabi, Patrick McGuire

Funded by

National Science Foundation

Preface

It is with great pleasure that we present to you the papers describing the research performed by the NSF-funded Research Experience for Undergraduates (REU) students, who spent 10 weeks during the summer of 2023 at the University of Colorado, Colorado Springs. Within a very short period of time, the students were able to choose cutting-edge projects involving machine learning in the areas of natural language processing, bioinformatics and computational medicine; write proposals; design interesting algorithms and approaches; develop code; perform analysis; and write scholarly papers describing their findings. We hope that the students will continue working on these projects and submit papers to conferences and journals within the next few months. We also hope that it is the beginning of a fruitful career in research and innovation for all our participants.

We thank the National Science Foundation for funding our REU site. We also thank the University of Colorado, Colorado Springs, for providing an intellectually stimulating environment for research. In particular, we thank Dr. Terrance Boult, who was a helpful and stimulating mentor for the REU students. We also thank Sharon Huscher for working out all the financial and administrative details. We thank Dr. Donald Rabern, the Dean of the College of Engineering and Applied Science, and Dr. Thottam Kalkur, the Chair of the Electrical and Computer Engineering Department for their support. We also thank our students, in particular, Ali AlShami, Uma Chinta, H.M.A. Mohit Chowdhury, Steve Cruz, Timothy Flink, Melkamu Mersha, Daniel Otter, Yousra Shleibik, and Joseph Worsham for helping the students with ideas as well as with presentations on some of the latest papers, and systems and programming issues. Our gratitude to Ginger Boult for being the "REU Mom" and having the welfare of the REU interns at her heart all through the summer. Special thanks to Parker Hicks (UCCS REU 2021) of University of Colorado---Denver Anschutz Medical Campus; Wesley Robbins (UCCS REU 2021, UCCS MS 2023) heading to University of Texas, Austin for his PhD studies; and Abigail Swenor (UCCS BS 2022, UCCS REU 2021) of Notre Dame University, for taking part in panel discussions on how to apply to graduate school.

Sincerely,

Jugal Kalita
jkalita@uccs.edu

Oluwatosin Oluwadare
ooluwada@uccs.edu

Adham Atyabi
aatyabi@uccs.edu

Patrick McGuire
pmcguire@uccs.edu

Table of Contents

SimCSP: A Simple Contrastive Model for Splice Site Prediction

Kevin Stull
University of Colorado Boulder
Email: kest3869@colorado.edu

Oluwatosin Oluwadare
University of Colorado Colorado Springs
Email: ooluwada@uccs.edu

Abstract

Splice site prediction plays a vital role in the gene expression pipeline and language models have leveraged the pre-training, fine-tuning paradigm to make such predictions with great success. A weakness of traditional BERT architectures is the robustness of their internal representations, which has been addressed in human language models through the introduction of a contrastive objective function during pre-training. Hence SimCSP, a Simple Contrastive model for Splice site Prediction, is proposed. However, since the effect of contrastive learning during pre-training on splice site prediction is not well understood, a new method has been developed to investigate the connection. Which leads to the conclusion that applying a contrastive learning objective function during pre-training can improve metrics correlated with accurate classification, but that does not necessarily lead to better downstream performance after fine-tuning. The paradigmatic phenomena commonly referred to as catastrophic forgetting may provide some insight into the surprising results elucidated by this study of the SimCSP algorithm and its effects on splice site prediction.

Introduction

Accurately modelling gene expression is one of the great unsolved problems in biology (Dev 2015). DNA Splice Site Prediction (SSP) is a critical step in that pipeline that needs more robust investigation. Given the large cost of experimentally determining those locations, computational models have received a lot of attention from the scientific community. The primary drawback of such an approach is their insufficient reliability for predicting locations correctly (Chen et al. 2023).

Recently, deep learning has provided a great deal of progress in the field through two approaches, Convolutional Neural Networks (CNN)s (Akpokiro et al. 2023) and Masked Language Models (MLM)s (Yelmen and Jay 2023). CNNs leverage large swaths of labelled data to suss out the features which inform the location of splice sites (Ji et al. 2021). MLMs further leverage the abundance of data through a process called pre-training. Pre-training is a self-supervised machine learning algorithm that allows a model to create internal representations of a language through automatic labelling of an unlabelled training corpus(Erhan et al. 2010). One such architecture applied to modelling DNA is the BERT (Devlin et al. 2019) architecture. **B**iderectional **E**ncoder **R**epresentations from **T**ranformers models pre-train on large unlabelled corpora by masking some portion of their inputs then predicting how the masks should be filled in.

It has been shown that the embeddings produced by BERT architectures can be improved through the use of contrastive learning (CL) (Gao, Yao, and Chen 2022). While this technique was used specifically to improve performance on semantic similarity tasks for human language, it is unclear how transferable this is to a DNA based tasks, particularly the classification of splice sites. The SimCSE algorithm did not see an improvement in all binary classification tasks, however there was no further investigation into the rationale for this phenomena since it was not the main focus of their study. This also turned out to be the case with the TaCL (Su et al. 2021) algorithm, which introduced CL at the token level instead of at the sentence level. The TaCL algorithm saw the least improvement in binary classification which provides SimCSP with the opportunity pick up where others in the field have left off, further exploring the connection between contrastive learning and binary classification for language models.

Related Work

Traditional Machine Learning

Before the mainstream adoption of deep learning, there were several different approaches to the problem of SSP. GeneSplicer (Pertea, Lin, and Salzberg 2001), for example, used an ensemble of feature detectors and Markovian techniques to detect splice sites. In 2003, support vector machines (Zhang et al. 2003) were applied to the problem. Later on, support vector machines were combined with other techniques, like principal component analysis (Pashaei et al. 2016) for improved results.

Convolutional Neural Networks and Long Short Term Networks

Deep learning's contributions to SSP began with the application of CNNs. Models such as Deep Splicer

(Fernandez-Castillo et al. 2022), Splice2Deep (Albaradei et al. 2020), and EnsembleSplice (Akpokiro, Martin, and Oluwadare 2022) surpass traditional machine learning approaches. Other notable CNN networks include, but are not limited to, SpliceRover (Zuallaert et al. 2018) and SpliceFinder (Wang et al. 2019). All of the models mentioned can suffer from the shortcomings commonly associated with CNNs, including limited receptive fields and a propensity to over-fit during training. Long-short term networks have also been applied (Singh, Nath, and Singh 2022) and while they do address the problem of a local receptive field, vanishing and exploding gradients limit the size of the input that can be processed by the model. Further, because CNNs are highly sensitive to the training set used (Scalzitti et al. 2021), they are commonly limited to fully supervised training.

Language Models

These facts motivate the introduction language modelling to the DNA SSP problem. There have been several successful generalizations of the BERT algorithm to DNA representation and SSP (Dalla-Torre et al. 2023) (Ji et al. 2021)(Mo et al. 2021) (Cahyawijaya et al. 2022). Further, it has been shown that evolutionary and genetic information is encoded in the layers of the transformer architecture (Chen et al. 2023). These encodings can be visualized and inform what features of a nucleotide sequence are most useful for the identification of splice sites (Chen et al. 2023). It is possible that the information gained from these visualizations could be used to inform the choices made during pre-training.

Problem Statement

Put succinctly, we investigate which changes made during pre-training, due to contrastive learning (CL), affect the downstream classification result after performing fine-tuning. Expressed formally, language models have hidden layers l and a classification head c. Therefore, a simplified language model m can be represented as $m = l + c$. We let c_2 be some general method to transform the output of the language model's hidden layers into some binary classification (BC).
However, the self-supervised task which the model is pre-trained on, usually masked language modelling (MLM) or in the case of SimCSP, MLM followed by CL, uses a different classifier whose many decision boundaries divide the space into subsets which each represent one member of the pre-training task's output space. We call this classification head c_p, where p is the number of outputs for the pre-training task where it is assumed that $p > 2$. Since only l is shared between the two models, we can summarize our two models as: $m_p = l + c_p$ and $m_f = l + c_2$, where m_p is the pre-trained model and m_f is the fine-tuned model. Since $p > 2$, there is no direct metric which can compare m_p and m_f.
Assuming $f()$ is some permutation of a model, MLM, CL, BC and $e()$ is some evaluation metric; F1, accuracy, AUC. The only qualitative means of measuring how $f(m_p)$ relates to $e(m_f)$ is to transfer l to a new model where a c_2 classification head can be fit to the downstream task of F1 score. Which is feasible when only fitting c, but becomes restrictive when l is also fine-tuned to the downstream task, as is commonly the case with language models.
Given that there exists some evaluation metrics $e^*()$ which can be applied directly to l. How does $f_{CL}(m_p)$ affect $f_{BC}(m_f)$ and is $e^*(l)$ sufficient to predict the relationship between them?

Approach

Dataset

For pre-training, all chromosomes of the primary assembly GRCh38/hg38 were used, the data set was loaded using The Nucleotide Transformer's (Dalla-Torre et al. 2023) HuggingFace train split. This is an unlabelled data set (with respect to SSP) containing DNA sequences from humans. For fine-tuning, the Spliceator data set (Scalzitti et al. 2021) is used, the code used to process and load the data set were obtained from the github page of the SpliceBERT paper (Chen et al. 2023). The data set contains DNA sequences of varying length, 400 or 600, that are labelled as a non-splice site, an acceptor site, or a donor sites. The acceptor donor distinction is not used as this study is interested in binary classification. That gives a final data set which contains 400 nucleotide sequences that are labelled as either 0 non-splicing or 1 splicing sites.
For the purpose of evaluating SimCSP in an self-supervised setting, the Spliceator data set was re-arranged into a new data set called Spliceator for Semantic Similarity (S3). In this new data set, the labelled sequences are randomly paired together without replacement. Then, if the elements share a label (both are splice sites or both are not splice sites), they are given a new label of 1, otherwise, the pair is given a label of 0. Each pair is considered a single training example of sequences that are (0) not semantically similar, or (1) semantically similar. Using the SCCS metric described in greater detail in the Evaluation Metrics section, this new data set, S3, can be used to evaluate the model's understanding of SSP during pre-training and during fine-tuning.
When bench-marking SimCSP for comparison with other methods, the zebra fish, fruit fly, worm, and arabidopsie were used.

Theoretical Foundations

A contrastive loss function is used for the pre-training of SimCSP. It is functionally identical to the one introduced for SimCSE (Gao, Yao, and Chen 2022), which was used for the unsupervised contrastive learning of sentence embeddings. If we let x_i be one of the inputs in a batch of N inputs to the model. Then $\hat{x}_i$ and $\bar{x}_i$ are two embeddings produced by that same input x_i to the encoder with two different dropout masks. A dropout mask is a shorthand term to describe the dropout applied throughout a standard BERT architecture. For readability's sake, subscripts are not applied to each dropout mask but it is assumed that no two are identical to one another. Then the contrastive loss is given by the expression:

$$loss(\hat{x}_i) = -\log \frac{\exp(sim(\hat{x}_i, \bar{x}_i))}{\sum_{j=1}^{N} \exp(sim(\hat{x}_i, \bar{x}_j))} \quad (1)$$

Where $\text{sim}(x_1, x_2)$ is defined as the cosine similarity between two vectors. That is:

$$sim(x_1, x_2) = \frac{x_1^T x_2}{x_1 \cdot x_2} \quad (2)$$

In application, the Multiple Negatives Ranking Loss function is used from the sentence transformers library (Reimers and Gurevych 2019), where positive pairs are generated by taking the same sequence twice with different dropout masks and other sequences are assumed to be negative samples.

Evaluation Metrics

The foremost metric used to evaluate SimCSP is Splice Site Prediction (SSP) F1 score (SSP). ROC AUC is also used to validate the models during fine-tuning. These metrics can only be applied to labelled data, thus they cannot be used to directly quantify the changes caused by CL during pre-training, for that reason, other metrics are introduced.

Two metrics will be introduced, Normalized Mutual Information (NMI) and Spearman Correlation with Cosine Similarity (SCCS). The possibility that either of these serve as a proxy for F1 is investigated. These metrics have been selected because they can be directly applied to this hidden layers of the network without the need to pass through a classification head.

The Spearman Correlation with Cosine Similarity (SCCS) of the last layer's [CLS] token is used as one measure of how similar the model "believes" two sequences are. Cosine similarity can be used to evaluate both the fine-tuned model, and the pre-trained model that it is based on.

The authors of SpliceBERT (Chen et al. 2023) utilized the UMAP technique to visualize their nucleotide embeddings, then used the Leiden algorithm to cluster them. The SpliceBERT embeddings displayed a better degree of separability across coding and non-coding DNA inputs, quantified by the plot's higher Normalized Mutual Information (NMI). Better separability indicates that the its internal representation of a splice site is more robust. CL is applied to the model during pre-training, and its affect on F1 score is studied.

Architectural Characteristics

The SimCSP framework uses Contrastive Learning (CL) to investigate the pre-training of DNA Language Models (LM)s and its effects on SSP. It is used as an additional layer of pre-training after convergence is reached with Masked Language Modelling (MLM). This gives:

SimCSP Architecture

1. Pre-train MLM
2. Pre-train CL
3. Fine-tune SSP

In practice, the SpliceBERT-510.nt-human pre-trained model (Chen et al. 2023) is loaded and it's parameters are modified using Contrastive Learning (CL). A learning rate of $1 * 10^{-4}$ is used with a batch size of 512 and a weight decay of $1 * 10^{-6}$. There are 6 transformer blocks with a hidden size of 512 and 16 self-attention modules per block. The [CLS] token is used as the input to the classifier.

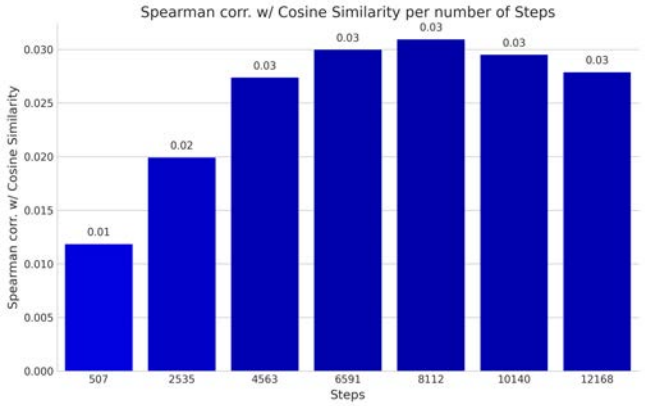


Figure 1: The Spearman correlation with cosine similarity of the [CLS] token with semantically similar and dissimilar validation examples from the Spliceator data set given varying amounts of pre-training.

Model Evaluation

The metrics used to evaluate SimCSP are SCCS, NMI, ROC AUC, and F1 score. The SCCS and NMI metrics can be applied to the hidden layers of the model during pre-training and fine-tuning. The best pre-trained model is selected using the Human Reference Genome (Dalla-Torre et al. 2023) for NMI and the Spliceator data set (Scalzitti et al. 2021) training split for SCCS. The benchmarks used to compare model performance across different architectures is only used for inference and inference was only performed once by the best model. The ROC AUC and F1 scores can only be used during fine-tuning, therefore the ROC AUC was used to score the models using the validation split of the Spliceator data set. The best model was chosen by taking the highest F1 score on the testing split of the Spliceator data set.

Results

Effect of Contrastive Learning on the [CLS] Token

When Contrastive Learning (CL) is applied to the [CLS] token of the DNA Language Model (DNA LM), the effects can be seen in Figure 1. As pre-training progresses, the Spearman Correlation with Cosine Similarity (SCCS) increases until reaching a peak at around 8,112 steps before slowing falling back off. The scale of the change is also worth noting, while model performance does more than double, the numerical distance in performance between the least and most performant model is around 2%.

Effect of Contrastive Learning on the NMI of SimCSP's Layers

Figure 2 is the highest NMI score for SimCSP and occurs after 2,535 batches of CL during pre-training. The plot is from the 4^{th} transformer layer supporting the result of SpliceBERT (Chen et al. 2023), which is that the $2^{nd} - 5^{th}$ layers of the network are the most informative for the prediction of splice sites. This is further supported by Figure 3, which shows the average NMI across a differing number of

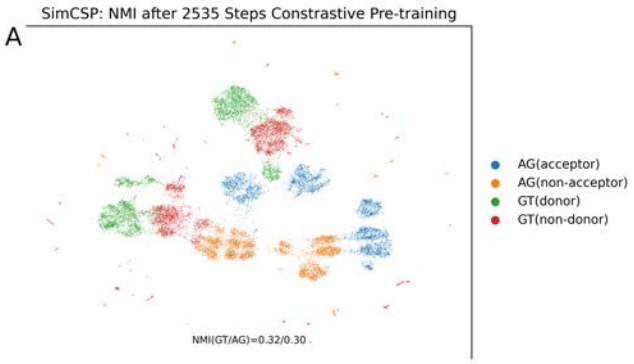


Figure 2: The NMI of the 4^{th} layer of SimCSP after 2,535 steps of Contrastive Learning during pre-training.

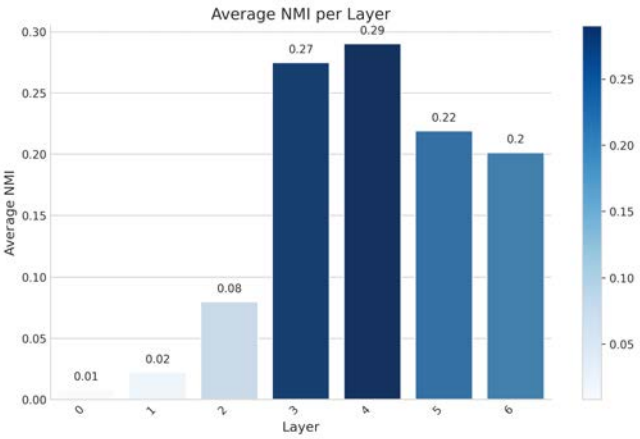


Figure 3: The average NMI score of each layer averaged across varying amounts of pre-training given a contrastive loss function.

training steps by layer. It suggests that most of the semantic information relating to SS are located in 3^{rd} and 4^{th} layers of the model. It is clear that in the long run, the NMI score decreases as more contrastive learning is introduced. However, there is a local peak early on in the epoch that is higher than the starting point, which is used as the best NMI pre-trained model.

Given the previous results, the 4^{th} layer of the model is analyzed in greater detail. Upon inspection, Figure 4 shows a general trend downward as more CL is introduced during pre-training. However, there is a small increase in NMI at 2535 pre-training steps.

Effect of Fine-tuning on NMI and SCCS

When the pre-trained model is fine-tuned, we observe a slight increase in NMI of the plots of the embeddings and a substantial increase in the model's performance on the SCCS metric. With NMI increasing from 0.13 to 0.16 and SCCS increasing from 0.03 to 0.80 on average.

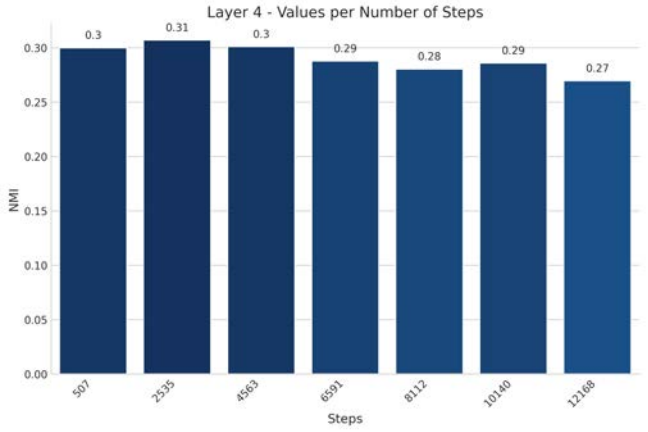


Figure 4: The NMI score of the 4^{th} transformer layer of the SimCSP architecture given varying amounts of pre-training with a CL loss function.

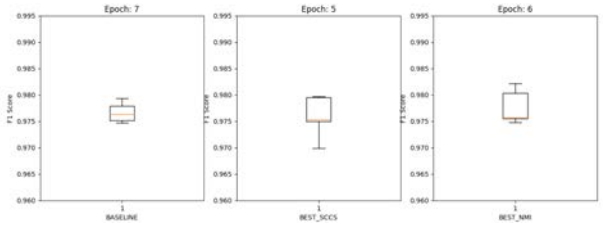


Figure 5: The F1 scores of the Baseline Pre-trained model, the best SCCS from pre-training and the best NMI from pre-training.

F1 Scores of Pre-trained models with best performance on SCCS and NMI

From figure 5, notice that box plots F1 scores of the baseline SpliceBERT-human model are all relatively close, with best NMI skewed towards being slightly better than baseline while best SCCS is skewed towards being slightly worse than baseline. The mean of the baseline model is 0.9766, the mean of the best SCCS model is 0.9759, and the mean of the best NMI model is 0.9777, which means that all models fall within 0.3% of one another in terms of performance.

Discussion

NMI and SCCS are impacted by Fine-tuning

While it is clear that SimCSP has marginal effect on the NMI and SCCS scores during pre-training. It is clear that fine-tuning plays a role in both. This implies that they can serve as proxies for F1 score in a setting where fine-tuning is not practical. While the pattern of slightly increased SCCS and NMI scores due to SimCSP is consistent, it is quite minor. One possible explanation can be provided when considering the differences between MLM and CL.

Contrastive Learning occurs at the Feature Level but MLM happens at the Token Level

CL seeks to improve the grouping of hidden features within the representation space of a model. MLM however, seeks to create generalized relationships between tokens. It is possible that these two objectives are not completely amicable to one another. CL seems to be improving the hidden classes of the representation space at the expense of token-level information. This phenomenon where the model learns new information but forgets old information is referred to as Catastrophic Forgetting (CF). It is possible that CF may be taking place during the training of SimCSP, which would explain why it is possible to optimize parameters associated with better F1 scores, while simultaneously, producing a model that is the same as or worse at its downstream task. CF helps inform why the best models, according to NMI and SCCS, occur after so few steps of CL. It could be the case that 1,000 steps of CL is a local minima where the most utility can be gained from CL before too much is lost due to CF.

Conclusion

Language models are a promising new technique for studying many facets of gene expression, including the prediction of splice sites. Even if a metric can be strongly correlated to better SSP, simply improving that metric by a method such as CL, as was the case for SimCSP, is not sufficient to endow a guaranteed improvement in the downstream performance of the language model. Introducing a new form of learning to a network can also lead to forgetting of information that is necessary for the downstream task of Splice Site Prediction. SimCSP reveals that there are no free lunches when training a deep learning model, each lesson comes at a price. This deeper understanding of contrastive learning's relationship to splice site prediction is crucial if the scientific field is going to produce robust DNA language models.

Acknowledgements

The work in this paper is supported by the National Science Foundation under grant No. 2050919. Any opinions, findings, conclusions, or recommendations expressed in this work do not necessarily reflect the views of the National Science Foundation.

References

Akpokiro, V.; Chowdhury, H. M.; Olowofila, S.; Nusrat, R.; and Oluwadare, O. 2023. Cnnsplice: Robust models for splice site prediction using convolutional neural networks. *Computational and Structural Biotechnology Journal*.

Akpokiro, V.; Martin, T.; and Oluwadare, O. 2022. EnsembleSplice: ensemble deep learning model for splice site prediction. *BMC Bioinformatics* 23(1):413.

Albaradei, S.; Magana-Mora, A.; Thafar, M.; Uludag, M.; Bajic, V. B.; Gojobori, T.; Essack, M.; and Jankovic, B. R. 2020. Splice2Deep: An ensemble of deep convolutional neural networks for improved splice site prediction in genomic DNA. *Gene* 763:100035.

Cahyawijaya, S.; Yu, T.; Liu, Z.; Mak, T. T.; Zhou, X.; Ip, N. Y.; and Fung, P. 2022. Snp2vec: Scalable self-supervised pre-training for genome-wide association study. *arXiv preprint arXiv:2204.06699*.

Chen, K.; Zhou, Y.; Ding, M.; Wang, Y.; Ren, Z.; and Yang, Y. 2023. Self-supervised learning on millions of pre-mRNA sequences improves sequence-based RNA splicing predictio. Technical report. Type: article.

Dalla-Torre, H.; Gonzalez, L.; Mendoza Revilla, J.; Lopez Carranza, N.; Henryk Grywaczewski, A.; Oteri, F.; Dallago, C.; Trop, E.; Sirelkhatim, H.; Richard, G.; et al. 2023. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv* 2023–01.

Dev, S. B. 2015. Unsolved problems in biology—the state of current thinking. *Progress in Biophysics and Molecular Biology* 117(2-3):232–239.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs].

Erhan, D.; Courville, A.; Bengio, Y.; and Vincent, P. 2010. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 201–208. JMLR Workshop and Conference Proceedings.

Fernandez-Castillo, E.; Barbosa-Santillán, L. I.; Falcon-Morales, L.; and Sánchez-Escobar, J. J. 2022. Deep Splicer: A CNN Model for Splice Site Prediction in Genetic Sequences. *Genes* 13(5):907. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Gao, T.; Yao, X.; and Chen, D. 2022. SimCSE: Simple Contrastive Learning of Sentence Embeddings. arXiv:2104.08821 [cs].

Ji, Y.; Zhou, Z.; Liu, H.; and Davuluri, R. V. 2021. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* 37(15):2112–2120. _eprint: https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/39622303/btab083.pdf.

Mo, S.; Fu, X.; Hong, C.; Chen, Y.; Zheng, Y.; Tang, X.; Shen, Z.; Xing, E. P.; and Lan, Y. 2021. Multi-modal Self-supervised Pre-training for Regulatory Genome Across Cell Types. arXiv:2110.05231 [cs, q-bio].

Pashaei, E.; Yilmaz, A.; Ozen, M.; and Aydin, N. 2016. A novel method for splice sites prediction using sequence component and hidden markov model. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3076–3079. IEEE.

Pertea, M.; Lin, X.; and Salzberg, S. L. 2001. GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Research* 29(5):1185–1190.

Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Scalzitti, N.; Kress, A.; Orhand, R.; Weber, T.; Moulinier, L.; Jeannin-Girardon, A.; Collet, P.; Poch, O.; and Thompson, J. D. 2021. Spliceator: multi-species splice site prediction

using convolutional neural networks. *BMC Bioinformatics* 22(1):561.

Singh, N.; Nath, R.; and Singh, D. B. 2022. Splice-site identification for exon prediction using bidirectional LSTM-RNN approach. *Biochemistry and Biophysics Reports* 30:101285.

Su, Y.; Liu, F.; Meng, Z.; Lan, T.; Shu, L.; Shareghi, E.; and Collier, N. 2021. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

Wang, R.; Wang, Z.; Wang, J.; and Li, S. 2019. SpliceFinder: ab initio prediction of splice sites using convolutional neural network. *BMC Bioinformatics* 20(23):652.

Yelmen, B., and Jay, F. 2023. An Overview of Deep Generative Models in Functional and Evolutionary Genomics. *Annual Review of Biomedical Data Science* 6(1):null. _eprint: https://doi.org/10.1146/annurev-biodatasci-020722-115651.

Zhang, X. H.; Heller, K. A.; Hefter, I.; Leslie, C. S.; and Chasin, L. A. 2003. Sequence information for the splicing of human pre-mrna identified by support vector machine classification. *Genome Research* 13(12):2637–2650.

Zuallaert, J.; Godin, F.; Kim, M.; Soete, A.; Saeys, Y.; and De Neve, W. 2018. SpliceRover: interpretable convolutional neural networks for improved splice site prediction. *Bioinformatics* 34(24):4180–4188.

HiCForecast: Dynamic Network Optical Flow Estimation Algorithm for Spatiotemporal Hi-C Data Forecasting

Dmitry Pinchuk
University of Wisconsin-Madison
500 Lincoln Dr
Madison, WI 53706
dpinchuk@wisc.edu

Oluwatosin Oluwadare
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918
ooluwada@uccs.edu

Abstract

The evolution of the 3D chromosome structure in time (*4D nucleome*) plays a crucial role in time-dependent processes in the cell. Reconstruction of the 3D genome and 4D nucleome is dependent on experimentally obtained Hi-C data. Due to the sparsity of data it is important to be able to forecast Hi-C data at future time-points from time-series Hi-C data. This study uses a dynamic network optical flow estimation video prediction algorithm to forecast spatiotemporal Hi-C data. The best variation of this model achieves validation predictions with Pearson correlation and HiCRep about 1 percent below that of the only existing model for this problem.

1. Introduction

Chromosome 3D structure is of vital interest to biologists studying the relationships between chromosome structure and gene regulation, expression and transcription. Current methods use experimentally obtained high throughput chromosome conformation capture (Hi-C) data (Lieberman-Aiden et al. 2009) to reconstruct the 3D structure of a chromosome (Oluwadare, Highsmith, and Cheng 2019). The Hi-C method obtains the frequency of contact between different loci, which are fragments of DNA corresponding to a gene. This contact data is represented by an $n \times n$ Hi-C contact matrix, where n is the number of loci in a chromosome and the ij-th entry is the number of contacts between loci i and j in the chromosome.

Hi-C data and the 3D structure reconstructed from it are obtained for a specific point in time. Many biological processes in a cell are time dependent and analyzing 3D chromosome structure as it evolves in time or the *4D nucleome* is crucial to understanding them (Di Stefano et al. 2021). 4D structure analysis is dependent on the availability of Hi-C data at different time points from which 3D data is reconstructed; however, due to the sparsity of data it is important to be able to forecast future Hi-C data points from Hi-C data in previous time points.

2. Related Work

Several studies have explored the interpolation of 3D chromosome structures between two given time points including TADdyn by Di Stefano et al. (2020) and 4DMax by Highsmith and Cheng (2021). However, there is currently only one research effort that focuses on forecasting future Hi-C data based on Hi-C time series. The HiC4D method, introduced by Liu and Wang (2023), treats Hi-C contact matrices as frames of a video and employs a Long Short-Term Memory (LSTM) based video prediction algorithm. Specifically, Liu and Wang developed the ResConvLSTM model by adding residual skip connections between ConvLSTM (Shi et al. 2015) layers. Their study demonstrates the superior performance of ResConvLSTM compared to ConvLSTM, ST-LSTM (Wang et al. 2017), SimVP (Gao et al. 2022), and a naive network video prediction algorithm.

3. Approach

This study uses a dynamic neural network video prediction model to forecast spatiotemporal Hi-C data. The training and evaluation of the model are done using the same datasets and metrics as in the HiC4D study.

3.1 Model

This study uses the Dynamic Multi-Scale Voxel Flow Network (DMVFN) (Hu et al. 2023) video prediction algorithm to predict future Hi-C contact data from a series of existing time-frames. It takes the frames at time points t_2 and t_3 as inputs and predicts the frames at time points t_4, t_5 and t_6. The model consists of MVFB blocks that estimate optical flow, which is the pixel-wise motion between frames. An MVFB block takes the ouptut of a previous MVFB block together with two input frames to synthesize the next frame and estimate the optical flow. The model has 9 MVFB blocks each of which scales the input by some factor. A Routing Module adaptively selects with of these blocks will be included in the model for a particular input. The final estimate of the optical flow from the MVFB blocks together with the first two images is used to reconstruct the next frame. DMVFN is currently state of the art on the Cityscapes (Cordts et al. 2016), KITTI (Geiger, Lenz, and Urtasun 2012) and DAVIS 2017 (Pont-Tuset et al. 2017) datasets for video prediction.

3.2 Data

The Gene Expression Omnibus (GEO) and Genome Sequence Archive (GSA) databases are used to acquire the following spatiotemporal Hi-C datasets of mouse and human cells during embryogenesis:

- GEO GSE82185 is from preimplantation mouse embryos contributed by Du et al. (2017). The time points correspond to the gamete, zygote, early 2-cell, late 2-cell, 8-cell, inner cell masses and stem cell stages. The last six stages are utilized in this study.
- GSA PRJCA000241 contributed by Ke et al. (2017) is also from mouse embryos corresponding to the gamete, early embryo, 2-cell, 4-cell, 8-cell, embryonic day (E)3.5 and E7.5 stages. Again only the last six time points are used.
- GEO GSE146001 contributed by Chen et al. (2020) is Hi-C data taken from somatic cell nuclear transfer (SCNT) mouse embryos. The 12hpa, early 2-cell, late 2-cell, 8-cell, ICM and TE stages are taken as time points from this dataset.
- GSA CRA000852 contributed by Chen et al. (2019) contains Hi-C data from human embryogenesis with the sperm, 2-cell, 8-cell, morula, blastocysts, and six-week-old embryo stages taken as the time points.

The first dataset (GEO GSE82185) is used for training, validation and testing. Chromosome 19 is used for validation, chromosomes 2 and 6 are used for testing and the remaining chromosomes are used for training. The entirety of the data from the remaining three datasets will be used for testing. At this stage of the study only the first dataset has been used.

3.3 Implementation Details

The models were implemented using PyTorch (Paszke et al. 2019). We used the AdamW (Loshchilov and Hutter 2017) optimizer. The models were trained according to a cosine annealing schedule with the learning rate decaying from 10^{-4} to 10^{-5}. Models were trained on patches with dimensions $32 \times 32, 48 \times 48, 64 \times 64, 80 \times 80$ and 96×96 with batches of 8, 16, 64 and 256. Although the first three time steps are available to the model, the model only takes time steps 2 and 3 as input.

Loss Functions Given input frames I_{t-1} and I_t the i-th MVFB block outputs a prediction of the next frame $\tilde{I}^i_{t+1}$. The loss functions used has the following general framework:

$$L = \sum_{i=1}^{n} 0.8^{n-i} d(\tilde{I}^i_{t+1}, I_{t+1}) + \alpha L_{\text{VGG}}(\tilde{I}_{t+1}, I_{t+1}),$$

where $n = 9$ is the amount of MVFB blocks, L_{VGG} is the VGG loss (Ledig et al. 2017), d was taken to be either the l_1 loss, MSE loss or the l_1 loss on the Laplacian pyarmid representations (Paris, Hasinoff, and Kautz 2011) as is default in DMVFN, and the parameter α is either 0 or 0.5.

Data Normalization The distribution of the values in each chromosome in the training dataset (Figure 1) shows that the majority of the values are below 100, which is the default normalization in the HiC4D study. The distribution of the values in Figure 1 is heavily skewed towards values in the 0-50 range, which occur less frequently at lower genomic distances. Figure 2 shows that the majority of the values at lower genomic distances are under 400. Due to these heuristics, the study used both 100 and 400 for normalization of the data. Additionally this study used normalization by 255. When doing normalization by 100 or 400 both at inference and training 100 or 400 were set as the cut off for values in the data, but this was not done when normalizing by 255.

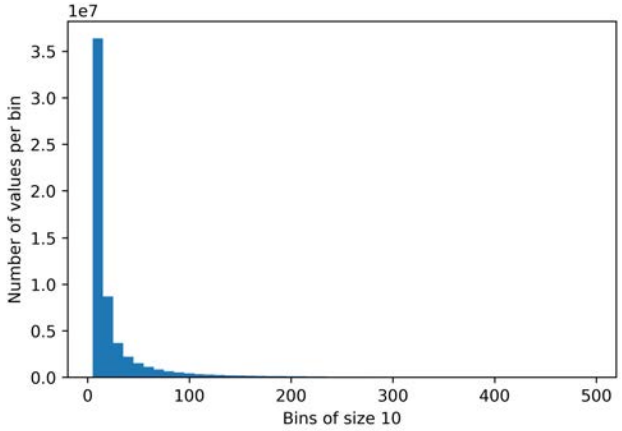


Figure 1: Histogram with amount of values ranging from 0 to 500 in the 96×96 training dataset averaged over the 17 chromosomes in the training set.

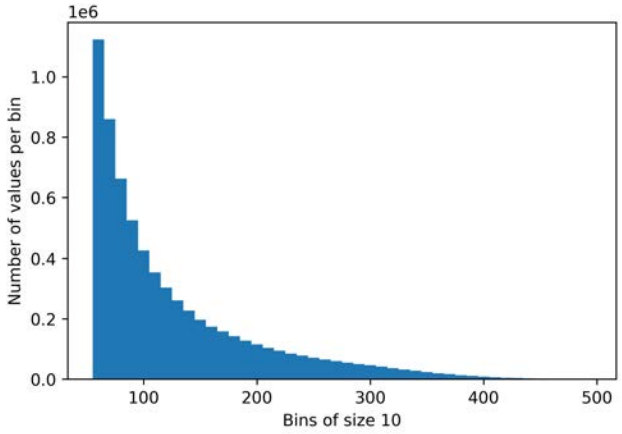


Figure 2: Histogram with amount of values ranging from 50 to 500 in the 96×96 training dataset averaged over the 17 chromosomes in the training set.

3.4 Evaluation Metrics

The performance of the algorithm is evaluated using the Pearson correlation coefficient with the ground truth at each genomic distance between bins 0 and 35 with resolution 40kb. We used the stratum-adjusted correlation coefficient from HiCRep (Yang et al. 2017) with lower bound genomic distance between loci set to 400 000 bases and upper bound set to 1 600 000, and the smoothing parameter was set to 5. We also used HiCRep with same smoothing parameter and upper bound, but with the lower bound set to 40 000 bases.

Patch Size	Loss	Pearson Average (0-35)			HiCRep (40k-1600k)			HiCRep (400k- 1600k)		
		t_4	t_5	t_6	t_4	t_5	t_6	t_4	t_5	t_6
(HiC4D) 50	MSE	0.6928	0.6763	0.6775	0.8372	0.7828	0.7514	0.8549	0.7780	0.7416
96	Default no VGG	0.6802	0.6641	0.6608	0.8239	0.7696	0.7251	0.8048	0.7338	0.6891
96	MSE no VGG	0.6628	0.6401	0.5912	0.7977	0.7367	0.5899	0.7906	0.7042	0.6158
64	Default no VGG	0.6538	0.6279	0.6211	0.7808	0.7027	0.6607	0.7637	0.6620	0.6202

Table 1: Pearson correlation and HiCRep metrics between the ground truth and predictions made by models with various hyperparameter combinations on validation chromosome 19. The first row of data pertains to the HiC4D model against which the other models are compared. The default loss is the l_1 loss evaluated at the Laplacian pyramid representations.

4. Results

The validation results for the best three models trained with various hyperparameter combinations are displayed in Table 1. The best model is only about a percentage away from HiC4D accoring to the Pearson correlation coefficient. For timesteps t_3 and t_4 the best model is also about one percentage point away from HiC4D using HiCRep with the lower bound at 40 000 bases. The best model is about 5 percent lower than HiC4D according to HiCRep with lower bound set to 400 000. The best models all turned out to have a batch size of 8 and normalization of 255. They were trained for 50 epochs.

5. Conclusion

The only existing study on forecasting Hi-C data used an LSTM based video prediction algorithm to solve the problem. This study adapts the state of the art DMVFN model. The current models are close to achieving state of the art results. Blindly testing on the test chromosomes and on other datasets will be reserved until hyperparameter search and model modification are complete, and HiCForecast will improve upon HiC4D in all timsteps using HiCRep. Solving the Hi-C data forecasting problem will increase the availability of Hi-C data, which will lead to a better understanding of the 4D structure of the chromosome because existing methods reconstruct the 3D structure from expensive Hi-C data.

6. Acknowledgements

This work is supported by the National Science Foundation under Grant No. 2050919. Any opinion, finding, or conclusion in this study is that of the authors and does not necessarily reflect the views of the National Science Foundation.

References

Chen, X.; Ke, Y.; Wu, K.; Zhao, H.; Sun, Y.; Gao, L.; Liu, Z.; Zhang, J.; Tao, W.; Hou, Z.; et al. 2019. Key role for ctcf in establishing chromatin structure in human embryos. *Nature* 576(7786):306–310.

Chen, M.; Zhu, Q.; Li, C.; Kou, X.; Zhao, Y.; Li, Y.; Xu, R.; Yang, L.; Yang, L.; Gu, L.; et al. 2020. Chromatin architecture reorganization in murine somatic cell nuclear transfer embryos. *Nature Communications* 11(1):1813.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Di Stefano, M.; Stadhouders, R.; Farabella, I.; Castillo, D.; Serra, F.; Graf, T.; and Marti-Renom, M. A. 2020. Transcriptional activation during cell reprogramming correlates with the formation of 3d open chromatin hubs. *Nature communications* 11(1):2564.

Di Stefano, M.; Paulsen, J.; Jost, D.; and Marti-Renom, M. A. 2021. 4d nucleome modeling. *Current Opinion in Genetics & Development* 67:25–32.

Du, Z.; Zheng, H.; Huang, B.; Ma, R.; Wu, J.; Zhang, X.; He, J.; Xiang, Y.; Wang, Q.; Li, Y.; et al. 2017. Allelic reprogramming of 3d chromatin architecture during early mammalian development. *Nature* 547(7662):232–235.

Gao, Z.; Tan, C.; Wu, L.; and Li, S. Z. 2022. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3170–3180.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Highsmith, M., and Cheng, J. 2021. Four-dimensional chromosome structure prediction. *International Journal of Molecular Sciences* 22(18):9785.

Hu, X.; Huang, Z.; Huang, A.; Xu, J.; and Zhou, S. 2023. A dynamic multi-scale voxel flow network for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6121–6131.

Ke, Y.; Xu, Y.; Chen, X.; Feng, S.; Liu, Z.; Sun, Y.; Yao, X.; Li, F.; Zhu, W.; Gao, L.; et al. 2017. 3d chromatin structures of mature gametes and structural reprogramming during mammalian embryogenesis. *Cell* 170(2):367–381.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690.

Lieberman-Aiden, E.; Van Berkum, N. L.; Williams, L.; Imakaev, M.; Ragoczy, T.; Telling, A.; Amit, I.; Lajoie, B. R.; Sabo, P. J.; Dorschner, M. O.; et al. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science* 326(5950):289–293.

Liu, T., and Wang, Z. 2023. Hic4d: Forecasting spatiotemporal hi-c data with residual convlstm. *Briefings in Bioinformatics* bbad263.

Loshchilov, I., and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Oluwadare, O.; Highsmith, M.; and Cheng, J. 2019. An overview of methods for reconstructing 3-d chromosome and genome structures from hi-c data. *Biological procedures online* 21(1):1–20.

Paris, S.; Hasinoff, S. W.; and Kautz, J. 2011. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph.* 30(4):68.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc. 8024–8035.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28.

Wang, Y.; Long, M.; Wang, J.; Gao, Z.; and Yu, P. S. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems* 30.

Yang, T.; Zhang, F.; Yardımcı, G. G.; Song, F.; Hardison, R. C.; Noble, W. S.; Yue, F.; and Li, Q. 2017. Hicrep: assessing the reproducibility of hi-c data using a stratum-adjusted correlation coefficient. *Genome research* 27(11):1939–1949.

ScanChIP-P: A Clustering Approach to Identifying Topologically Associated Domains From HiChIP

Ashley Doerfler
Oregon State University - Cascades
1500 SW Chandler Ave
Bend, Oregon 97702
doerflas@oregonstate.edu

Oluwatosin Oluwadare
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918
ooluwada@uccs.edu

Abstract

Chromatin conformation capture finds information about the three-dimensional organization of chromosomes in the nucleus. Conformation capture is often sequenced for the entire genome, but recent studies have localized their methods to capture conformation centered around proteins. Protein-centric data permits lower cost to process matrix data displaying identified interactions and higher resolution results. There are numerous callers that are used to detect topologically associated domains (TADs) from high-throughput chromosome conformation capture data sets. Topologically associated domains are linked to gene expression, transcription, and functionality of the genome across mammalian species. HiChIP, a protein-centric chromatin conformation method. Using HiChIP, can provide higher resolution data used to identify binding patterns of proteins to DNA. ScanChIP is a proposed method to identify TAD boundaries from HiChIP. ScanChIP implements DBSCAN, a cluster machine learning algorithm to identify clusters of interactions as TADs.

Introduction

Three dimensional conformation of chromatin can help identify and describe functionality of a genome (Sexton et al. 2007). Methods of chromosome conformation capture (3C) paired with sequencing methods that advances our ability to separate functional elements, as well as find relationships between chromatin structures, gene activity, and functional state (Lieberman-Aiden et al. 2009). In the past, fluorescence in situ hybridization (FISH) was used to visualize specific DNA sequences (Price 1993). Research advanced to 3C, a high-throughput chromosome conformation capture method. It is used to analyze the spacial organization of chromosomes (Dekker et al. 2002). While 3C can only apply to specific regions of the genome, Hi-C can capture interactions genome-wide (Belton et al. 2012). In Hi-C, biotin labels are used in ligation in order to have selective purification of chimeric DNA ligation (Belton et al. 2012). A heatmap matrix that plots normalized interaction values is used to observe pairwise interactions (Belton et al. 2012). These interactions run diagonally across the matrix (Belton et al. 2012). This study revealed the compartmentalization of genomic regions (Belton et al. 2012).

The neighborhoods of interaction described in the Hi-C study was later named Topologically associated domains (TADs). TADs are chromatin regions where intra-interactions take place (Dixon et al. 2012). The frequency of interaction is graphed on a heatmap matrix. TAD boundaries are identified by looking at frequency of interactions between regions of the genome (Fig. 1) (Dixon et al. 2012). A study in 2012 found that these domains are packed with the insulator binding proteins CTCF, housekeeping genes, transfer RNA's, and short interspersed element (SINE) retrotransposons (Dixon et al. 2012). CTCF is a chromatin architectural protein found in high concentrations at TAD boundaries (Hyle et al. 2023). It serves a role in transcriptional regulation by being both a transcriptional activator and repressor (Hyle et al. 2023). TAD boundaries are rich in CTCF which stops spread of heterochromatin (Dixon et al. 2012). CTCF acts as a boundary for TADs, and they facilitate interactions between transcription regulatory sequences (Ong and Corces 2014). Topologically associated domains are important in gene expression, transcription, and functionality across mammalian species (Dixon et al. 2012). Disruption of TAD boundaries has a correlation developmental and psychiatric diseases (Lupiáñez et al. 2015) (Halvorsen et al. 2020) (Krijger and De Laat 2016). While there were great advancement in chromatin functionality, sequencing of the entire genome and identifying their boundaries can become computationally expensive and yield low resolution solutions.

Rather than focusing on the entire genome, studies have worked on centering their data collections on proteins (Collas 2010) (Fullwood et al. 2009) (Li et al. 2010) (Li et al. 2017). ChIP is a technique where a selected protein is immunoprecipitated from chromatin to determine associated DNA sequences (Zheng et al. 2007). A different strategy to analyze chromatin interaction that was based off of ChIP, ChIA-PET, was introduced in 2009 (Fullwood et al. 2009). ChIA-PET is a protein-focused study. In ChIA-PET, proximity ligation connects DNA linkers to Tethered DNA fragments, and from there, pair end tags (PETs) are extracted for sequencing (Fullwood et al. 2009). It offered a different mapping strategy approach for analyzing chromatin interactions using paired end tag sequencing. HiChIP aimed to improve protein based chromatin conformation capture. Like ChIA-PET, HiChIP extracts pair end sequencing for

sequencing, and DNA proteins are enriched with ChIP in both methods (Fullwood et al. 2009)(Mumbach et al. 2016). HiChIP is a protein conformation capture method is also inspired by Hi-C (Mumbach et al. 2016). This method can analyze the 3D structure of a genome while also finding binding patterns to DNA (Mumbach et al. 2016).

Hi-C Based TAD Detection Methods

The discovery of Hi-C in a 2009 study, later led to the discovery of TADs (Lieberman-Aiden et al. 2009) (Wang, Cui, and Peng 2017). Since, there have been numerous algorithms that aim to increase the capability and quality of the process including HiTAD, CaTCH, OnTAD, ClusterTAD, CASPIAN as well as many others (Wang, Cui, and Peng 2017) (Zhan et al. 2017) (An et al. 2019) (Oluwadare and Cheng 2017) (Gong et al. 2022). HiTAD reduces impact of genomic distance by enriching the intra-domain interaction and inter-domain interaction frequencies and utilizes recursion to optimize detection (Wang, Cui, and Peng 2017). ClusterTAD is an example of a caller that specifically uses a cluster algorithm (Oluwadare and Cheng 2017). Kmeans is used in ClusterTAD, but this there are other kinds of clustering algorithms including density-based clustering. CASPIAN is another example of a clustering TAD caller. It uses HDBSCAN to the clustering process because it doesn't require any parameters. HDBSCAN can be used to have an undefined amount of clusters, and it aims to identify areas where points are close together within a given space (Gong et al. 2022). TADs similarly create dense regions of intra-interactions in chromatin.

3C With Protein of Interest

Rather than focusing on the entire genome, other callers use data sets that are centered around specific protein structures. Mango was popular analysis pipeline for ChIA-PET that calculates statistical confidence estimation of interactions (Phanstiel et al. 2015). A more recent study, published in 2023, created a model called HPTAD. These data sets are both specifically used for high resolution enhancer-promoter interaction detection (Rosen et al. 2023). HPTAD is a method used to detect TADs in HiChIP and PLAC-seq data. The method has a statistical approach where a regression model is used to identify topologically associated domains (Rosen et al. 2023). Their goal wasn't to make a better TAD caller, but rather create a caller for Hi-ChIP data. Rather than the input data residing in an $N \times N$ contact matrix, such as Hi-C, the normalized file is in BEDPE format to identify pair end tags.

Approach

TADs have CTCF at their boundaries to contain interactions locally, therefore large groups of interaction frequency should be physically close. ScanChIP-P takes in a normalized contact frequency matrix to identify topologically associated domains by clustering contact frequency together.

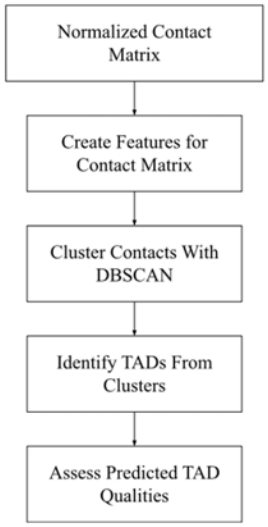


Figure 1: ScanChIP-P workflow.

Prepare the normalized contact matrix

HPTAD's pipeline was used was used to prepare a normalized contact matrix intended to be used in this study (Rosen et al. 2023). The normalized data created from the pipeline was converted into a symmetrical contact matrix for our purposes (Figure 2).

3240000	3280000	3320000	3360000	3400000	3440000
0	1.49493108260814	0.83024177529267	1.0862717817118	0.917955137966109	1.05326784345106
1.49493108260814	0	0.903477276274877	1.18166653287152	1.30243535485259	1.80057555504553
0.83024177529267	0.903477276274877	0	1.08584952430497	1.17432747596433	1.62288471807686
1.0862717817118	1.18166653287152	1.08584952430497	0	1.15060691177623	1.46878785818808
0.917955137966109	1.30243535485259	1.17432747596433	1.15060691177623	0	0.941848442751056
1.05326784345106	1.80057555504553	1.62288471807686	1.46878785818808	0.941848442751056	0
1.04476002168305	1.42402691954495	1.39838916734587	1.23336771183887	0.743881252067915	1.20623679277113
0.646291918438249	1.08500709061693	1.02916020900442	1.28626483637705	0.621023914566258	0.812429736159543
0.772280853250778	0.860131689328986	1.08361103402078	1.14529134219569	0.919291528408021	1.1494978877011
0.24038961709641 2	0.725255850660894	0.867450807314803	0.970589098044366	0.780401058512701	1.08647113558045
0.511915945955311	0.953835556162734	1.2574467425986	0.869566233912375	1.02594092932602	0.719712049867533

Figure 2: Normalized data retrieved from HPTAD's normalization formatted into a symmetric matrix.

In order to have proof of concept, this study has used a 30x30 contact matrix from ClusterTAD before we moved to a larger set. The matrix is not symmetrical, therefore is was reflected the bottom left data to the top to obtain a symmetrical matrix we could use to replicate HP data more accurately (Figure 3).

Create features for contacts

There was three different approaches used to do so. The first feature extraction method collected each feature in an L-shape from a specified window W (Figure 4 a). The window would move down along the diagonal $M[i, j]$ to collect a total of N features the size of $N/W \times 2$. The idea of the window was to ignore some of the unneeded noisy data by limiting the view of the features. There was an issue where the window would extend past the data when the algorithm

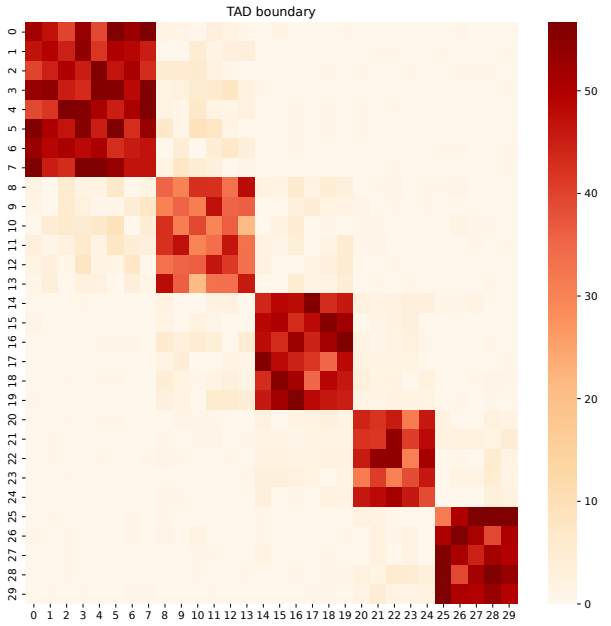


Figure 3: Heat map of the symmetrical contact matrix derived from ClusterTAD's 30x30 matrix.

reaches the diagonal $M[i, j]$ that extends past the length of $N - N/W$. In order to compensate for this on our models, we began reading the data in reverse from the diagonal point.

Another approach this study took was collecting each feature in a square-shape (Figure 4 b). This approach will allow the features to hold more information about the data around the diagonal $M[i, j]$. It will also use a window such as the L-shaped feature extraction has, but each feature will contain every row in the window making it $N/W \times N/W$ numbers long.

Comparing these extractions with the feature extraction method used in ClusterTAD (Oluwadare and Cheng 2017), our last method creates a cross like extraction within a window where $M[i, j]$ is the center data point. Similarly to the L-shaped extraction, the window will shift down with the diagonal to collect a total of N features the length of $N/W \times 2$ (Figure 4 c). To accommodate for sections of the diagonal that cannot fit the diagonal to the center, the window will shift right and downward if it is in the first half of the data, and it will shift left and up if it is in the second half. The Diagonal is kept as central as possible in the process. Once the features are extracted, they are ready to fit to DBSCAN's clustering algorithm.

Clustering using DBSCAN

This study intends to implement a cluster DBSCAN, a density based machine learning algorithm. This algorithm is a simple implementation that includes two parameters including a given neighborhood's radius and a minimum number of neighboring points is required to be considered a core point. The minimum number of neighbors and what distance warrants being considered a neighbor are both user defined.

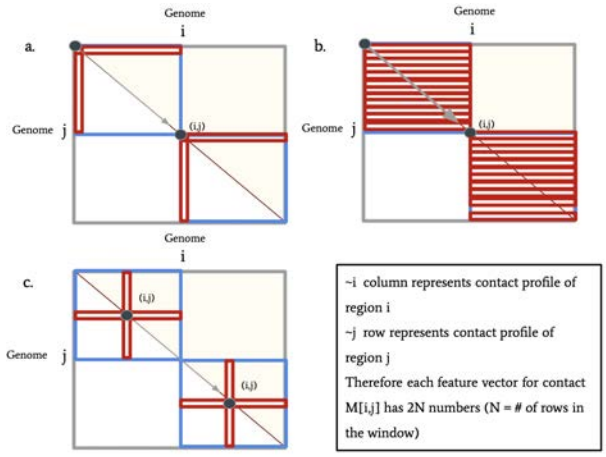


Figure 4: a. L-shaped feature extraction method. b. Square-shaped feature extraction method. c. Cross-shaped feature extraction method.

The number of neighbors is required for clustering the contact interactions on the genome to identify TAD boundaries. This is estimated by finding how many bins are required to reach the minimum size of a topologically associated domain. Since this is an estimation, we added a feature that evaluates the calculated value ± 1.

At first, the esp, or size of a neighborhood is calculated by approximating the elbow of the K-Nearest Neighbor function. The graphs provided had little to no elbow, so instead a wide range of epsilon values were tested and the clusters created were evaluated using silhouette score to determine the best results.

Identify TADs

The TADs in the data are identified using the same concept that was used to determine the minimum points required for DBSCAN. The length of each cluster was calculated and compared to the minimum amount of bins required to reach the minimum size of a topologically associated domain using the labels created from DBSCAN. Once the algorithm reaches the last label, if there are enough consecutive labels equivalent to it, the cluster will be identified as the last TAD.

TAD quality

The quality of the discovered topologically associated domains is measured with Rand index and Fowlkes-Mallows score. These libraries are used to evaluate agreement of two clusters. They were specifically selected to compare to CASPIAN's data.

Results

Currently, the features attempted are not showing very promising results. We began by testing each feature extraction approach using DBSCAN. First, an L-shaped feature extraction was performed within varying windows of data. This yielded poor results with an average silhouette score of 0.348, and unexpectedly, smaller windows [1/8, 1/9, or 1/10

of the data set] provided a higher silhouette score of 0.628. The clusters identified were at 0 to 5, 8 to 11, 13 to 17, 20 to 22, and 24 to 29 (fig 5 a).

The results from the L-shaped features were not ideal, so a square shape was tested out and similar results were concluded. Contrary to the L-shaped feature extraction, using DBSCAN with the square feature was only able to produce TADs in windows of 1/8, 1/9, or 1/10 of the data. The clusters produced were the same as the square-shaped features (Figure 5 a).

The last feature was a cross-like figure. The cross is used with a window. The idea was that it is similar to the feature extraction method used in ClusterTAD where the entire row and column at the diagonal was collected. This method still gave no promising results with the best window being 1/7 of the 30x30 matrix it was tested on and a lower silhouette score of 0.541 (Figure 5 b).

ClusterTAD's feature extraction had clear clusters at 0 to 7, 8 to 13, 14 to 19, 20 to 24, and 25 to 29, and this feature clustering had a silhouette score of 0.821 and a quality score of 44.140 on all three trials testing each clustering algorithm (Figure 5 c). Based on these findings, the next step is to slowly shorten the window to find when it begins to lose enough information that the TADs cannot be identified.

As a last effort using DBSCAN, we took another approach with the cross features. Instead of creating a window size with a ratio, we tried subtracting from N. The clusters identified were 0 to 7, 8 to 13, 20 to 24, and 25 to 29 with a quality score of 44.097. While this was slightly better, we wanted to explore if we could do better (Figure 5 d).

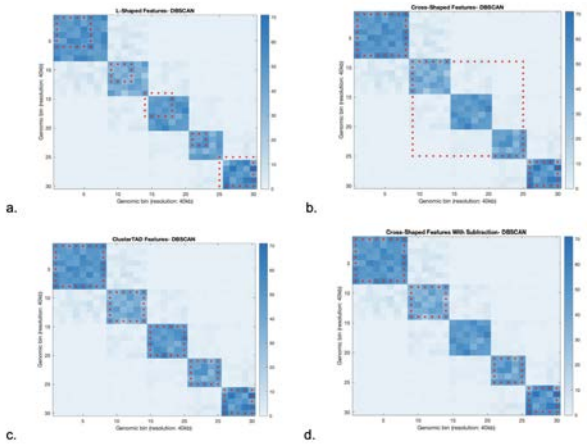


Figure 5: a. Predicted TADs from DBSCAN fitting to L-Shaped features. The dotted-red lines indicates the TAD boundaries. Square-shaped feature extraction yielded the same results. b. Predicted TADs from DBSCAN fitting to Cross-Shaped features using a ratio to find the window size. c. Predicted TADs from DBSCAN fitting to ClusterTAD features. d. Predicted TADs from DBSCAN fitting to Cross-Shaped features using subtraction to get the window size.

Our novel features retrieved less than ideal results with DBSCAN. In order to ensure our results were the problem,

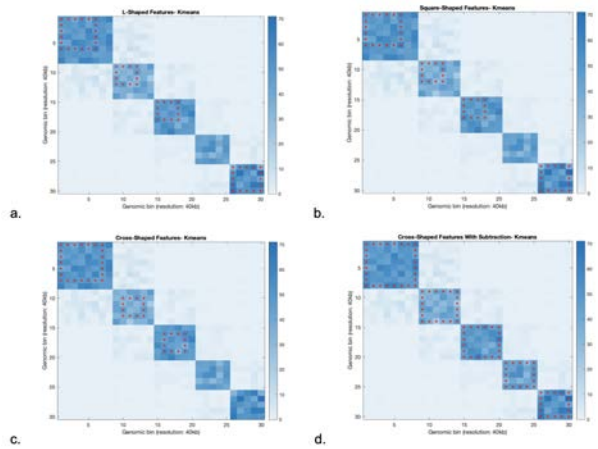


Figure 6: a. Predicted TADs from Kmeans fitting to L-Shaped features. The dotted-red lines indicates the TAD boundaries. b. Square-shaped feature extraction predicted TAD boundaries c. Predicted TADs from Kmeans fitting to Cross-Shaped features using a ratio to find the window size. d. Predicted TADs from Kmeans fitting to Cross-Shaped features using subtraction to get the window size.

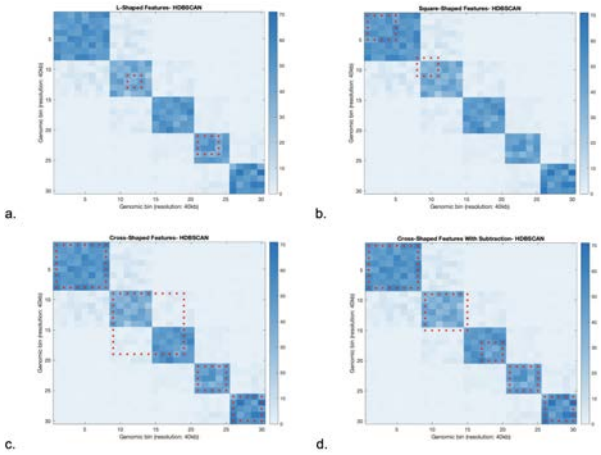


Figure 7: a. Predicted TADs from HDBSCAN fitting to L-Shaped features. The dotted-red lines indicates the TAD boundaries. b. Square-shaped feature extraction predicted TAD boundaries c. Predicted TADs from HDBSCAN fitting to Cross-Shaped features using a ratio to find the window size. d. Predicted TADs from HDBSCAN fitting to Cross-Shaped features using subtraction to get the window size.

we continued to retest each algorithm by following pursuit of ClusterTAD and CASPIAN by using Kmeans and HDBSCAN respectively (Figure 6 and 7).

Kmeans was able to accurately identify the TADs from the cross data that used subtraction to create the window size. While accurate, it may be noted that the maximum amount that the window could be reduced by is 4 data points, and it was most optimal when only subtracting two data points from each end of the window. Even at its most optimal, this method had a lower quality score than ClusterTAD at 0.751 (Figure 6 d). The other feature methods were able to get a general direction of where the TADs were but either failed to grab the entirety of each TAD or simply left out a TAD.

HDBSCAN was unable to identify any of the TADs correctly from any of the novel features created. Many of the TADs were missing from the L-shaped and square-shaped feature data, but the TADs that were identified were in the correct general area (Figure 7 a and b). Both cross-shaped features were close, but got the boundaries slightly off (Figure 7 c and d).

Conclusion

Topologically associated domains are a crucial piece to understand genome expression and regulation. There are many models that detect TADs within Hi-C data, but it can become rather costly and reduce result resolution. HiChIP data offers bright prospects for enhancing our knowledge of genome regulation in specific proteins and identifying chromosomes affected disease. There aren't many algorithms for protein specific 3C. Hence, in this work, we propose a TAD detection model for HiChIP that implements a DBSCAN cluster algorithm. A cluster algorithm can identify and group together intra-interactions to identify TAD boundaries. While DBSCAN has bright prospects, this study needs to continue to investigate the the best novel way to create feature extractions in order to enhance the quality of our TAD detection algorithm.

Schedule

June 5, 2023	Introduction and finish pre-proposal
June 9, 2023	Finalize proposal and Present
June 15, 2023	Run Reference Code
July 7, 2023	Complete Midsummer Report
July 30, 2023	Implement small scale model
August 3, 2023	Present Findings

Acknowledgements

The work in this paper is supported by the National Science Foundation under grant No. 2050919. Any opinions, findings, conclusions, or recommendations expressed in this work do not necessarily reflect the views of the National Science Foundation.

References

An, L.; Yang, T.; Yang, J.; Nuebler, J.; Xiang, G.; Hardison, R. C.; Li, Q.; and Zhang, Y. 2019. Ontad: hierarchical domain structure reveals the divergence of activity among tads and boundaries. *Genome biology* 20(1):1–16.

Belton, J.-M.; McCord, R. P.; Gibcus, J. H.; Naumova, N.; Zhan, Y.; and Dekker, J. 2012. Hi–c: a comprehensive technique to capture the conformation of genomes. *Methods* 58(3):268–276.

Collas, P. 2010. The current state of chromatin immunoprecipitation. *Molecular biotechnology* 45:87–100.

Dekker, J.; Rippe, K.; Dekker, M.; and Kleckner, N. 2002. Capturing chromosome conformation. *science* 295(5558):1306–1311.

Dixon, J. R.; Selvaraj, S.; Yue, F.; Kim, A.; Li, Y.; Shen, Y.; Hu, M.; Liu, J. S.; and Ren, B. 2012. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485(7398):376–380.

Fullwood, M. J.; Liu, M. H.; Pan, Y. F.; Liu, J.; Xu, H.; Mohamed, Y. B.; Orlov, Y. L.; Velkov, S.; Ho, A.; Mei, P. H.; et al. 2009. An oestrogen-receptor-α-bound human chromatin interactome. *Nature* 462(7269):58–64.

Gong, H.; Yang, Y.; Zhang, X.; Li, M.; Zhang, S.; and Chen, Y. 2022. Caspian: A method to identify chromatin topological associated domains based on spatial density cluster. *Computational and Structural Biotechnology Journal* 20:4816–4824.

Halvorsen, M.; Huh, R.; Oskolkov, N.; Wen, J.; Netotea, S.; Giusti-Rodriguez, P.; Karlsson, R.; Bryois, J.; Nystedt, B.; Ameur, A.; et al. 2020. Increased burden of ultra-rare structural variants localizing to boundaries of topologically associated domains in schizophrenia. *Nature communications* 11(1):1842.

Hyle, J.; Djekidel, M. N.; Williams, J.; Wright, S.; Shao, Y.; Xu, B.; and Li, C. 2023. Auxin-inducible degron 2 system deciphers functions of ctcf domains in transcriptional regulation. *Genome Biology* 24(1):1–30.

Krijger, P. H. L., and De Laat, W. 2016. Regulation of disease-associated gene expression in the 3d genome. *Nature reviews Molecular cell biology* 17(12):771–782.

Li, G.; Fullwood, M. J.; Xu, H.; Mulawadi, F. H.; Velkov, S.; Vega, V.; Ariyaratne, P. N.; Mohamed, Y. B.; Ooi, H.-S.; Tennakoon, C.; et al. 2010. Chia-pet tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. *Genome biology* 11:1–13.

Li, X.; Luo, O. J.; Wang, P.; Zheng, M.; Wang, D.; Piecuch, E.; Zhu, J. J.; Tian, S. Z.; Tang, Z.; Li, G.; et al. 2017. Long-read chia-pet for base-pair-resolution mapping of haplotype-specific chromatin interactions. *Nature protocols* 12(5):899–915.

Lieberman-Aiden, E.; Van Berkum, N. L.; Williams, L.; Imakaev, M.; Ragoczy, T.; Telling, A.; Amit, I.; Lajoie, B. R.; Sabo, P. J.; Dorschner, M. O.; et al. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science* 326(5950):289–293.

Lupiáñez, D. G.; Kraft, K.; Heinrich, V.; Krawitz, P.; Brancati, F.; Klopocki, E.; Horn, D.; Kayserili, H.; Opitz, J. M.; Laxova, R.; et al. 2015. Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell* 161(5):1012–1025.

Mumbach, M. R.; Rubin, A. J.; Flynn, R. A.; Dai, C.; Khavari, P. A.; Greenleaf, W. J.; and Chang, H. Y. 2016. Hichip: efficient and sensitive analysis of protein-directed genome architecture. *Nature methods* 13(11):919–922.

Oluwadare, O., and Cheng, J. 2017. Clustertad: an unsupervised machine learning approach to detecting topologically associated domains of chromosomes from hi-c data. *BMC bioinformatics* 18(1):1–14.

Ong, C.-T., and Corces, V. G. 2014. Ctcf: an architectural protein bridging genome topology and function. *Nature Reviews Genetics* 15(4):234–246.

Phanstiel, D. H.; Boyle, A. P.; Heidari, N.; and Snyder, M. P. 2015. Mango: a bias-correcting chia-pet analysis pipeline. *Bioinformatics* 31(19):3092–3098.

Price, C. 1993. Fluorescence in situ hybridization. *Blood reviews* 7(2):127–134.

Rosen, J.; Lee, L.; Abnousi, A.; Chen, J.; Wen, J.; Hu, M.; and Li, Y. 2023. Hptad: A computational method to identify topologically associating domains from hichip and plac-seq datasets. *Computational and Structural Biotechnology Journal* 21:931–939.

Sexton, T.; Schober, H.; Fraser, P.; and Gasser, S. M. 2007. Gene regulation through nuclear organization. *Nature structural & molecular biology* 14(11):1049–1055.

Wang, X.-T.; Cui, W.; and Peng, C. 2017. Hitad: detecting the structural and functional hierarchies of topologically associating domains from chromatin interactions. *Nucleic acids research*.

Zhan, Y.; Mariani, L.; Barozzi, I.; Schulz, E. G.; Blüthgen, N.; Stadler, M.; Tiana, G.; and Giorgetti, L. 2017. Reciprocal insulation analysis of hi-c data shows that tads represent a functionally but not structurally privileged scale in the hierarchical folding of chromosomes. *Genome research* 27(3):479–490.

Zheng, M.; Barrera, L. O.; Ren, B.; and Wu, Y. N. 2007. Chip-chip: Data, model, and analysis. *Biometrics* 63(3):787–796.

ASPECT: Alternative SPlicing Events Classification with Transformer

Miguelangel Tamargo
Florida International University
11200 SW 8th St,
Miami, FL 33199
mtamargo028@fiu.edu

Oluwatosin Oluwadare
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy,
Colorado Springs, CO 80918
ooluwada@uccs.edu

Abstract

Alternative splicing (AS) is a biological process that rearranges distinct segments of pre-mRNA, resulting in a variety of transcripts. These transcripts are subsequently translated into diverse proteins from a single gene. Dysregulation of AS in human diseases can lead to the generation of abnormal protein isoforms, which can disrupt cellular functions and contribute to disease progression. Historically, AS research has primarily focused on two prevalent events: alternatively skipped exons (exon cassettes) and constitutively spliced exons. However, there is a significant gap in the methods available to classify a broader range of AS events beyond these two types. Current "splicing codes" leverage convolutional neural networks (CNNs) to analyze and classify AS events. While other approaches have attempted to tackle similar problems, they have become outdated, relying on models such as Support Vector Machines (SVMs) and Convolutional Neural Networks (CNN). Recent advancements in deep learning, particularly the development of EfficientNets, have significantly improved the efficiency of traditional CNNs. These advancements present an opportunity to challenge and potentially surpass the current state-of-the-art AS classification models. Furthermore, the application of transformer models, renowned for their performance in various tasks, could provide a novel approach to AS classification. This research aims to explore these possibilities and push the boundaries of AS classification.

Introduction

The process of alternative splicing (AS) involves the manipulation and rearrangement of pre-mRNA exons and introns to create a transcriptional code for proteins. The code is determined by the arrangement of the exons as they are spliced back together. Remarkably, AS can lead to the production of up to 95% of human genes, each with varying structures and roles [1]. Moreover, AS has been implicated in 15% of hereditary diseases and cancers [2]. The identification and understanding of splice sites are crucial in this context, and through these deeper understandings is how machine learning models can better drive the advancements in personalized medicine.

Convolutional Neural Networks (CNNs) have gained significant attention in genomics applications, particularly in predicting regulatory sequences [5]. These architectures perform convolutional operations on input data and utilize pooling to reduce the size of the input data. Layers in CNNs are fully connected, with each neuron in a layer connected to the neurons in neighboring layers. This structure allows the input data to be classified by comparing it to a set of known or similar classes [7].

Historically, models like that of Busch and Hertel have employed Support Vector Machines (SVMs) trained on four types of AS events: a set of constitutive exons, a set of exons having an alternative 3' splice site, a set of exons with an alternative 5' splice site, and a set of cassette exons [3] This particular method focused on the binary classification aspect and was state-of-the-art for its time. The use of the personally created database of HEXEvent helped also push the research of AS events further by streamlining the data collection and filtration from the UCSC Genome Browser.

However, recent advancements in deep learning have introduced more efficient and accurate models for classification tasks. One such advancement in the advances of CNNs has shown that compounding scaling can create a balanced distribution of computational resources, leading to improved accuracy without significantly increasing computational requirements [6]. Another advancement is the use of transformer models, which have become state-of-the-art in various classification and prediction tasks involving NLP. Transformer models, such as the Bidirectional Encoder Representations from Transformers (BERT), leverage self-attention mechanisms and can be pre-trained on large amounts of data, making them particularly effective for complex tasks [9].

Applying these advancements, combined with transformer models like BERT, could potentially revolutionize the classification of AS events and push the boundaries of our understanding of genomics.

Related Work

CNN-Based Algorithms

Deep Splice Code (DSC) has been built on convolution neural networks (CNN) using methods presented in the Deep Motif Dashboard (DeMo Dashboard)[8] in order to train its data set through DNA sequences through the extraction of competitive alternative splice sites as well as motifs of important splicing factors[5]. The DSC CNN architecture uti-

lized convolutional blocks which were divided into two architectures. Each convolutional block contained 3 convolutional layers [5]. The sizes of the layers filter went as follows: The first convolutional layer contained 32 filters and a window size of 7 units. The second convolutional layer contained 8 filters with a window size of 4. The final convolutional layer contained 8 filters and a window size of 3, depending on the comparison data, determining the layer size of 2 or 3. The model is trained and tested using the k-fold cross-validation technique with a k size of 10. Totaling the number of models trained to 40 models trained for each comparison model [5].

In relation to DSC, we have a CNN architecture known as EfficientNet. Based on ResNet architecture EfficientNet increased performance by introducing a technique known as compound scaling [6]. Another introduced technique to EfficientNet is known as AutoML. Using machine learning to automatically search for the best network architecture to achieve state-of-the-art results [6]. EfficientNet works on the relationship of compound scaling implemented into the architecture of traditional CNN architectures. Compound scaling uniformly scales all dimensions of depth, width resolution, rather than the traditional which scales arbitrarily. Using a grid search to understand the dimensions relationship coefficient. This scaling coefficient of the dimensions mentioned above is applied to the baseline network in order to reach the target model size or computational budget. The applied architecture EffiicentNet B0 (EFB0) uses AutoML, which can apply a mobile inverted bottleneck convolution (MBConv) followed by each dimension being scaled. This family of modules is termed EfficientNet[6].

Other work on the topic was performed by Hertel and Busch with their work on a ”splicing code” as they termed it. Used a Support Vector Machine (SVM) to classify AS. They also used 2 different architectures as well. Their first 3 models utilized a binary classification and then tweaked this model for their fourth which was multi-classification. Its performance compared to DSC portrays why CNN is vastly a superior model. Both used the same database compilating the most up-to-date dataset from the HEXEvent database consisting of the four types of AS events we’re classifying.

Transformer-Based Algorithms

Transformer-based models, such as the Bidirectional Encoder Representations from Transformers (BERT), have shown significant promise in various fields, including genomics. These models leverage self-attention mechanisms and can be pre-trained on large amounts of data, making them particularly effective for complex tasks.

One such application in genomics is DNABERT, a pre-trained model that has been used to predict promoter regions driving gene expression directly from sequences without using any structural or biological signals [13]. DNABERT has also been used to create a refined foundation model, DNABERT-2, which employs an efficient tokenizer and multiple strategies to overcome input length constraints, reduce time and memory expenditure, and enhance capability [12].

Another notable transformer-based model in genomics is GENA-LM, a suite of transformer-based foundational language models capable of handling input lengths up to 36 base pairs. GENA-LM has been used to fine-tune complex biological questions with modest requirements and has shown performance either matching or exceeding prior models, whether task-specific or universal [11].

In the context of pathogenic viruses, the COVID-DeepPredictor, a deep learning framework based on the Long Short Term Memory Recurrent Neural Network, has been used to identify unknown sequences of these pathogens. This model uses the k-mer technique to create a Bag-of-Descriptors (BoDs) in order to generate a Bag-of-Unique-Descriptors (BoUDs) vocabulary, subsequently preparing an embedded representation for given sequences. The COVID-DeepPredictor has shown superior results over state-of-the-art techniques based on Linear Discriminant Analysis, Random Forests, and the Gradient Boosting Method [10].

Problem Statement

The classification of alternative splicing (AS) events is a complex task that has been traditionally addressed using models trained on constitutive exons, alternative 3’ splice sites (SS), alternative 5’ SS, and alternatively skipped exons. These models have primarily employed Support Vector Machines (SVMs) and traditional Convolutional Neural Network (CNN) architectures. However, with the advent of more computationally efficient models and the availability of larger datasets, there is an opportunity to improve the accuracy of AS event classification beyond what is currently achievable.

The goal of this research is to classify 5’ and 3’ splice events and possibly intron retention with a high degree of accuracy. To achieve this, we propose to leverage the power of transformer-based models, specifically the pre-trained DNABert model, and follow the pipeline of the DeepPredictor approach. This will allow us to expand upon the current state-of-the-art methods and later branch to more complex models like DNABert-2 and the Nucleotide Transformers from InstaDeepAI, which was pre-trained on a 2.5 billion parameter model [14].

Approach

Our approach to improving the classification of alternative splicing (AS) events involves leveraging the power of transformer-based models, particularly the pre-trained DNABert model. We propose to follow a pipeline similar to that of the COVID-DeepPredictor, which has shown significant success in identifying unknown sequences of pathogenic viruses.

The COVID-DeepPredictor employs a deep learning framework based on the Long Short-Term Memory Recurrent Neural Network. It uses the k-mer technique to create a Bag-of-Descriptors (BoDs) in order to generate a Bag-of-Unique-Descriptors (BoUDs) vocabulary, subsequently preparing an embedded representation for given sequences [10].

In our proposed model, which we term ASPECT, we will adapt this pipeline to the task of AS event classification. We will train our model using the dataset from DSC and Bosch and Hertel [4][5], and utilize the EfficientNet CNN architecture for feature extraction and hyperparameter search, which has shown promise in uniformly scaling all dimensions of depth, width, and resolution.

To evaluate the performance of our proposed method, we will compare it against several other models. These include the SVM method, the Deep Splice Code (DSC) method, and more advanced models like DNABert-2, LM-Gena, and the Nucleotide Transformers from InstaDeepAI, which was pretrained on a 2.5 billion parameter model [14] All three of these alternative models utilized Byte BPE k-mers and improved upon DNABert. This will be used as the benchmark when measuring the performance of other models.

The performance of these models will be analyzed based on the Area Under the Curve (AUC) metric. Still, it will also us f1 metric to evaluate our models performance to that of DeepPredictor. Through this comparative analysis, we aim to demonstrate the potential of transformer-based models in improving the classification of AS events.

Formulas

The Area Under the Curve (AUC) for the Receiver Operating Characteristic (ROC) curve is a measure of the model's ability to discriminate between positive and negative samples. It is calculated using the trapezoidal rule:

AUC-ROC = $\int_0^1 TPR(FPR)\, d(FPR)$

Where TPR is the True Positive Rate (Sensitivity) and FPR is the False Positive Rate (1-Specificity).

The Usage Level (usageLevel) formula, which represents the proportion of transcripts that utilize a specific alternative splicing event, can be calculated as follows:

usageLevel = number of transcripts using the alternative event$\overline{total number of transcripts for the gene}$

PhastCons scores are used to measure the evolutionary conservation of a genomic region. The PhastCons scores formula calculates a conservation score for a given genomic position, typically derived from multiple sequence alignments:

PhastCons score = -$\log(likelihood of conservation)$

$= -\log\left(\frac{number of conserved sequences}{total number of sequences}\right)$

The Maximum Entropy Score (MES) formula is used to predict splice site strength, providing a score for a given splice site sequence. It is based on the principle of maximum entropy:

MES = -$\sum_{i=1}^{n} P_i \log_2(P_i)$

These additional formulas provide relevant metrics and scores for evaluating alternative splicing events, aiding in the assessment of the ASPECT model's performance and its contribution to advancing the classification of alternative splicing.

Results

In addition to the preliminary results mentioned above, ongoing trials for fine-tuning the ASPECT model have shown encouraging improvements over the current state-of-the-art methods. Notably, one of the significant shortcomings observed in the existing state-of-the-art methods, such as the Support Vector Machines (SVM) method and the Deep Splice Code (DSC) method, was their inefficient data distribution. Specifically, these methods exhibited an extremely skewed imbalance of labels, which can pose significant challenges in binary classification tasks.

In the context of imbalanced data, traditional machine learning algorithms may tend to favor the majority class, leading to inflated accuracy metrics that do not accurately reflect the model's true performance on unseen data. The ASPECT model's approach, leveraging transformer-based models and the EfficientNet architecture, aims to mitigate these imbalances and provide more reliable and meaningful performance metrics.

The ASPECT model's ability to efficiently distribute and handle imbalanced data has shown promising potential in the classification of AS events. By employing the power of transformer-based models, the model is better equipped to capture intricate patterns and dependencies within the data, contributing to improved classification accuracy.

It is worth noting that while the current results are promising, they are part of an ongoing research effort toward refining the ASPECT model. We acknowledge the importance of comprehensive testing and evaluation to ensure the model's robustness and generalization capabilities. As the research progresses, we anticipate that further experimentation and refinement will lead to more definitive findings and potentially make significant contributions to advancing the state-of-the-art in AS event classification.

In summary, the early findings from our ASPECT model suggest promising improvements over existing methods, particularly in addressing data imbalances. By incorporating transformer-based models and EfficientNet, we aim to develop a robust and efficient classification tool for AS events that can provide valuable insights into the complexities of alternative splicing mechanisms. Continued research and thorough evaluation will be pivotal in realizing the full potential of the ASPECT model and its applicability in genomics and personalized medicine.

article multirow

Metrics	Classification Tasks		
	Multi-Classification	3' vs. 5'	Const vs 3'
Avg F1	0.408	0.573	0.9422
Avg Acc	0.472	0.584	0.961
Test F1	0.382	0.554	0.927
Test Acc	0.479	0.683	0.935

Table 1: ASPECT - Results Table

Conclusion

In conclusion, our study explores the potential of the proposed ASPECT model, which combines Convolutional Neural Networks (CNNs) and transformer-based architectures like EfficientNet, for classifying alternative splicing (AS) events. Leveraging advanced machine learning techniques and robust hardware resources, our research aims to refine the accuracy of current state-of-the-art methods for AS event classification.

The preliminary results are promising, demonstrating improvements over existing methods. However, further model evolution and fine-tuning are ongoing to ensure robustness and generalization across diverse datasets.

The ability of the ASPECT model to efficiently handle imbalanced data is a significant advantage in binary classification tasks. By leveraging transformer-based models, we aim to capture intricate patterns and dependencies within the data, leading to improved classification accuracy.

Our research underscores the transformative potential of advanced machine learning techniques in understanding complex biological processes, such as alternative splicing. Accurate classification of AS events holds immense promise in medical diagnostics, potentially revealing previously unknown associations between AS events and diseases. This advancement can revolutionize personalized medicine, where a deeper understanding of AS events may lead to tailored therapies for individual patients.

As we proceed with further experimentation and refinement, we anticipate more conclusive results validating the effectiveness and robustness of the ASPECT model. A thorough evaluation of the model's performance on diverse datasets is crucial to ensure its applicability and reliability in real-world scenarios.

In conclusion, our research contributes to the state-of-the-art in alternative splicing event classification, enriching the field of genomics and opening new avenues for personalized and precision medicine. We envision reshaping the future of healthcare through advancements in machine learning and genomics, ultimately benefiting patient outcomes and healthcare as a whole.

Acknowledgments

The work in this paper is supported by the National Science Foundation under grant No. 2050919. Any opinions, findings, conclusions, or recommendations expressed in this work do not necessarily reflect the views of the National Science Foundation.

References

Q. Pan, O. Shai, L. J. Lee, B. J. Frey, and B. J. Blencowe, "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing," *Nature genetics*, vol. 40, no. 12, pp. 1413–1415, 2008.

Y. Marquez, J. W. Brown, C. Simpson, A. Barta, and M. Kalyna, "Transcriptome survey reveals increased complexity of the alternative splicing landscape in arabidopsis," *Genome research*, vol. 22, no. 6, pp. 1184–1195, 2012.

Y. Cui, M. Cai, and H. E. Stanley, "Comparative analysis and classification of cassette exons and constitutive exons," *BioMed Research International*, vol. 2017, 2017.

A. Busch and K. J. Hertel, "Splicing predictions reliably classify different types of alternative splicing," *RNA*, vol. 21, no. 5, pp. 813–823, 2015.

Z. Louadi, M. Oubounyt, H. Tayara, and K. T. Chong, "Deep splicing code: Classifying alternative splicing events using deep learning," *Genes*, vol. 10, no. 8, p. 587, 2019.

M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

J. Lanchantin, R. Singh, B. Wang, and Y. Qi, "Deep motif dashboard: visualizing and understanding genomic sequences using deep neural networks," in *Pacific Symposium on Biocomputing 2017*, pp. 254–265, World Scientific, 2017.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

I. Saha, N. Ghosh, D. Maity, A. Seal, and D. Plewczynski, "Covid-deeppredictor: recurrent neural network to predict sars-cov-2 and other pathogenic viruses," *Frontiers in genetics*, vol. 12, p. 569120, 2021.

V. Fishman, Y. Kuratov, M. Petrov, A. Shmelev, D. Shepelin, N. Chekanov, O. Kardymon, and M. Burtsev, "Gena-lm: A family of open-source foundational models for long dna sequences," *bioRxiv*, pp. 2023–06, 2023.

Z. Zhou, Y. Ji, W. Li, P. Dutta, R. Davuluri, and H. Liu, "Dnabert-2: Efficient foundation model and benchmark for multi-species genome," *arXiv preprint arXiv:2306.15006*, 2023.

X. Wang, X. Gao, G. Wang, and D. Li, "miprobert: identification of microrna promoters based on the pre-trained model bert," *Briefings in bioinformatics*, vol. 24, no. 3, p. bbad093, 2023.

H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, N. L. Carranza, A. H. Grzywaczewski, F. Oteri, C. Dallago, E. Trop, H. Sirelkhatim, G. Richard, *et al.*, "The nucleotide transformer: Building and evaluating robust foundation models for human genomics," *bioRxiv*, pp. 2023–01, 2023.

(0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0) (0)

Subject Transfer in Motor Motion and Motor Imagery EEG Recordings

Jason Cuthbert
University Of Colorado Colorado Springs
1420 Austin Bluffs Parkway
Colorado Springs, 80921
jcuthber@uccs.edu

Adham Atyabi
University Of Colorado Colorado Springs
1420 Austin Bluffs Parkway
Colorado Springs, 80921
aatyabi@uccs.edu

Abstract

This study implements a 3D CNN classification model for subject transfer to classify both Motor Motion (MM) and Motor Imagery (MI) tasks together in EEG recordings. An KMeans clustering algorithm is used to increase subject similarity in training data. Subject Transfer combines multiple subjects' EEG recordings to create a robust and generalized model for use in classification of subjects who are not part of the training data. MI exhibits higher classification potential than MM due to fewer artifacts. It has been the focus of the majority of studies due to historically inadequate classification methods. One of the benefits of subject transfer is that it enlarges the training dataset, which correlates with higher accuracy under deep learning. Additionally the ability to capture high resolution MM expressions enables finer control and accuracy during MI classification in BCI-controlled prostheses.

1 Introduction

1.1 Electroencephalograms (EEG)

EEG is a way of recording brain activity using an array of potential sensors on the scalp. These electrodes record the amalgamated activity of a large number of neurons. The high number of neurons represented by a single sensor, often numbered in the hundreds of millions, means that the extracranial sensors do not provide highly localized information. However, for any given task on the human scale, many neurons are utilized. Thus, through analyzing the oscillations in potential, especially with the aid of signal processing techniques, relevant patterns for a whole host of conditions and actions may be deduced. EEG is a proven tool for medical diagnosis, finding use in treating epilepsy, sleep disorders, and the extent of anesthesia, among others.

1.2 Brain-Computer Interface (BCI)

In recent decades, there has been interest in using EEG for creating a BCI, enabling users to control a computer system through mental effort alone or in conjunction with other methods of input. A BCI system can control a physical assembly, like a bionic limb, or serve as a supplemental interface for software control. This technology holds the promise of enhancing independence and interaction for individuals with paralysis or other conditions that hinder the use of standard interfaces.

1.3 Subject Transfer

The shifting and reforming nature of the brain proves to be a challenging environment for pattern recognition. Indeed, changes in a single person's EEG over the course of a few weeks may render previously found patterns useless. This is the result of new learning and adaptation within the brain, which is not fully understood. Subject Transfer is a solution to this problem, as well as providing benefits in reduced analysis time, increased sample size, and generalization[10]. Subject Transfer involves using EEG data from multiple subjects performing the same tasks to create a more robust and generalized model. This model can be applied beyond the original subject population, enhancing its usability. It relies upon the assumption that, among the general population, for a given task, similar brain functions are used, especially in the context of tasks that utilize older and more basic brain functionality, such as motor functions.

1.4 Motor Imagery (MI) and Motor Motion (MM)

Motor Imagery is the mental process of simulating movement without physical movement. Motor Motion is the mental and physiological process wherein a person moves voluntarily. The former, in a loose way, being a subset of the latter's mental processes [2]. The exact extent to which the two overlap is not currently known. Using EEG to record Motor Motion is made difficult by the increase in artifacts that accompany movement. Artifacts stem from both internal sources, by way of muscle movements and physiological processes, as well as external sources, by way of environmental interference [11]. Motor Imagery, in contrast, is easier to classify due to a lack of the artifacts associated with subject movement.

2 Related Works

This section enumerates studies that utilize the same dataset which this study uses. Note when viewing the accuracy table that the goal of each study, in terms of what each model is optimized for, may differ from one another, and indeed does differ from the goal of this study.

2.1 Bird et al.

Bird et al. proposes preproccessing EEG classification data by converting the recordings into visual images using dimensionality reduction. The visual representations are then processed using 2D and 3D convolutional neural networks (CNNs) to extract additional features. Experimental results demonstrate high classification accuracy; the highest seen with this dataset, indicating the effectiveness of the proposed approach in extracting useful features. Its worth noting however that the main focus of this study was their own data collection; the EEGMMIDB[9] was only used as a validation dataset; utilizing two classes: eyes open, and eyes closed. Hence the smaller portion of subjects used. For this reason this study while holding the highest accuracy has been excluded from related performance as accuracy on a subsection of the dataset is not a valid comparison to accuracy on the dataset as a whole.

2.2 Dose et al.

Dose et al. uses a Convolutional Neural Network (CNN) layers for learning features and a Fully Connected (FC) layer for classification, applied to raw EEG data. The results show that the DL model achieves high accuracy, with the expected dip in performance in the tests with higher class numbers.

2.3 Mammone et al.

Mammone et al. uses Auto Encoder-Filter Bank Common Spatial Paterns (AE-FBCSP) for classification. AE-FBCSP combines the FBCSP approach with a global and subject-specific transfer learning approach.

2.4 Karacsony et al.

Karacsony et al. presents a real-time EEG-based MI-BCI system with a virtual reality (VR) game as motivational feedback for stroke rehabilitation. The system utilizes deep learning a CNN architecture with a unique trial onset detection technique to achieve improved classification performance. The classifier was tested online and offline. The offline results for 6s intervals are listed in the table below, achieving the highest beyond-binary accuracy among the related studies. The online results, although not classified as such, would constitute subject transfer. The accuracy measurements for the online portion are not reported within the study.

2.5 Wang et al.

Wang et al. diverges from the standard goal of high accuracy demonstrating an embedded MI-BCI with a focus on classification under the hardware limitations of low-power micro-controller units (MCUs). Using the ARM Cortex-M family as a flag-bearer for such devices; down-sampling, channel selection, and narrowing of the classification window are used to further reduce the memory requirements of the model with minimal accuracy degradation.

3 Problem Statement

This study proposes a model for the task classification of Motor Motion (MM) and Motor Imagery (MI) datasets for subject transfer. MI, due to its generally lower number of artifacts, traditionally exhibits higher classification potential compared to MM. Achieving subject transfer between MM and MI would allow for larger datasets, which, in turn, tends to correlate with higher accuracy under the deep learning paradigm. Furthermore, greater resolution in the physical expression of MM would enable finer control and accuracy of intention during MI classification in the context of brain-computer interface (BCI) controlled prostheses. Greater levels of bio-fidelity being the abiding goal of BCI prostheses due to the physical[8] and physiological benefits[13]. The physical expression of MM can be captured using existing technologies such as mmWave Sensors[5] or more traditional video-based methods[7].

ref	#Subjects	CV=	Classes	Performance %
[1]	105	10	2,3,4	80.38, 69.82, 58.58
[6]	105	5, 6	2,3,4,5	74.75, 72.32, 69.12, 68.04
[4]	105	5	2,3,4	85.94, 88.50, 76.37
[12]	105	5	2,3,4	82.43, 75.07, 65.07

Table 1: Related Performance

Year	Preprocessing	Optimization	Classification Model
2021	One-Rule, Kullback-Leibler Divergence, and Symmetrical Uncertainty.	DEvoMLP	2D-CNN: Visual Space Learning
2021	One-Rule, Kullback-Leibler Divergence, and Symmetrical Uncertainty.	DEvoMLP	3D-CNN: Visual Space Learning
2018	Adam		FC
2023	LASSO	AE + FBCSP filtering	FNN
2019	Butterwork BP filter 0.5-75Hz, 50Hz Notch filter, FC Running Standardization		CNN
2020	Temporal down-sampling		Scaled EEG-NET

Table 2: Related Processing

4 Dataset

This study uses the EG Motor Movement/Imagery Dataset[9] featuring 109 participants and 10 classes; 4 MM, 4 MI, and 2 Baseline classes. Out of the 109 subjects, 105 will be used in this study due to sample rate errors in four subjects. The subjects with errors to be left out are: 88, 92, 100, and 104.

Table 3: EEG Motor Movement/Imagery Dataset

Year	Type	System	Classes	(Hz)	Ch	Subjects
2009	EEG		8	160	64	109

Table 4: EEG Motor Movement/Imagery Dataset Tasks

Recordings	Tasks					
	MM	MI	Left Fist	Right Fist	Both Fists	Both Feet
1,2	Eyes open/closed					
3,7,11	✓		✓	✓		
4,8,12		✓	✓	✓		
5,9,13	✓				✓	✓
6,10,14		✓			✓	✓

5 Approach

5.1 Preprocessing

Several preprocessing have been employed: FIR Band Pass Filter, Common Average Reference (CAR), Independent Component Analysis (ICA), and Baseline Correction. Gramfort et al. [3] developed Python MNE a python library for bio-metric data processing library that was used in this study. Additionally the data consisting of four second epochs has been sliced into overlapping half-second sub epochs, fifteen per epoch.

Different Band-Pass filters were used for the Motor-Movment and Motor-Imagery portions of the dataset. The filter for the Motor-Movement was 1-79Hz while the filter for the Motor-Imagery was 1-30Hz. The reason for this is that the former portion has considerably more movement artifacts. This meant that by reducing these artifacts preprocessing would remove a larger portion of the useful data within the Motor-Movement set.

5.2 Clustering

Before classification, data for each subject is divided into two sets at a ninety-to-ten ratio. These are subsequently referred to as the testing set and the clustering set respectively. The smaller clustering set from the target subject's data, along with the complete data from all other subjects, is clustered into two groups using KMeans clustering. The cluster containing the majority of the target subject's clustering set is then selected. From this chosen cluster the target subject's clustering set is removed, leaving behind only the epochs from other subjects that the clustering has identified as most similar to the target subject's epochs. These other subjects similar epochs, whether in the company of the whole of the originating subject epochs, or being sole representatives are used as training data for the classification model. This model is then tested on the target subject's testing set.

5.3 Rasterization

A process of rasterization was undertaken adding a spatial component to the time series data. This process utilized a standard montage of the International 10-10 System. The three dimensional coordinates denoting electrode placement on the scalp were flattened using an orthographic projection.

Table 5: Binary Classification Accuracy

Subject	2 class MM	2 class MI
1	0.681	0.630
2	0.703	0.961
3	0.760	0.988
4	0.813	0.682
5	0.662	0.780
6	0.618	0.729
7	0.751	0.747
8	0.713	0.759
9	0.676	0.752
10	0.742	0.715
11	0.608	0.835
12	0.688	0.911
13	0.637	0.638
14	0.653	0.753
15	0.737	0.667
16	0.582	0.744
17	0.701	0.889
18	0.667	0.656
19	0.656	0.620
20	0.643	0.573
21	0.794	0.598
22	0.638	0.682
23	0.565	0.960
24	0.620	0.712
25	0.638	0.643
26	0.657	0.685
27	0.680	0.675
28	0.615	0.707
29	0.809	0.778
30	0.603	0.703
31	0.615	0.635
32	0.701	0.649
33	0.732	0.777
34	0.743	0.689
35	0.851	0.759
36	0.656	0.969
37	0.620	0.609
38	0.561	0.660
39	0.544	0.788
40	0.655	0.631
41	0.611	0.677
42	0.809	0.809
43	0.656	0.802
44	0.683	0.593
45	0.805	0.735
46	0.614	0.747
47	0.605	0.671
48	0.831	0.596
49	0.661	0.692
50	0.588	0.666
51	0.604	0.667
52	0.701	0.671
53	0.674	0.646

Table 6: Binary Classification Accuracy

Subject	2 class MM	2 class MI
54	0.711	0.724
55	0.677	0.634
56	0.704	0.716
57	0.663	0.777
58	0.805	0.991
59	0.705	0.671
60	0.722	0.620
61	0.678	0.644
62	0.694	0.730
63	0.703	0.620
64	0.590	0.724
65	0.665	0.604
66	0.584	0.732
67	0.635	0.783
68	0.777	0.803
69	0.689	0.766
70	0.683	0.665
71	0.759	0.750
72	0.807	0.623
73	0.645	0.771
74	0.669	0.890
75	0.605	0.679
76	0.527	0.800
77	0.588	0.757
78	0.573	0.814
79	0.788	0.776
80	0.716	0.759
81	0.541	0.718
82	0.677	0.794
83	0.669	0.721
84	0.680	0.874
85	0.809	0.901
86	0.802	0.708
87	0.585	0.839
89	0.650	0.754
90	0.735	0.772
91	0.718	0.889
93	0.724	0.782
94	0.671	0.781
95	0.580	0.783
96	0.685	0.897
97	0.682	0.806
98	0.616	0.695
99	0.656	0.723
101	0.595	0.810
102	0.667	0.623
103	0.703	0.803
105	0.671	0.800
106	0.670	0.647
107	0.747	0.623
108	0.630	0.681
109	0.753	0.697
Average:	0.6677	0.7357

The flattened coordinates were then rasterized into a 17x17 frame for each discrete sensor read within a sub-epoch.

5.4 Model

In order to capture the spatial information provided by the rasterization a 3D CNN was used to traverse each (80, 17, 17) sequence of frames. Several regularization steps were taken to reduce over-fitting. L2 regularization, dropouts, and batch normalization were used in the model. After experimentation GRU's were chosen for the slight performance improvement that they provided. While a transformer model was tested the amount of available data was insufficient to garner the known advantages in the architecture. After experimentation a parallel model was chosen due a slight increase in performance. Lastly spatial drop-outs have been applied randomly at the frame level within each sub-epoch at a rate of 35% of sub-epochs and a coverage area of 2x2 which were replaced with zeros. The complete model can be found in Figure 1.

After training on all other subjects the model was tuned with the subjects 30% of the subjects' data significantly increasing performance.

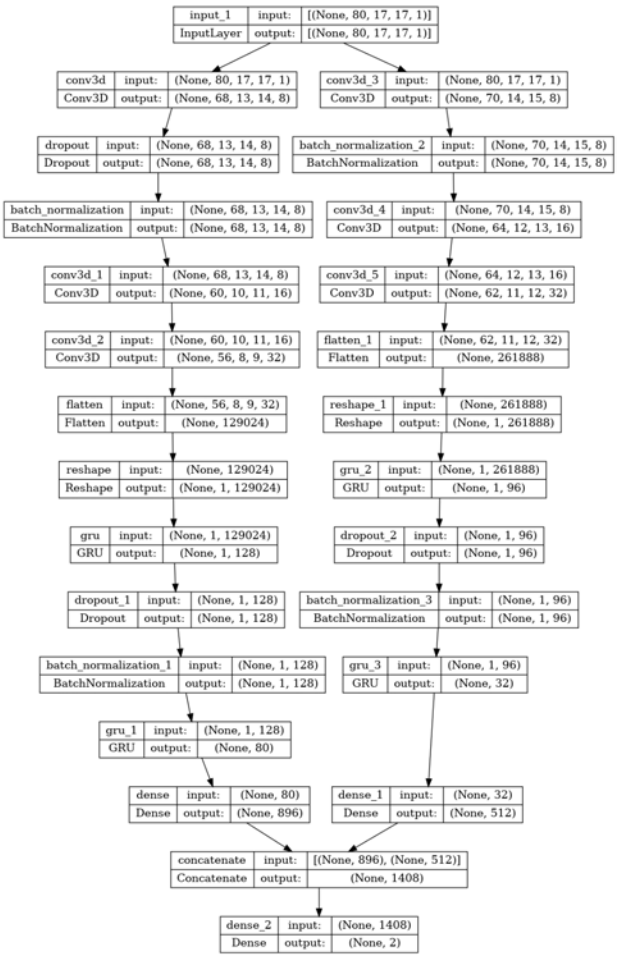


Figure 1: CNN Architecture

7 Conclusion

A 3D CNN model with GRU has been implemented for the classification of EEG Motor-Movement and Motor-Motion. While the subject transfer Motor-Movement results are below the state of the art; the Motor-Motion results surpass it. Additionally the single trial EEG results for both Motor-Movement and Motor-Motion are at par with the state of the art at times reaching 100% classification accuracy. The addition of spatial dropout as a preprocessing step was significant to the achievement of the stated results. The implementation of a spatial drop-out system; otherwise known additionally as cutout, is novel in EEG classification. Spatial drop-out increased performance significantly in ablation testing, and warrants further study.

9 Acknowledgements

The work reported in this paper is supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Hauke Dose et al. “An end-to-end deep learning approach to MI-EEG signal classification for BCIs”. In: *Expert Systems with Applications* 114 (2018), pp. 532–542.

[2] Emmanuel Gerardin et al. “Partially overlapping neural networks for real and imagined hand movements”. In: *Cerebral cortex* 10.11 (2000), pp. 1093–1104.

[3] Alexandre Gramfort et al. “MEG and EEG Data Analysis with MNE-Python”. In: *Frontiers in Neuroscience* 7.267 (2013), pp. 1–13. DOI: 10.3389/fnins.2013.00267.

[4] Tamás Karácsony et al. “Brain computer interface for neuro-rehabilitation with deep learning classification and virtual reality feedback”. In: *Proceedings of the 10th Augmented Human International Conference 2019*. 2019, pp. 1–8.

[5] Yilin Liu et al. “Leveraging the Properties of mmWave Signals for 3D Finger Motion Tracking for Interactive IoT Applications”. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6.3 (2022), pp. 1–28.

[6] Nadia Mammone et al. “AutoEncoder Filter Bank Common Spatial Patterns to decode Motor Imagery from EEG”. In: *IEEE Journal of Biomedical and Health Informatics* (2023).

[7] Franziska Mueller et al. “Ganerated hands for real-time 3d hand tracking from monocular rgb”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 49–59.

[8] Michael A Saliba, Maria Cutajar, and Gabriel Agius Pascalidis. “Prototype Development of a Prosthetic Derivative of the Minimal Anthropomorphic Artificial Hand”. In: *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2022, pp. 14–21.

[9] Gerwin Schalk et al. “BCI2000: a general-purpose brain-computer interface (BCI) system”. In: *IEEE Transactions on biomedical engineering* 51.6 (2004), pp. 1034–1043.

[10] Wenting Tu and Shiliang Sun. “A subject transfer framework for EEG classification”. In: *Neurocomputing* 82 (2012), pp. 109–116.

[11] Jose Antonio Urigüen and Begoña Garcia-Zapirain. “EEG artifact removal—state-of-the-art and guidelines”. In: *Journal of neural engineering* 12.3 (2015), p. 031001.

[12] Xiaying Wang et al. “An accurate eegnet-based motor-imagery brain–computer interface for low-power edge computing”. In: *2020 IEEE international symposium on medical measurements and applications (MeMeA)*. IEEE. 2020, pp. 1–6.

[13] Yuxuan Wu. “Applications of EEG-Based Brain-Computer Interface Devices in Rehabilitation”. In: *Highlights in Science, Engineering and Technology* 39 (2023), pp. 809–815.

Examining the Efficacy of Deep Transfer Learning in Forecasting Seizures

Brett Ford
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918

Adham Atyabi
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918

Abstract

Epilepsy is a condition characterized by chronic seizures. EEG(electroencephalogram) can be used to predict and analyze seizures. However, the work of reading EEG requires a large amount of training and still has a large amount of variability and error when performed by humans. Several studies have attempted to train artificial intelligence to forecast upcoming seizures. In this endeavor, steady progress has been made, but there is still potential for improvement. This study proposes training a LSTM(Long Short-Term Memory) model for the purpose of forecasting seizures. Additionally, transfer learning is implemented to fine-tune the model between subjects. The model was trained on data from the CHB(Children's Hospital Boston) Database. Once a model was made, it was fine-tuned and tested on a subject to evaluate its performance. The model achieved an accuracy of 95% when tested after fine-tuning.

Introduction

Epilepsy affects 3 million Americans and 50 million people worldwide. Nearly one-third of individuals with epilepsy continue to have seizures despite medications intended to prevent them(Sheob 2009). People who experience epileptic seizures often experience fear, anxiety, and depression in their everyday lives, largely because of the fear of an unpredictable seizure. Epileptic seizures may come in several stages: most notably the ictal phase, which includes the seizure itself, the preictal phase, a period before the seizure begins, and the postictal phase, which occurs after the seizure. Some people with epilepsy may have some symptoms associated with the preictal phase, but others have little or no warning(Stirling et al. 2020).

EEG(electroencephalography) is a method for measuring neurological signals. An EEG reads the fluctuations of voltage at various locations around a subject's head. These fluctuations give insights into the state and activity of corresponding regions of their brain. The many ways the voltage sensors can be placed are called montages. EEG data can be collected non-invasively so it has real and meaningful potential to detect, classify, and forecast seizures without harming the individual. EEG has traditionally been interpreted by specialists which requires extensive training(Patel et al 2020).

Professionals have noted a recognizable difference in the patterns of epileptic subjects in a preictal phase. These patterns are often measurable physiologically(Patel et al 2020). This gave researchers hope that oncoming seizures could be predicted, however, the expression of the preictal period on an EEG between subjects can be very different(Sheob 2009).

Before features are extracted from the data, it is common to do some preprocessing to remove artifacts. Usually, any signal caused by the power to the EEG recording device is removed. Often the spectrum of frequencies is shortened, either by removing very high-frequency data or very low-frequency data(Truong, 2018).

Because EEG is a well-established discipline that has been used by researchers for many years, many types of feature extraction have been developed. measuring spectra power is one of the most common, as well as strategies to quantify frequency like fast Fourier transform and discrete wavelet transform, are very common. Additionally, many approaches look for spikes or other recognizable patterns. These approaches, and others, have traditionally been used in interpreting EEG signals and have been proven to be able to identify Seizures(Tsiouris et al. 2018). This study attempted to perform minimal preprocessing because the effectiveness of these strategies in the field of machine learning is not well established(Delorme, 2023).

Some ideas have been put forward pertaining to seizure forecasting. Mona Nasseri et al. have proposed an ambulatory seizure prediction system in two parts. One part is implanted invasively to receive more accurate EEG signals, and one component is worn on the wrist. When a seizure is forecast according to a model, the wrist-worn piece will give the wearer advance warning. This could do a tremendous amount to free wearers from the fear of unexpected seizures(Nassari et al., 2021).

Related Work

Several studies have attempted to forecast the onset of seizures prior to the ictal phase. Primarily four classification strategies can be found in the literature to solve this problem: SVM, CNN, LDA, and LSTM. SVM(support vector machines) are a strategy that seeks to find the maximum distance between data on opposite sides of a boundary. LDA (linear discriminant analysis) is a statistical tool similar to ANOVA. CNN(convolutional neural network) is a type of

machine learning primarily used to analyze images. LSTM(long short-term memory) is a form of deep learning which is often used in language processing. Table 1 shows a selection of previous work done forecasting seizures on the CHB-MIT scalp EEE Database.

In 2018, Tsiouris et al. achieved a very strong result using an LSTM. This team used a wide variety of feature extraction techniques to provide the model they used as much data as possible; they made use of cross correlation, time domain, frequency domain, and graph theory to extract information from the data and provide it to the model. They corrected for the imbalance in the data by oversampling data from the period before a seizure. The model made by this team was trained on each subject separately. the resulting model achieved a sensitivity of 99.84% on a window of 2 hours before a seizure, and 99.7% on a period of one hour before(Tsiouris et al. 2018)

Truong et al. created a model that was more generalizable than many others. They chose a 2-dimensional CNN to extract features from the signals instead of providing the model with many features they extracted themselves, although, a short-time Fourier transform was utilized. Once their model was completed, it could be applied to new patients without retraining. To overcome the imbalance in their dataset, they developed a method of oversampling data from the period before a seizure by using a sliding frame. This generates data that is unique but shares some data with overlapping windows. To reduce the occurrence of false positives, the team took data from several consecutive samples and only reported a positive prediction if a predetermined number of samples predicted one. Their approach achieved a 75% sensitivity rate on data from the CHB-MIT scalp EEG Database. They used two additional databases(Truong et al 2018).

Problem Statement

Despite many successes with different types of models, there are still shortcomings that will need to be overcome. Many of the high-performing models are trained on a specific case. This has advantages as it helps improve the metrics of the model, but it also means that the model cannot be generalized to new subjects easily and may require extensive training to apply to a new subject. The most useful model should have very high accuracy, a low frequency of false positives, be very generalizable, and be simple.

In order to make a seizure forecasting system that does not need to be trained on each subject individually, this study proposes a transfer learning model. A robust model has been generated using data from many subjects. Once the model is trained it has been fine-tuned with a minimal amount of data from a new subject. Transfer learning helps to relax some of the assumptions normally required of training data. In transfer learning, data does not need to be independent, identically distributed to the test data, and very abundant(Tan et al. 2018). By the use of transfer learning and LSTM, a model can be made which can be very quickly fit to new subjects but which will also be effective at forecasting encroaching seizures.

Dataset

The CHB-MIT Scalp EEG Database(Children's Hospital Boston-Massachusetts Institute of Technology Electroencephalogram Database). is a collection of EEG recordings from 22 subjects with intractable seizures. The dataset has cases from 5 male and 17 female subjects who range in age from 3 to 22 years old. One female subject was repeated several years later and is recorded as a second dataset. Data were collected for several days after their anti-seizure medication was removed. The dataset contains 916 hours of continuous recordings at 256Hz. the recordings were made using signals from 23, 24, or 26 sensors. The electrodes were placed using the international 10-20 system(Shoeb, and Guttag, 2010).

173 seizures are indicated in the dataset. each seizure was annotated by experts to show when it began and ended. The recordings are divided into one-hour data sets. Each subject has between 9 and 42 data sets. 129 data sets contain at least one seizure. A 24th subject was added to this database after the initial 23, however, the records do not contain record start times so that subject is not included in the following analysis.

Methodology

Broadly, the plan for this study was.

1. The EEG data will be read using mne in python
2. The data will have a high pass filter at 0.2Hz applied to remove noise
3. The data from most of the subjects will be re-balanced using SMOTE and under-sampling to make the training groups close to 50% of each label.
4. the subject randomly selected for fine-tuning will have their data normalized, but not re-balanced.
5. The data will be split up into parts for training, testing, and validation.
6. The training data will be used to train a model
7. That model will then be fine-tuned on a small portion of data taken from one subject not previously used to fit the model
8. The training data will be validated to assess the model and the applicability of the approach

The data EEG data is processed into frames that have 18 channels and 600 time steps per channel. 600 time steps represent about 2.34 seconds of EEG recording. These frames are labeled according to whether a seizure occurs in a specified period after the recording ends; the period for this study is one hour. This period should roughly align with the preictal period. Frames that include seizures are excluded. All other frames are shuffled along with their label. Frames are normalized using Z-score normalization. It is worth mentioning that each frame is normalized independently, not normalized by subject, or by channel data outside that period.

The data is inherently very imbalanced with far more data coming from periods when nothing is indicated than from

Authors	Number of subjects	Classifier	Sensitivity	FP/h	Preictal length (min)
Zhang and Parhi	17	SVM	98.68	0.046	60
Cho et al.	21	SVM	82.44	-	5
Alotaiby et al.	24	LDA	89	0.39	120
Truong et al.	13	CNN	81.2	0.16	30
Tsiouris et al.	24	LSTM	99.84	0.02	120

Table 1: A table of previous work

the period of interest before a seizure. Data is considered imbalanced when one label occurs far more frequently than another. The class with the greater proportion of labels is called the majority class, while the class with fewer labels is called the minority. Imbalanced data can prevent a model from fitting to data correctly (Chen, Chang, and Guo, 2021). In order to address this, two strategies have been implemented on the data in this study: near miss under-sampling, and SMOTE. Near miss under-sampling is a strategy that seeks to remove data from the majority class to reduce the size of that class and decrease the difference in the sizes. It attempts to remove data that resembles the other classes such that the remaining data is as distinct as possible and the model can train more easily. SMOTE(Synthetic Minority Oversampling Technique) is a tool that attempts to generate new data that resembles the minority class. Chen et al. have noted that SMOTE tends to perform better on EEG data than under-sampling. In this study most of the re-balancing was done using near-miss under-sampling, SMOTE was not permitted to increase the size of the minority class by more than 50% for any given subject. Each subject was re-balanced separately so that SMOTE would generate new seizure data for each subject.

The model selected for the analysis has convolutional layers, a LSTM, and dense layers. It is expected that the convolutional layers find features in the data coming from each of the EEG sensors. This data is then fed into a max pooling layer in part to reduce the amount of time information being fed into the LSTM(long-short term memory). Up until this point in the model, the data from the 18 channels(the data from the 18 sensors in the bipolar montage selected for this study) has been left individual. The work of combining that data is left up to the LSTM and the subsequent dense network. This scheme was selected because the channel data is unordered, so a CNN performed poorly in finding patterns. The dense network at the end of the model is the only portion of the model that will be altered during fine-tuning.

Results

To create a model that would be as accurate as possible, all of the subject data available was used except for subject chb12 who was randomly selected to be excluded for testing. The training data was re-balanced to include 47.3% data that is inside the period of interest. The period of interest for this trial was 60 minutes. From the balanced dataset, 0.5% was set aside for testing, this accounts for 2419 data points. These data points are separate from subject chb12, whose data is also set aside. The remaining 484,000 or so data points were used for training the model. The model described above achieved an accuracy of 88.45% with a loss of 0.245. figure 2 shows a confusion matrix of its predictions compared to the actual labels. It had a sensitivity of 88.15% and a specificity of 89.08%.

The model above however performed meaningfully worse when applied to real-world data. A set of testing data taken from subject chb12(deliberately excluded from the training set) contained 3537 datapoints indicated as being outside the period of interest, and only 154 inside that period. In this case, the period of interest was one hour. When the model above was applied to this testing data without any fine-tuning, it only achieved 43.15% accuracy and had a loss of 1.975. Figure 3 shows the confusion matrix for this trial.

The model above was then fine-tuned on the data from subject chb12. About 9.5% of the data from this subject was used to fine-tune the model. 85% of the remaining data from this subject was used to test the model. Some data was labeled as validation data, but it was not used for either. The convolutional layers and the LSTM were locked so that they would not train. After locking the model, it had 8,444 trainable parameters. 66,360 were non-trainable. The model was fine-tuned for 12 epochs, after which it was evaluated on the same testing data it had been tested on previously(a distinct dataset from the one being used for training). The model after this fine-tuning achieved 95.88% accuracy and a loss of 0.191. The confusion matrix for this test can be seen in Figure 4. This model has a sensitivity of 0.5% and a Specificity of 96.1%.

Conclusion

The CNN-LSTM model seems to be effective at forecasting approaching seizures at a period of one hour prior. Additionally, transfer learning effectively steered a previously trained model to predict seizures with a minimal amount of fine-tuning. Additional work should be done to verify these results. They should be cross-validated to prove they are effective in many subjects. Additionally, The model should also be tested on non-shuffled data to search for patterns in its false labels.

Acknowledgment

The work reported in this paper is supported by the National Science Foundation under Grant No. 2050919. Any

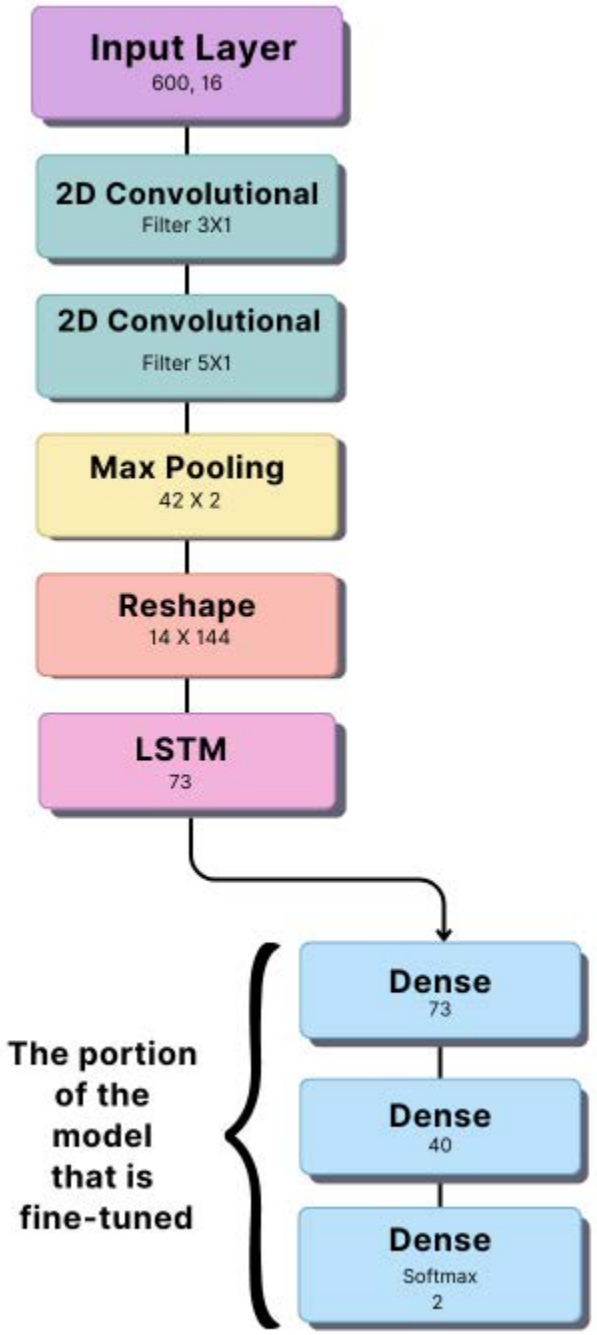


Figure 1: The Layout of The Highest Performing Neural Network.
This model is the highest-performing model found in this study, although better ones may exist elsewhere. It uses convolutional neural networks and a LSTM to find features, max pooling to reduce the number of timesteps, and a dense neural network to classify the data.

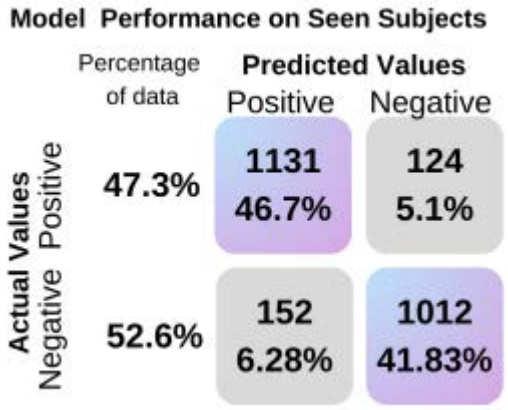


Figure 2: A confusion Matrix of the selected model
this matrix shows the accuracy of the select model's predictions compared to the actual labels. It was tested on a testing dataset of 2419 data points that was balanced at 47.3% of data inside the period of interest(positive) and 52.7% outside the period of interest(negative), both the model and the testing data in this figure exclude subject chb12.

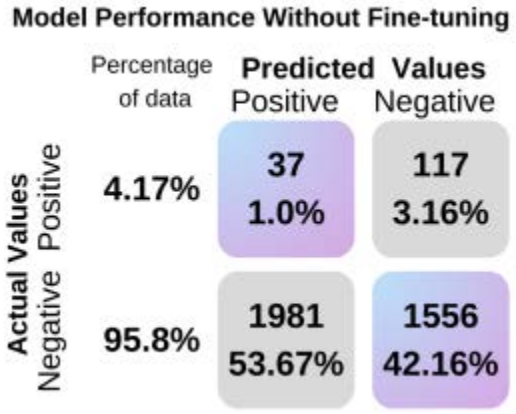


Figure 3: A confusion Matrix of the selected model on data from subject chb12 This matrix shows the considerably higher failure rate on data that resembles real-world data. The imbalanced labels in the testing data may have caused the model to perform considerably worse.

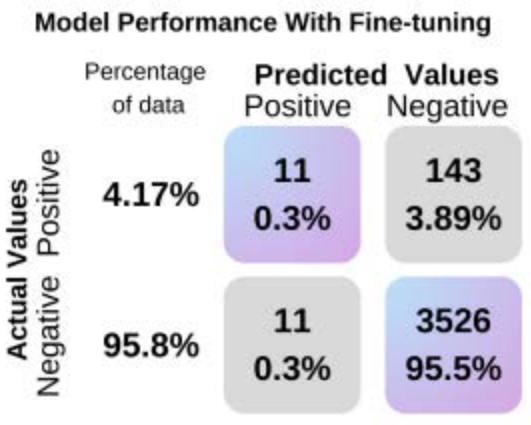


Figure 4: Performance of a fine-tuned model
This confusion matrix shows the performance of a model after being fine-tuned. It is tested on the same testing data as Figure 3.

opinions, findings, and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet *Components of a new research resource for complex physiologic signals*. Circulation [Online]. 101 (23), pp. e215–e220. https://physionet.org/content/chbmit/1.0.0/

Stirling, Rachel E., Mark J. Cook, David B. Grayden, and Philippa J. Karoly.March 2020 *Seizure Forecasting and Cyclic Control of Seizures*. Epilepsia 62, no. S1 (2021): S2–14. https://doi.org/10.1111/epi.16541.

Tsiouris, Kostas M., Vasileios C. Pezoulas, Michalis Zervakis, Spiros Konitsiotis, Dimitrios D. Koutsouris, and Dimitrios I. Fotiadis. August 1, 2018. *A Long Short-Term Memory Deep Learning Network for the Prediction of Epileptic Seizures Using EEG Signals*. Computers in Biology and Medicine 99 24 – 37. https://doi.org/10.1016/j.compbiomed.2018.05.019.

Mona Nasseri, Tal Pal Attia1, Boney Joseph, Nicholas M. Gregg1, Ewan S. Nurse, Pedro F. Viana, Gregory Worrell1, Matthias Dümpelmann, Mark P. Richardson, Dean R. Freestone and Benjamin H. Brinkmann 2021 *Ambulatory Seizure Forecasting with a Wrist-Worn Device Using Long-Short Term Memory Deep Learning.* Scientific Reports 11:21935

Ozcan, Ahmet Remzi, and Sarp Erturk. November 2019 *Seizure Prediction in Scalp EEG Using 3D Convolutional Neural Networks With an Image-Based Approach.* IEEE Transactions on Neural Systems and Rehabilitation Engineering 27, no. 11: 2284–93. https://doi.org/10.1109/TNSRE.2019.2943707.

Tan, Chuanqi, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, Chunfang Liu. 2018 *A Survey on Deep Transfer Learning.* Artificial Neural Networks and Machine Learning – ICANN 2018. 270–79. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-01424-7.

Obeid, Iyad, and Joseph Picone. 2016 *The Temple University Hospital EEG Data Corpus.* Frontiers in Neuroscience 10 https://www.frontiersin.org/articles/10.3389/fnins.2016.00196.

Truong, Nhan Duy, Anh Duy Nguyen, Levin Kuhlmann, Mohammad Reza Bonyadi, Jiawei Yang, Samuel Ippolito, and Omid Kavehei. 2018 *Convolutional Neural Networks for Seizure Prediction Using Intracranial and Scalp Electroencephalogram.* Neural Networks 105 104–11. https://doi.org/10.1016/j.neunet.2018.04.018.

Stevenson, N. J., K. Tapani, L. Lauronen, and S. Vanhatalo. March 5, 2019*A Dataset of Neonatal EEG Recordings with Seizure Annotations.* Scientific Data 6, no. 1 : 190039. https://doi.org/10.1038/sdata.2019.39.

Shoeibi, Afshin, Delaram Sadeghi, Parisa Moridian, Navid Ghassemi, Jónathan Heras, Roohallah Alizadehsani, Ali Khadem, et al. 2021*Automatic Diagnosis of Schizophrenia in EEG Signals Using CNN-LSTM Models.* Frontiers in Neuroinformatics 15. https://www.frontiersin.org/articles/10.3389/fninf.2021.777977.

Delorme, Arnaud. February 9, 2023*EEG Is Better Left Alone.* Scientific Reports 13, no. 1: 2372. https://doi.org/10.1038/s41598-023-27528-0.

Chen, Yu, Rui Chang, and Jifeng Guo.2021*Effects of Data Augmentation Method Borderline-SMOTE on Emotion Recognition of EEG Signals Based on Convolutional Neural Network.* IEEE Access 9 : 47491–502. https://doi.org/10.1109/ACCESS.2021.3068316.

Patel, Vibha, Sanjay Buch, and Amit Ganatra. 2020 *A Review on EEG Based Epileptic Seizure Prediction Using Machine Learning Techniques.* In Intelligent Computing, Information and Control Systems, edited by A. Pasumpon Pandian, Klimis Ntalianis, and Ram Palanisamy, 384–91. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-30465-$2_4$3.

Shoeb, Ali, and Guttag, John. 2010 *Application of Machine Learning To Epileptic Seizure Detection* Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel

Finger Flex Classification for Brain Computer Interfaces

Cynthia Chen
cynthia.c@rutgers.edu
Rutgers University - New Brunswick
57 US Highway 1
New Brunswick, NJ 08901

Adham Atyabi
aatyabi@uccs.edu
University of Colorado Colorado Springs
1420 Austin Bluffs Parkway
Colorado Springs, Colorado 80918

Abstract

Brain Computer Interfaces (BCIs) allow users to control an external device with only their mind. However, the electroencephalography (EEG) signals that are generally used for the operation of these BCIs are noisy and high dimensional, making it difficult to classify fine motor movements such as individual finger flexes. Additionally, brain states across different patients vary, and brain states can even vary between recording sessions, making it difficult to develop a classification model that generalizes well. This means that BCIs generally require a costly and time consuming tuning period for each new user. The calibration of BCIs to detect the movement of individual fingers is also difficult because signals from different fingers are very similar to one another. These challenges have made it difficult to use BCIs for practical purposes. However, with the advent of deep learning, features can now be automatically extracted from data and large datasets can be leveraged to train powerful models. This study applies deep learning techniques and models to finger flexion data in order to create better classifiers. We also explore the possibility of subject transfer in this domain.

1 Introduction

Brain Computer Interfaces

Brain computer interfaces (BCIs) are devices that allow for direct communication between a human brain and an external device (Wolpaw et al., 2002). These devices read brain signals acquired through various methods such as Electroencephalography (EEG) in order to decipher the intent of the user and communicate this intent to the device the user operates. This device can be anything from a wheelchair, computer, or prosthetic. However, the types of signals commonly used for BCI are difficult to classify because they have a low signal to noise ratio and are often non-stationary (Khademi et al., 2023). The signals are also easily contaminated by artifacts such as muscle twitches, eye blinking, static electricity or other irrelevant background brain processes. EEG signals are measured with an electrode cap that sits on the scalp, meaning that there is a layer of skin and bone between the electrodes and one's actual brain. Furthermore, there is a limit to how many electrodes can be placed on an EEG cap, limiting the spatial resolution of the brain signals. Because of these difficulties interpreting electrical brain signals, BCIs have not been adapted for everyday use. Machine learning models are not able to obtain very accurate classification results on subtle brain signals, such as those representing fine motor movements or emotional states. However, with the increasing advances in deep learning, there is an increased ability to bypass manually cleaning and extracting the important aspects of the data by relying on the model itself to do that. These technological improvements may allow us to achieve better classification accuracy results on previously unsolved BCI applications and improve the usability and functionality of BCIs.

Motor Imagery

One useful application of Brain computer interfaces is Motor Imagery (MI) classification. Motor imagery is the imagination of an intended motor movement, such as moving one's hand or closing one's fist. This imagination can show up in the brain signal as an event related desynchronization, which is a desynchronization from the baseline brain rhythm, or event related potential, which is strong polarization of the brain signal. MI is a popular BCI study paradigm because of the many potential applications of being able to interpret MIs. There have been significant advancements in the field, with classification accuracies in the 90s (Hekmatmanesh et al., 2020) for individual subjects. However, most studies focus on differentiating the signals that come from more general movement paradigms like left and right hand movements, and do not attempt to classify fine motor movements like individual finger flexions. This may be because finger flexions are controlled by a small and crowded region on the brain, and these brain signals are less distinguishable than brain recordings of more general motor imagery paradigms such as the right and left arm paradigm. In addition to the inherent difficulties distinguishing between fine motor movements, there are not many publicly available datasets demonstrating the finger flexion paradigm. In 2008, the BCI Competition IV (**?**) released a dataset of ECoG recordings of 3 subjects. This was one of the first finger movement datasets released publicly, however ECoG data is obtained through invasive brain surgery, which is not practical for the average user, and the recordings released were of physical motor movement, not motor imagery. In 2018, Kaya et al. (Kaya et al., 2018) released a large dataset of

motor imagery paradigm EEG recordings, which included 8 subjects who performed finger flex paradigm experiments. Because of the nature of EEG recordings, the spatial dimension of the signal is less precise and therefore EEG data is harder to work with than ECoG data.

Deep Learning Techniques

Recently, there has been a huge surge in the use of deep learning techniques in the fields of computer vision and natural language processing. Deep learning refers to the use of multilayer, or "deep" neural networks that are both more complex and more powerful than previous machine learning models. These models employ large amounts of data and many parameters in order to create better predictions. Popular deep learning architectures are Convolutional Neural Nets (CNNs), Recurrent Neural Nets (RNNs), and Transformers. Because deep learning techniques are able to automatically learn high level features from raw data, they are able to bypass the complex and intricate task of feature extraction. Because of this property, deep learning has begun to be applied to various signal processing tasks, including the classification of EEG signals for BCIs. Based on a literature review of deep learning techniques that have been applied to BCI development, CNNs are by far the most popular deep learning model, and while CNNs are usually used by themselves, they are also combined with other deep learning models in hybrid architectures (Altaheri et al., 2021).

Subject Transfer

The classification accuracy for finger movements on individual subjects is not high, and no substantial attempts on subject transfer have been made in this domain. There is much room for improvement in classifying finger flexions, as the classification of other motor imagery paradigms have reached much higher accuracies. Being able to have a model trained on one subject perform well on another subject, or "Transfer learning", is important for practical implementation of BCIs. However, due to the differences between peoples' brains, this has been pretty difficult to successfully implement. Because deep learning techniques are designed to automatically extract features and look for similarities between the data, we hope that deep learning can perform better on subject transfer because the model will be able to determine the important commonalities between subjects.

2 Problem Statement

In order for BCIs to have practical applications, the problem of decoding fine motor movements must be solved. By analyzing EEG recordings of finger flex data, we hope to apply deep learning techniques to improve the classification accuracy of these motor imageries and develop more useful BCI systems.

3 Related Works

In order to develop a model that can accurately classify finger flexions, we look at the preprocessing, feature extraction, and classification techniques used in past models and compare their classification accuracy results.

Data

The EEG data that is used for this study is the data from the Kaya et al 2018 dataset, which consists of 19 recording sessions that were administered on 8 subjects. The EEG signals are recorded in 200 and 1000 Hz, but downsampled to 200 Hz for consistency, and the electrode cap used was an EEG-1200 JE-921A consisting of 19 electrodes in the standard 10/20 configuration. Each recording session consists of about 900 trials where a cue is shown for 1s and then there is a rest period of 1.5-2.5s after the cue disappears. There are five possible cues, one for each finger. The subjects were instructed to imagine the finger movement of the finger shown one time per cue.

State of the Art

For the Kaya et al dataset (Kaya et al., 2018) there have only been four relevant studies utilizing it apart from the study that originally released the dataset. In the past five years, there has not been much improvement on the classification performance on this dataset. A variety of architectures and feature extraction methods have been explored, however even models that have performed well on other datasets, such as EEGNet, only achieve classification accuracy values in the 50s (Limbaga et al., 2022) (Lawhern et al., 2018). The highest classification accuracy on the dataset that has been reported was 77% using an autonomous deep learner (ADL), however that model was only tested on a subset of the subjects (Anam et al., 2020). There are also a few other EEG finger flex paradigm datasets, however these datasets were created for specific studies and have not been released publicly. Liao et al (Liao et al., 2014) measured EEG signals from 11 subjects in their right hand only, on all five finger flexes. They extracted features by applying principal component analysis to power spectral density data. The model designed to classify these signals was a SVM and had a classification accuracy of 39% when decoding individual finger movements (Xiao and Ding, 2015). Another study by Alazrai et al measured 18 subjects on all five fingers of the right hand (Alazrai et al., 2019). In this study, they distinguished between different types of finger movements, yielding four classes that corresponded to a thumb movement and two classes for each of the other fingers. Their model utilized quadratic time-frequency distribution based features and employed a SVM classification strategy on two layers - the first layer classified the finger that the signal represented and the second layer classified the specific movement. On the finger-level classification, their model achieved an accuracy of $85.85 \pm 1.1\%$, far higher than any other model, however, this study utilized more data than the other models which may have contributed to the impressive results.

Model Architecture choice

Most classifiers for the finger flexion paradigm have been standard machine learning models, such as SVMs and RF models. There have been a few deep learning architectures applied to finger flex classification, such as EEGNet, which is a CNN, and the ADL described in Anam et al 2020. There have not been any studies that utilize a transformer based

Table 1: Studies utilizing Kaya et al. 2018

Study	Year	Feature extraction	Model Architecture	Classification Accuracy
Kaya et al.	2018	Fourier Transform	SVM	$43 \pm 10\%$
Anam et al.	2019	CSP	SVM, kNN, RF, Linear discriminant analysis	RF - 54%
Anam et al	2020	CSP	ADL	77%
Limbaga et al.	2022		EEGNet (CNN)	51.74% w/ TL
Luo et al.	2023		Siamese network	49.02%

model. A transformer is a type of deep neural net that relies on an attention mechanism to keep track of long distance dependencies in data (Vaswani et al., 2023). This architecture is the state of the art in natural language processing and computer vision applications, and is beginning to be explored in the domain of signal processing. A transformer is a suitable choice for EEG signals, since EEG signals are a type of sequence and there may be long range dependencies in it (Sun et al., 2021). Previous models show that the transformer is more effective when it is combined with a CNN that is designed for feature extraction. We take inspiration from an existing convolutional transformer to develop our own convolutional transformer model to classify finger flexions (Song et al., 2023).

4 Methods

Unlike the previous models trained on these datasets, our proposed method for classifying finger flexions does not include preprocessing and feature extraction, and instead relies on the convolutional layers of the deep neural network itself to extract features and classify them. We epoched the raw data into separate trials, starting at the cue and ending one second after the cue, which is when the cue was removed from the screen. The data is downsampled to a frequency of 200Hz for consistency, which results in 200 data points per sample. This data is directly fed into the model, which starts with a CNN that extracts temporal features. This CNN then compresses the data on the space dimension, yielding an output of shape (64, 1, 173). These extracted features become the embeddings for the transformer part of the model. The transformer consists of 8 layers with 16 attention heads in each layer. The encodings developed by the transformer are then fed to two fully connected layers that do the final classification. This model was first tested on each of the subjects independently, training on 70% of each subject's data and testing on the remaining 30%. After finding that the model performed well on individual subjects, the model was used to perform transfer learning. To do that, a "leave-one-out" strategy was used, where the model was trained on all the subjects except the test subject, and then fine-tuned on the test subject. We fine tuned on 0%, 30% and 40% of the test subjects' data, seeing improvements as more fine tuning was used.

5 Results

We evaluated our models using classification accuracy, recording the model's classification results on the five fingers for each subject, and then looking at the average results from that subject.

Single Subject

Using this convolutional transformer hybrid model, we find that we make an improvement over other deep learning models that classify on raw data. This model achieves a maximum accuracy of 70% on subject E, averaging around 56% accuracy across all subjects.

Subject	Transformer
Subj A	41.55
Subj B	56.46
Subj C	68.30
Subj E	72.04
Subj F	56.32
Subj G	52.55
Subj H	39.15
Subj I	57.81

Table 2: Classification accuracy on single subject

Subject Transfer

Since this model achieved an acceptable classification accuracy, we attempt to use this model to perform transfer learning between subjects. Although the results without fine-tuning are not that high, with 40% of the test subject's data, the fine-tuned results are almost as good as results obtained from subject dependent classification. This implies that there are underlying commonalities between different peoples' representations of finger movement motor imageries, and that there is a generalizable ground truth that underlies these motor imageries.

Subject	40% FT	30% FT	0% FT
A	43.98	39.64	35.65
B	53.96	48.99	37.65
C	66.58	55.48	39.28
E	69.05	63.79	39.96
F	54.55	49.90	31.28
G	53.37	52.07	42.29
H	39.71	41.95	36.98
I	59.95	55.81	45.59
Average	55.14	50.84	38.59

Table 3: Results with Transfer Learning

Conclusion

Our hybrid convolutional transformer model outperformed other state of the art deep learning models, indicating that the transformer architecture is well suited to the classification of finger flexions. Although this model did not outperform the ADL model, the ADL model used a feature extraction method and did not test on all of the subjects either, which means the ADL model may not be able to generalize as well past the scope of that study. More importantly, this model is able to take the raw features from each subject, and performs well on all subjects, making it easy to apply this model onto new data.

Acknowledgements

The work in this paper is supported by the National Science Foundation under grant No. 2050919. Any opinions, findings, conclusions, or recommendations expressed in this work do not necessarily reflect the views of the National Science Foundation.

References

Alazrai, R., Alwanni, H., and Daoud, M. I. (2019). EEG-based BCI system for decoding finger movements within the same hand. *Neuroscience Letters*, 698:113–120.

Altaheri, H., Muhammad, G., Alsulaiman, M., Amin, S. U., Altuwaijri, G. A., Abdul, W., Bencherif, M. A., and Faisal, M. (2021). Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: a review. *Neural Comput & Applic*.

Anam, K., Bukhori, S., Hanggara, F. S., and Pratama, M. (2020). Subject-independent Classification on Brain-Computer Interface using Autonomous Deep Learning for finger movement recognition. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 447–450. ISSN: 2694-0604.

Hekmatmanesh, A., Wu, H., Jamaloo, F., Li, M., and Handroos, H. (2020). A combination of CSP-based method with soft margin SVM classifier and generalized RBF kernel for imagery-based brain computer interface applications. *Multimed Tools Appl*, 79(25):17521–17549.

Kaya, M., Binli, M. K., Ozbay, E., Yanar, H., and Mishchenko, Y. (2018). A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces. *Sci Data*, 5(1):180211. Number: 1 Publisher: Nature Publishing Group.

Khademi, Z., Ebrahimi, F., and Kordy, H. M. (2023). A review of critical challenges in MI-BCI: From conventional to deep learning methods. *Journal of Neuroscience Methods*, 383:109736.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. (2018). EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. *J. Neural Eng.*, 15(5):056013. arXiv:1611.08024 [cs, q-bio, stat].

Liao, K., Xiao, R., Gonzalez, J., and Ding, L. (2014). Decoding Individual Finger Movements from One Hand Using Human EEG Signals. *PLOS ONE*, 9(1):e85192. Publisher: Public Library of Science.

Limbaga, N. J., Mallari, K. L., Yeung, N. R., and Monje, J. C. (2022). Development of an EEG-based Brain-Controlled System for a Virtual Prosthetic Hand. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1714–1717.

Song, Y., Zheng, Q., Liu, B., and Gao, X. (2023). EEG Conformer: Convolutional Transformer for EEG Decoding and Visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:710–719. Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

Sun, J., Xie, J., and Zhou, H. (2021). EEG Classification with Transformer-Based Models. In *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, pages 92–93.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention Is All You Need. arXiv:1706.03762 [cs].

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791.

Xiao, R. and Ding, L. (2015). EEG resolutions in detecting and decoding finger movements from spectral analysis. *Front Neurosci*, 9:308.

Applications of PSO-Based Dimension Reduction and Subject-Transfer in Motor Imagery Brain Computer Interfaces

Marios Petrov
New College of Florida
5800 Bay Shore Rd
Sarasota, Florida 34243
marios.petrov23@ncf.edu

Adham Atyabi
University of Colorado, Colorado Springs
1420 Austin Bluffs Parkway
Colorado Spring, Colorado 80918
aatyabi@uccs.edu

Abstract

Brain-Computer Interfaces (BCIs) have demonstrated immense potential in a myriad of applications, from neuroprosthetics to rehabilitation. However, the practical deployment of BCIs is often impeded by the significant time and resources required to train the classifier on individual subject data. This challenge necessitates efficient solutions that maintain high performance while reducing the time and effort needed for model training. This study addresses this issue by exploring the use of transfer learning in conjunction with Wavelet Packet Decomposition (WPD), Common Spatial Patterns (CSP), and Particle Swarm Optimization (PSO) for EEG classification. The proposed methodology is rigorously evaluated on three renowned motor imagery (MI) datasets from the BCI competitions III and IV. The preprocessing pipeline draws inspiration from the Delorme 2023 approach, enhanced by four additional preliminary preprocessing steps. CSP and WPD are employed for feature extraction, and PSO is deployed for dimension reduction. The proposed approach exhibits superior performance compared to state-of-the-art models, not only on transfer learning-related tasks but also single subject classification on all three datasets. The results underscore the effectiveness of pre-trained models and subject transfer learning in significantly reducing model training time without compromising accuracy, offering promising advancements in real-world BCI applications.

1 Introduction

As a transformative technology at the nexus of neuroscience, computer science, and engineering, Brain-Computer Interfaces (BCIs) have begun to profoundly reshape numerous fields, including but not limited to neuroprosthetics and rehabilitation. BCIs work by forging a direct communication pathway between the brain and external devices, thereby transforming brain activity into control signals for these devices. This innovation has profound implications, especially for individuals with severe motor impairments, as it provides an alternative conduit for communication and control that does not rely on the body's peripheral nerves and muscles. BCIs, therefore, herald a new level of independence for such individuals, offering them the ability to interact with the world around them in ways that were previously unimaginable. Yet, the reach of BCIs is not confined to those with motor impairments. They present a vast horizon of possibilities for augmenting human cognition, elucidating the intricacies of brain processes, and even transforming entertainment through gaming applications. The potential for BCIs to revolutionize cognitive training and provide neurofeedback solutions accentuates their societal significance, thus making the advancement of BCI technology a pressing priority in scientific and engineering research.

However, the broader deployment of BCIs often encounters hurdles primarily due to the time and resource demands associated with training classifiers on subject-specific data. The complexity of brain signals, their susceptibility to non-stationarity and noise, and inter-individual variability necessitate sophisticated machine learning algorithms that often require significant computational resources and time for training.

In the context of EEG-based BCIs, these challenges are further amplified. EEG signals, which capture the brain's electrical activity via electrodes placed on the scalp, exhibit high temporal resolution but comparatively lower spatial resolution. They are typically characterized by a high degree of noise and variability, both within and between subjects, which compounds the difficulty of training models on subject-specific data. This task not only places a heavy computational burden but also incurs practical challenges such as participant fatigue and frustration, alongside financial implications associated with individual calibration sessions. These factors limit the broader adoption of BCI systems, thereby highlighting the pressing need for effective methodologies that can maintain high performance while reducing the resource-intense requirements of model training.

Subject transfer learning, a strategy where a model pre-trained on data from some subjects is applied to new, unseen subjects, has slowly been emerging as a promising solution for EEG-based BCIs. By leveraging knowledge from pre-existing subjects to new subjects, the time-consuming model training process can be significantly reduced. However, for this strategy to be effective, it necessitates a robust approach incorporating signal processing, feature extraction, and dimension reduction techniques. Currently, subject transfer methods tend to trend towards lower performance when compared to experiments.

This paper proposes an integration of subject transfer learning with Wavelet Packet Decomposition (WPD), Com-

mon Spatial Patterns (CSP), and Particle Swarm Optimization (PSO). WPD, a time-frequency signal analysis tool, is used to dissect EEG signals into different frequency bands. CSP, a spatial filtering technique commonly employed in BCI applications, is then used for feature extraction, which enables the differentiation between mental tasks based on the spatial patterns in the EEG signals. Finally, PSO, a population-based stochastic optimization technique, is implemented for dimension reduction in the feature space, and performance optimization.

The proposed methodology is rigorously evaluated on three renowned motor imagery (MI) datasets from the BCI competitions III and IV. The preprocessing pipeline is based on the Delorme 2023 approach, supplemented by four additional preliminary preprocessing steps to ensure data quality. The results indicate an improved performance compared to state-of-the-art models, in both transfer learning-related tasks and single subject classification.

2 Related Works

2.1 Evolutionary Dimension Reduction

Plenty of work in the domain of seizure prediction using EEG classification, has demonstrated the potential of particle swarm optimization(PSO)-based methods as a means of distilling down feature sets to their most representative constituents using PSO-filtering or binary masking(1)(2).

The proposed dimension reduction methodology is inspired by previous results demonstrating the efficacy of evolutionary-based approaches in the realm of BCI (3). However unlike previous PSO implementations which reduce dimensionality in the spatial and spectral dimension(4)(5), our method goes a step further and also reduces dimensionality in the temporal dimension, due to the utilization of the Wavelet Packet Decomposition as opposed to the Fourier Transform.

2.2 Subject Transfer Classification

Kang et al. 2009(6) pioneered the use of CSP to form covariance matrices from various combinations of subjects, thereby facilitating subject-to-subject transfer. They evaluated numerous training and testing splits for each subject, and achieved an impressive average CV accuracy of 75%.

Following up, Atyabi et al. 2013(4) delved into the use of evolutionary algorithms, seeking to minimize training time while enhancing performance for 0-trial subject transfer classification and fine-tuning subject transfer classification. The 0-trial protocol they adopted mirrors ours: a classifier is trained on every subject bar one test subject and is then entirely assessed on this test subject. The fine-tuning subject transfer is an extension of the 0-trial, whereby the classifier is additionally trained on a small subset of the test subject's data. Their findings, indicating no significant difference after employing a 40% fine-tuning set, align closely with our results.

Uran et al. 2019(7) explored the interplay between various learning methods and deep models to gauge subject transfer performance. In their study, they scrutinized "Standard Learning","Distributed Learning", "Split Learning", and "Frozen Learning". Each method systematically adjusted which sessions and subjects the model was trained and tested on. Notably, their "Frozen Learning" approach, which involved freezing layers during training, reached an accuracy of 77%.

The 2022 studies by Zaremba et al.(8) and Theng et al.(9) both employed 2D convolutional neural networks and adopted transfer learning protocols akin to ours (0-trial and fine-tuning). However, Zaremba et al. also experimented with Cross-Dataset subject transfer, whereby a classifier was trained on data from all subjects in one dataset and tested on each subject within another. This method yielded state-of-the-art results and bolstered the notion of a universally shared motor imagery construct across brains.

Finally, Weit et al. 2023(10) implemented Multi-Source Transfer Joint Matching. They achieved this by mapping each subject's spatial covariance matrices in a tangent space using a compounded centroid matrix, coupled with ensemble methods. Their findings were comparable to those of Zaremba et al. 2022, further strengthening the understanding of subject transfer classification.

2.3 Current State-of-the-Art

In order to robustly evaluate the effectiveness of the proposed method, we conducted comparative analyses on two distinct experimental paradigms: single-subject classification and subject transfer classification.

For the single-subject paradigm, our method's performance was contrasted against five contemporary, state-of-the-art studies for each dataset employed. This comparison helped us gauge where our method stands relative to existing techniques, and to discern any potential areas of improvement.

For the subject transfer paradigm, we juxtaposed our results against recent influential studies that have explored the realm of transfer learning applications in the context of BCI. This comparison for each respective dataset provided an assessment of our method's performance within the specific ambit of subject transfer tasks.

By holding our method up against these current benchmarks in the field, we aimed to provide a comprehensive picture of its relative strengths and weaknesses, ultimately contributing to the ongoing dialogue about optimizing BCI performance and accessibility.

2.3.1 Single Trial EEG The proposed method displayed significant strides in performance over the existing state-of-the-art approaches in the realm of single subject EEG (STE) classification. Particularly on the BCI IV 1 dataset, our method attained an impressive 5-fold cross-validation accuracy of 93.1%, surpassing the current benchmark of 84.7%. Similarly, the BCI III 4a dataset showed remarkable improvement with our method achieving an accuracy of 99.0%, outshining the state-of-the-art accuracy of 95.3%.

In contrast, on the BCI IV 2a dataset, the performance of our method remained in line with the current state-of-the-art. Our PSO+WPD+CSP+SVM algorithm achieved an accuracy of 95.4%, marginally lower but comparable to the current best accuracy of 96.7%. This consistent performance

Table 1: STE Classification State-of-the-Art Accuracy

BCI IV 1 Single Trial EEG State-of-the-Art						
Subject	Reference/Method					
ID	(8)	(11)	(12)	(13)	(14)	Ours
A	0.816	0.669	0.855	0.881	0.874	**0.958**
B	0.717	0.652	0.670	0.591	0.700	**0.984**
C	0.846	0.824	NaN	0.679	0.674	**0.900**
D	0.888	**0.946**	NaN	0.843	0.929	0.835
E	0.936	**0.949**	NaN	0.902	0.934	0.895
F	0.689	0.843	0.795	0.859	0.888	**0.954**
G	0.932	0.812	0.945	0.922	0.932	**0.991**
Avg.	0.832	0.813	0.816	0.811	0.847	**0.931**
BCI IV 2a Single Trial EEG State-of-the-Art						
Subject	Reference/Method					
ID	(15)	(16)	(17)	(18)	(19)	Ours
A01	**0.997**	0.868	0.898	0.854	0.906	0.939
A02	**0.967**	0.646	0.693	0.707	0.660	0.952
A03	**0.981**	0.958	0.910	0.952	0.951	0.937
A04	0.917	0.674	0.769	0.803	0.781	**0.957**
A05	0.932	0.681	0.602	0.703	0.800	**0.954**
A06	**0.981**	0.674	0.682	0.684	0.625	0.947
A07	**0.996**	0.806	0.896	0.910	0.917	0.980
A08	0.992	0.972	0.902	0.864	0.889	**0.995**
A09	0.942	0.924	0.871	0.845	0.830	**0.924**
Avg.	**0.967**	0.800	0.803	0.813	0.818	0.954
BCI III 4a Single Trial EEG State-of-the-Art						
Subject	Reference/Method					
ID	(8)	(20)	(18)	(21)	(22)	Ours
AA	0.810	0.818	0.795	0.935	**1.0**	0.988
AL	0.941	0.961	1.0	**0.975**	0.741	0.973
AV	0.641	0.725	0.725	0.937	0.679	**0.988**
AW	0.921	0.921	0.955	0.963	0.901	**1.0**
AY	0.936	0.903	0.885	0.955	0.893	**1.0**
Avg.	0.850	0.866	0.861	0.953	0.845	**0.990**

Table 2: Subject Transfer Classification State-of-the-Art Accuracy

BCI IV 1 Subject Transfer EEG State-of-the-Art					
Test	Reference/Method				
Subject	50%(9)	50%(8)	(10)	Ours	
A	0.593	0.873	0.895	**0.901**	
B	0.709	0.784	0.790	**0.961**	
C	0.615	0.758	0.832	**0.845**	
D	0.595	0.879	**0.984**	0.777	
E	**0.963**	0.936	0.686	0.813	
F	0.575	**0.985**	0.984	0.914	
G	0.872	**1.0**	0.817	0.947	
Avg.	0.703	0.844	**0.855**	0.880	
BCI IV 2a Subject Transfer EEG State-of-the-Art					
Test	Reference/Method				
Subject	50%(9)	50%(8)	(17)	(7)	Ours
A01	0.651	0.818	**0.898**	NaN	0.869
A02	0.488	0.485	0.694	NaN	**0.897**
A03	0.724	0.788	**0.910**	NaN	0.854
A04	NaN	0.622	0.769	NaN	**0.907**
A05	0.429	0.572	0.602	NaN	**0.907**
A06	0.348	0.633	0.682	NaN	**0.887**
A07	0.475	0.769	0.896	NaN	**0.947**
A08	0.418	0.674	0.902	NaN	**0.950**
A09	0.708	0.807	**0.871**	NaN	0.856
Avg.	0.530	0.718	0.803	0.770	**0.897**
BCI III 4a Subject Transfer EEG State-of-the-Art					
Test	Reference/Method				
Subject	50%(9)	50%(8)	(5)	(6)	Ours
AA	0.661	0.756	0.860	0.6825	**0.978**
AL	0.898	**1.0**	0.780	0.966	0.957
AV	0.649	0.708	0.780	0.613	**0.936**
AW	0.868	0.988	0.750	0.702	**0.994**
AY	0.696	**1.0**	0.750	0.787	**1.0**
Avg.	0.754	0.858	0.784	0.750	**0.973**

further bolsters the effectiveness and robustness of our approach across diverse datasets.

2.3.2 Subject Transfer EEG Given the scarcity of published work that attempts to address the challenge of subject transfer generalizability using our fine tuning method, we decided to benchmark our results against a diverse collection of transfer learning methodologies. This approach allowed us to assess our method's performance in a broader context, and to identify and highlight the strongest performances from each respective study.

Despite the varied approaches to subject transfer, our method demonstrated a consistently superior performance across all datasets. The 5-fold cross-validation accuracies attained were 88.0%, 89.7%, and 97.3% on the BCI IV 1, BCI IV 2a, and BCI III 4a datasets, respectively. These results significantly outperformed the previous state-of-the-art accuracies, which stood at 85.5%, 80.3%, and 85.8% for the same datasets. This indicates not only the efficacy of our approach but also its robustness when applied to different datasets. Furthermore, when comparing our work to methodologies that utilized fine-tuning we used 10% less of the test subject's data, furthering the real world applicability of this approach.

4 Methods

4.1 Datasets

Three motor imagery (MI) datasets from the BCI competitions III and IV were utilized to evaluate the proposed methods. These include BCI IV 1, BCI IV 2a, and BCI III 4a. Both IV 1 and IV 2a datasets comprise two classes: a foot class and a left/right-hand class. On the other hand, IV 2a is a 4-class dataset, incorporating left/right hand classes, a tongue class, and a feet class. However, to maintain consistency across datasets, only the left/right-hand classes and the feet class are utilized from BCI IV 2a. Before commencing the formal preprocessing, IV 1 and III 4a were downsampled to a frequency of 250Hz. Furthermore, only specific channels were retained from each dataset: Fz, FC3, FC1, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4, P1, Pz, P2.

Table 3: Datasets Used

Dataset	Classes	Sample Rate (Hz)	Channels (10-20)	Subjects
BCI IV 1	2	1000	64	7
BCI IV 2a	4	250	22+3EoG	9
BCI III 4a	2	1000	118	5

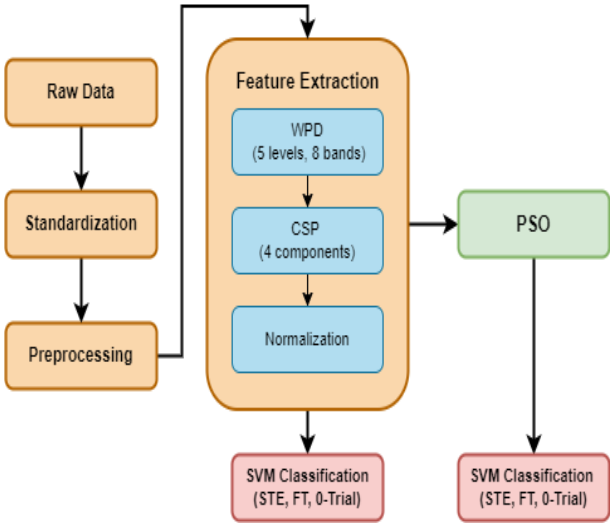


Figure 1: Overarching methodology employed for each of the three experimental paradigms: single trial eeg (STE), subject transfer with fine tuning (FT), and subject transfer with no fine tuning (0-trial).

4.2 Preprocessing

The data preprocessing methodology adopted in this research drew its inspiration from the approach outlined in (23) . A deviation was introduced in the form of a modified threshold for Independent Component Analysis (ICA), which was reduced to 80% compared to the initial 90%. This adjustment was implemented to increase the sensitivity of artifact detection in the EEG data. Furthermore, an enhancement to the initial methodology was made by integrating four supplementary preprocessing steps prior to the automated EEG data cleansing process.

4.3 Feature Extraction

Feature extraction plays an integral role in the process of EEG signal classification, as it enables the translation of raw EEG data into a meaningful set of features that are representative of the underlying mental tasks. It is during this step that discriminative information is extracted from the preprocessed EEG signals. This information encapsulates distinguishing characteristics of the different classes of motor imagery tasks and is crucial for the subsequent classification.

For EEG-based BCI systems, feature extraction is particularly crucial given the complex, multidimensional nature of EEG data. The high dimensionality, combined with a relatively lower signal-to-noise ratio, necessitates a robust feature extraction strategy that can distil the relevant information from the raw signals while filtering out noise and irrelevant components. This step is, therefore, instrumental in enhancing the performance of the classifier. The proposed methodology employs a combination of Common Spatial Patterns (CSP) and Wavelet Packet Decomposition (WPD) for feature extraction.

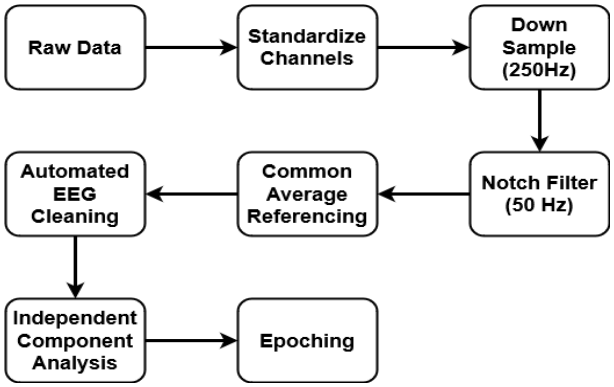


Figure 2: Preprocessing pipeline, implemented in MATLAB using EEGLAB (24), details the steps applied to each dataset. Notably, BCI IV 2a did not require notch filtering and down-sampling as these steps were carried out during data acquisition itself.

4.3.1 Common Spatial Patterns CSP is a popular method in BCI applications for the processing of EEG signals(22)(16). The goal of CSP is to yield spatial filters that provide maximal discriminative power between two different motor imagery tasks. CSP accomplishes this by creating spatial filters that maximize the variance of the signal from one class while minimizing the variance from the other.

The mathematical process behind CSP involves several steps. First, let's assume we have two sets of multivariate EEG signals $X_1 \in \mathbb{R}^{C\times T}$ and $X_2 \in \mathbb{R}^{C\times T}$, where C is the number of channels and T is the number of time points. These two sets represent two classes of motor imagery tasks.

The covariance matrices of the two sets are calculated as follows:

$$\Sigma_1 = \frac{1}{T} X_1 X_1^T \quad \Sigma_2 = \frac{1}{T} X_2 X_2^T$$

Then, we compute the composite spatial covariance as:

$$\Sigma_c = \Sigma_1 + \Sigma_2$$

Next, we perform an eigenvalue decomposition of Σ_c:

$$\Sigma_c = UPU^T$$

where P is a diagonal matrix of eigenvalues and U is a matrix of corresponding eigenvectors. We can find the whitening transformation matrix $P^{-\frac{1}{2}}U^T$. The covariance matrices Σ_1 and Σ_2 are then whitened:

$$\Sigma_1^{whitened} = P^{-\frac{1}{2}}U^T\Sigma_1 U P^{-\frac{1}{2}} \quad \Sigma_2^{whitened} = P^{-\frac{1}{2}}U^T\Sigma_2 U P^{-\frac{1}{2}}$$

Given that $\Sigma_1^{whitened} + \Sigma_2^{whitened} = I$, we only need to consider one whitened covariance matrix for the subsequent generalized eigenvalue problem, say $\Sigma_1^{whitened}$:

$$\Sigma_1^{whitened} W = W\Lambda$$

where Λ is a diagonal matrix of generalized eigenvalues and W is the matrix of corresponding eigenvectors. Each column of W is a CSP spatial filter and can be used to project the original EEG signals into the CSP feature space. Thus, the variance of the projected signals would be maximally different for the two classes of motor imagery tasks.

The output of the CSP operation is usually the log-variance of the filtered signals, which are then used as features for the classifier:

$$features = log(var(W^T X))$$

In our implementation, CSP is utilized with four components, no regularization, log-transformed, and no trace normalization

4.3.2 Wavelet Packet Decomposition Wavelet Packet Decomposition (WPD) is a signal processing technique that provides a flexible time-frequency representation of signals. It is an extension of the Wavelet Transform (WT), a method that decomposes a signal into a series of wavelets. Unlike WT, which only applies wavelets to the approximation coefficients, WPD applies wavelets to both the detail and approximation coefficients, resulting in a richer and more detailed representation of the original signal(25)(26)(27).

WPD relies on the use of wavelets, which are localized wave functions obtained from a single prototype function known as a mother wavelet $\psi(t)$. The mother wavelet is scaled and shifted in time to generate a family of wavelets represented by $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$, where j and k denote the scale and translation parameters, respectively.

Given a signal $X(t)$, WPD involves computing inner products between the signal and the wavelets at various scales and positions, resulting in a set of wavelet coefficients. The decomposition can be represented mathematically as follows:

$$X(t) = \sum_{k=0}^{N-1} c_{j,k}\psi_{j,k}(t)$$

where $\psi_{j,k}(t)$ represents the wavelet packet functions, which are constructed by translating and dilating the mother wavelet $\psi(t)$. Here, j is an integer that determines the scale of the wavelet (i.e., the width of the wavelet in the time domain), and k is an integer that determines the position of the wavelet. The coefficients $c_{j,k}$ are given by the inner product of the signal with the wavelet packet functions:

$$c_{j,k} = \langle X(t), \psi_{j,k}(t)\rangle = \int X(t)\psi_{j,k}(t)dt$$

In the context of this research, a Daubechies 10 (db10) wavelet is used, and a 5-level decomposition is performed; these parameters were deduced experimentally and other wavelets and decomposition levels can be viewed in the supplementary materials. At each level of the decomposition, the coefficients represent the signal information at different frequency bands. By extracting the coefficients from the 5th level, we obtain 8 sets of coefficients, each providing information about a specific frequency band of the original signal. These coefficients are then truncated or zero-padded to ensure a fixed length for further analysis.

WPD is paired with CSP to extract discriminating spatial and spectral features. The CSP operation is first applied to the EEG signals to obtain spatial filters that maximize the variance for different motor imagery tasks. The filtered signals are then processed with WPD to decompose them into different frequency bands. The resulting features represent the log-variance (from CSP) of the different frequency band signals (from WPD) and can provide effective discriminative power for motor imagery classification tasks(28)(29).

4.4 Dimension Reduction

Dimension reduction, also known as feature selection, is a broad term encompassing numerous techniques designed to reduce the dimensionality, complexity, or overall size of a dataset(30). Within the specific realm of EEG signals, several methods have exhibited promising results. However, Particle Swarm Optimization (PSO) and other evolutionary strategies hold particular intrigue. Their versatility allows for simultaneous dimensionality reduction across spectral, spatial, and temporal domains, making them uniquely effective.

In our study, we employ an adapted version of PSO, which we call Multi-Dimensional Particle Swarm Reduction (MDPSR) as a means to generate feature filters. Our methodology to generate these filters expands upon the work proposed in Atyabi 2017(5). The output filters from the PSO reduction function as masks that apply matrix multiplication of 1 or 0 to nullify features within each patient's corresponding feature vector. By honing in on the most informative features, these filters facilitate the classifier to operate with improved efficiency and heightened performance(4).

4.4.1 Particle Swarm Optimization (PSO) Particle Swarm Optimization (PSO) is a bio-inspired optimization method that manipulates a population of potential solutions, known as particles, to iteratively improve upon a candidate solution. This population-based stochastic optimization algorithm draws inspiration from the social behavior observed in bird flocks or fish schools. Each particle adjusts its position in the search space according to its own best known position (pbest) and the best known positions of the swarm (gbest), creating a balance between exploration and exploitation.

Mathematically, the movement of a particle i at time t is governed by the following equations:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot rand_1 \cdot (pbest_i^t - x_i^t) + c_2 \cdot rand_2 \cdot (gbest^t - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

Here, v_i^t represents the velocity of particle i at time t, x_i^t is the position of the particle in the search space, w is the in-

ertia weight, c_1 and c_2 are acceleration coefficients that control the influence of the personal and global best positions, respectively, and $rand_1$ and $rand_2$ are random numbers in $[0, 1]$. The parameter w can be considered as a trade-off between global (wide-ranging) and local (nearby) exploration of the search space.

4.4.2 Multi-Dimensional Particle Swarm Reduction (MDPSR) Building upon the standard PSO, we introduce the Multi-Dimensional Particle Swarm Reduction (MDPSR). The main distinction of MDPSR is its ability to handle multi-dimensional data, thus making it particularly suitable for the context of EEG signals. Previous PSO-Based methods have demonstrated efficacy in reducing dimensionality in spectral and spatial domains(3), whereas our method through the utilization of CSP and WPD, is inadvertently performing reduction in all three domains of EEG data (spectral, spatial, and temporal). MDPSR is algorithmically expressed as the following:

Employed MDPSR Hyper-Parameters

- n_particles: 80
- n_Iterations: 200
- w_max: 1
- w_min: 0.9
- c1: 2
- c2: 2

Algorithm 1 Multi-Dimensional Particle Swarm Reduction (MDPSR)

Require: Objective function, Number of particles (n), Number of iterations (m), Maximum Inertia weight (w_{max}), Minimum Intertia weight (w_{min}), Personal learning factor ($c1$), Social learning factor ($c2$)
Ensure: Global best position and score
Initialize each particle's position randomly using n-dimensional binary vector
Initialize each particle's velocity randomly between -1 and 1
Set each particle's best position to its initial position
Set each particle's best score and global best score to $-\infty$
for $i = 1$ to m **do**
    for each particle **do**
        Create a mask from particle's position
        if no elements in mask **then**
            Set the index of maximum velocity in mask to True
        end if
        Apply mask to features to get reduced features
        Compute score using the objective function with reduced features and labels
        if score $>$ particle's best score **then**
            Update particle's best position and score
            if score $>$ global best score **then**
                Update global best position and score
            end if
        end if
    end for
    Compute inertia weight $w = w_{\text{max}} - i \times \frac{w_{\text{max}} - w_{\text{min}}}{m}$
    for each particle **do**
        Compute new velocity: $v_{\text{new}} = w \times v_{\text{old}} + c1 \times \text{rand}() \times (\text{pbest} - \text{position}) + c2 \times \text{rand}() \times (\text{gbest} - \text{position})$
        Compute new position: if rand() $< \frac{1}{1+e^{-v_{\text{new}}}}$, then position = 1, else position = 0
        if no elements in new position **then**
            Set the index of maximum velocity in new position to 1
        end if
    end for
end for
return global best position and score

4.5 Experiments and Results

The results of our study were generated from three distinctive experiments, each with different protocols:

– **Single Trial EEG (STE):** This approach involves training a classifier on a subset (80% in our case) of a single subject's EEG data and testing it on the remaining data, thereby creating a unique classifier for each respective subject. This paradigm is widely adopted in current real-world BCI implementations. We observed that both the CSP+SVM and WPD+CSP+SVM methods outperformed their earlier counterparts. Additionally, our inclusion of MDPSR further augmented this improved performance, generating approximately an 8-12% boost in classification performance for both feature extraction methods (CSP and WPD+CSP). Remarkably, the MDPSR-enhanced WPD+CSP method demonstrated superior performance on the BCI IV 1 and BCI III 4a datasets and held its own against the state-of-the-art on the BCI IV 2a. As evaluation metrics, we employed 5-fold cross-validation (CV) accuracy and F1 score

– **Subject Transfer with Zero Trial EEG (ST0):** In this paradigm, a classifier is trained on data from every subject within a dataset, excluding a single "Target Subject". This pre-trained classifier is then tested on all of the Target Subject's data. This process is iteratively repeated for each subject within each dataset. Due to the structure of this approach,

cross-validation is not applicable, thus, we evaluated classifiers using standard test accuracy. Implementing MDPSR in this context reduced the size of each pre-trained target subject's model by approximately 50-70%. However, performance between the MDPSR-enhanced WPD+CSP+SVM and the standard WPD+CSP+SVM was found to be similar on average.

– **Subject Transfer with Fine-Tuning (STF):** Our final experiment followed the same procedure as ST0, but introduced a fine-tuning step on the Target Subject's data. We explored different levels of fine-tuning (15%, 25%, 40%, 80%). Remarkably, results from the MDPSR-enhanced method at a 25% fine-tuning level matched those obtained by WPD+CSP+SVM without MDPSR at 40%. This suggests that MDPSR not only contributes directly to reducing dimensionality, but also reduces the quantity of data required for fine-tuning.

Figure 3: Single Trial EEG Classification Performance using 5-fold CV

BCI IV 1 Single Trial EEG

Subject ID	CSP+SVM Accuracy	CSP+SVM F1-score	WPD+CSP+SVM Accuracy	WPD+CSP+SVM F1-score
A	**0.754**	**0.751**	0.916	0.915
B	0.701	0.700	**0.949**	**0.949**
C	0.615	0.610	0.850	0.849
D	0.650	0.647	0.800	0.799
E	0.640	0.636	0.840	0.839
F	0.653	0.650	0.907	0.906
G	0.654	0.647	0.947	0.946
Avg.	0.667	0.663	0.887	0.886

BCI IV 1 Single Trial EEG

Subject ID	MDPSR+CSP+SVM Accuracy	MDPSR+CSP+SVM F1-score	MDPSR+WPD+CSP+SVM Accuracy	MDPSR+WPD+CSP+SVM F1-score
A	**0.754**	**0.747**	0.958	0.958
B	0.701	0.700	0.984	0.985
C	0.630	0.624	0.900	0.900
D	0.650	0.647	0.835	0.834
E	0.650	0.643	0.895	0.895
F	0.658	0.657	0.954	0.953
G	0.699	0.688	**0.991**	**0.991**
Avg.	0.678	0.672	0.931	0.931

BCI IV 2a Single Trial EEG

Subject ID	CSP+SVM Accuracy	CSP+SVM F1-score	WPD+CSP+SVM Accuracy	WPD+CSP+SVM F1-score
A01	**0.727**	**0.714**	0.889	0.890
A02	0.633	0.631	0.914	0.915
A03	0.647	0.638	0.879	0.878
A04	0.517	0.500	0.913	0.914
A05	0.668	0.648	0.903	0.901
A06	0.579	0.573	0.904	0.902
A07	0.652	0.649	**0.965**	**0.965**
A08	**0.864**	**0.863**	**0.980**	**0.980**
A09	0.744	0.742	0.887	0.885
Avg.	0.670	0.662	0.915	0.914

BCI IV 2a Single Trial EEG

Subject ID	MDPSR+CSP+SVM Accuracy	MDPSR+CSP+SVM F1-score	MDPSR+WPD+CSP+SVM Accuracy	MDPSR+WPD+CSP+SVM F1-score
A01	0.732	0.726	0.939	0.939
A02	0.633	0.631	0.952	0.952
A03	0.672	0.653	0.937	0.938
A04	0.527	0.516	0.957	0.957
A05	0.668	0.648	0.954	0.953
A06	0.579	0.573	0.947	0.947
A07	0.662	0.652	0.980	0.980
A08	**0.864**	**0.863**	**0.995**	**0.995**
A09	0.744	0.742	0.925	0.924
Avg.	0.676	0.667	0.954	0.954

BCI III 4a Single Trial EEG

Subject ID	CSP+SVM Accuracy	CSP+SVM F1-score	WPD+CSP+SVM Accuracy	WPD+CSP+SVM F1-score
AA	0.852	0.851	0.958	0.958
AL	**0.914**	**0.914**	0.941	0.941
AV	0.704	0.688	0.964	0.964
AW	0.836	0.825	0.909	0.907
AY	0.867	0.859	**1.0**	**1.0**
Avg.	0.835	0.827	0.955	0.954

BCI III 4a Single Trial EEG

Subject ID	MDPSR+CSP+SVM Accuracy	MDPSR+CSP+SVM F1-score	MDPSR+WPD+CSP+SVM Accuracy	MDPSR+WPD+CSP+SVM F1-score
AA	0.852	0.851	0.988	0.988
AL	0.923	0.923	0.973	0.973
AV	0.741	0.726	0.988	0.988
AW	0.873	0.869	**1.0**	**1.0**
AY	**0.933**	**0.931**	**1.0**	**1.0**
Avg.	0.865	0.860	0.990	0.990

5 Discussion and Conclusion

For each conducted experiment, we evaluated our feature extraction and classification methodology both with and without the inclusion of Multi-Dimensional Particle Swarm Reduction (MDPSR). Notably, in the context of single-trial EEG classification (STE), the MDPSR-enhanced method outperformed the WPD+CSP+SVM approach across every dataset, and indeed, on every subject within each dataset.

Table 4: Best Performing Method For Each Subject

BCI IV 1 Best Methods			
Subject	Method	Accuracy	F1-score
A	WPD+CSP+PSO+SVM STE	0.958	0.958
B	WPD+CSP+PSO+SVM STE	0.984	0.985
C	WPD+CSP+PSO+SVM STE	0.900	0.900
D	WPD+CSP+PSO+SVM STE	0.835	0.834
E	WPD+CSP+PSO+SVM STE	0.895	0.895
F	WPD+CSP+PSO+SVM STE	0.954	0.953
G	WPD+CSP+PSO+SVM STE	0.991	0.991
BCI IV 2a Best Methods			
Subject	Method	Accuracy	F1-score
A01	WPD+CSP+PSO+SVM STE	0.939	0.939
A02	WPD+CSP+PSO+SVM STE	0.952	0.952
A03	WPD+CSP+PSO+SVM STE	0.937	0.938
A04	WPD+CSP+PSO+SVM STE	0.957	0.957
A05	WPD+CSP+PSO+SVM STE	0.954	0.953
A06	WPD+CSP+PSO+SVM STE	0.947	0.947
A07	WPD+CSP+PSO+SVM STE	0.980	0.980
A08	WPD+CSP+PSO+SVM STE	0.995	0.995
A09	WPD+CSP+PSO+SVM STE	0.925	0.924
BCI III 4a Best Methods			
Subject	Method	Accuracy	F1-score
AA	WPD+CSP+PSO+SVM STE	0.988	0.988
AL	WPD+CSP+PSO+SVM STE	0.973	0.973
AV	WPD+CSP+PSO+SVM STE	0.988	0.988
AW	WPD+CSP+PSO+SVM STE	1.0	1.0
AY	WPD+CSP+PSO+SVM STE	1.0	1.0

On both subject transfer experiments (0-trial and fine-tuning), MDPSR WPD+CSP+SVM was capable of achieving the same results as WPD+CSP+SVM using 15% less fine-tuning data. Compounded with the filter application of MDPSR, and the implication of doing more with less is present. Furthermore, the obtained results, being markedly superior or at par with the state of the art for each respective subject, underscore the importance and efficacy of evolutionary-based dimension reduction methodologies, even within a subject transfer setting.

Prior works have reported dimensionality reduction rates of 95-99%. Interestingly, our approach typically achieved around 75% reduction, a seemingly lower rate. Despite this, the robust 5-fold CV accuracy and F1 scores that were attained, clearly validate the potential of the MDPSR methodology.

That said, to fully realize universal BCI, additional testing with varying classes and datasets is required. Another avenue of exploration that may have potential utility in the domain of motor imagery BCI is utilizing evolutionary methods for individual subject selections, as opposed to "super subject" methodologies which involve concatenating all subjects except the test subject together. Bridging the current gap between the existing literature and the practical application of real-world systems is what this work aimed to achieve.

Figure 4: Subject Transfer Classification Performance using 5-fold CV

BCI IV 1 Subject Transfer WPD+CSP+SVM

Target	Zero-trial protocol		15% Re-tuning		25% Re-tuning		40% Re-tuning		80% Re-tuning	
Subject	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
A	**0.660**	0.657	0.764	0.752	0.859	0.858	0.858	0.857	0.890	0.889
B	0.521	0.519	**0.871**	0.870	**0.943**	**0.943**	**0.969**	**0.969**	**0.954**	**0.954**
C	0.540	0.539	0.735	0.716	0.810	0.808	0.800	0.798	0.855	0.854
D	0.570	0.566	0.595	0.532	0.690	0.680	0.795	0.793	0.805	0.804
E	0.530	0.530	0.670	0.640	0.735	0.726	0.730	0.715	0.840	0.839
F	0.606	0.604	0.798	0.795	0.881	0.879	0.891	0.889	0.891	0.890
G	0.602	0.604	0.814	0.803	0.894	0.890	0.903	0.902	0.930	0.928
Avg.	0.575	0.574	0.750	0.730	0.830	0.826	0.850	0.846	0.881	0.880

BCI IV 2a Subject Transfer WPD+CSP+SVM

Target	Zero-trial protocol		15% Re-tuning		25% Re-tuning		40% Re-tuning		80% Re-tuning	
Subject	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
A01	0.444	0.442	0.763	0.742	0.838	0.837	0.889	0.889	0.899	0.900
A02	0.262	0.255	0.838	0.836	0.805	0.800	0.890	0.890	0.910	0.910
A03	0.652	0.649	0.772	0.774	0.831	0.830	0.869	0.869	0.874	0.873
A04	0.469	0.468	0.879	0.880	0.874	0.873	0.903	0.903	0.923	0.923
A05	0.332	0.342	0.806	0.801	0.898	0.895	0.913	0.910	0.913	0.911
A06	0.541	0.541	0.799	0.793	0.832	0.829	0.842	0.838	0.914	0.912
A07	**0.776**	**0.776**	0.891	0.891	0.926	0.925	0.941	0.941	0.950	0.950
A08	0.563	0.574	**0.951**	**0.951**	**0.956**	**0.956**	**0.961**	**0.961**	**0.980**	**0.980**
A09	0.621	0.619	0.720	0.699	0.782	0.781	0.849	0.846	0.877	0.876
Avg.	0.518	0.518	0.824	0.819	0.860	0.858	0.895	0.894	0.916	0.915

BCI III 4a Subject Transfer WPD+CSP+SVM

Target	Zero-trial protocol		15% Re-tuning		25% Re-tuning		40% Re-tuning		80% Re-tuning	
Subject	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
AA	0.560	0.475	0.893	0.887	0.922	0.921	0.946	0.946	0.941	0.940
AL	0.473	0.328	0.896	0.895	0.910	0.909	0.937	0.936	0.941	0.941
AV	**0.671**	**0.627**	**0.915**	**0.914**	0.901	0.898	0.914	0.912	0.964	0.964
AW	0.527	0.377	0.836	0.801	0.891	0.876	0.909	0.902	0.909	0.907
AY	0.536	0.545	0.427	0.265	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Avg.	0.553	0.470	0.793	0.752	0.925	0.921	0.941	0.939	0.951	0.950

BCI IV 1 Subject Transfer MDPSR+WPD+CSP+SVM

Target	Zero-trial protocol		15% Re-tuning		25% Re-tuning		40% Re-tuning		80% Re-tuning	
Subject	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
A	0.513	0.512	0.848	0.847	0.886	0.886	0.901	0.901	0.938	0.938
B	0.660	0.660	**0.950**	**0.950**	**0.927**	**0.927**	**0.961**	**0.961**	0.959	0.959
C	0.535	0.535	0.779	0.778	0.816	0.815	0.845	0.845	0.895	0.895
D	0.530	0.530	0.693	0.691	0.753	0.753	0.777	0.776	0.835	0.834
E	0.655	0.654	0.809	0.809	0.784	0.783	0.813	0.813	0.860	0.860
F	0.653	0.650	0.870	0.870	0.884	0.884	0.914	0.914	0.918	0.918
G	**0.664**	**0.664**	0.849	0.844	0.925	0.924	0.947	0.947	**0.974**	**0.974**
Avg.	0.601	0.601	0.828	0.827	0.854	0.853	0.880	0.879	0.911	0.911

BCI IV 2a Subject Transfer MDPSR+WPD+CSP+SVM

Target	Zero-trial protocol		15% Re-tuning		25% Re-tuning		40% Re-tuning		80% Re-tuning	
Subject	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
A01	0.611	0.606	0.840	0.839	0.866	0.867	0.869	0.869	0.935	0.935
A02	0.267	0.267	0.850	0.850	0.870	0.870	0.897	0.896	0.933	0.933
A03	0.609	0.609	0.840	0.840	0.855	0.855	0.854	0.855	0.895	0.895
A04	0.304	0.305	0.847	0.847	0.860	0.861	0.907	0.908	0.933	0.933
A05	0.204	0.204	0.836	0.833	0.890	0.889	0.907	0.906	0.870	0.869
A06	0.459	0.461	0.829	0.829	0.848	0.847	0.887	0.887	0.900	0.900
A07	0.662	0.661	0.935	0.935	0.931	0.931	0.947	0.947	**0.980**	**0.980**
A08	0.597	0.596	**0.958**	**0.958**	**0.954**	**0.953**	**0.950**	**0.949**	0.976	0.976
A09	**0.687**	**0.676**	0.852	0.852	0.870	0.857	0.856	0.846	0.926	0.926
Avg.	0.489	0.487	0.865	0.865	0.883	0.883	0.897	0.897	0.928	0.927

BCI III 4a Subject Transfer MDPSR+WPD+CSP+SVM

Target	Zero-trial protocol		15% Re-tuning		25% Re-tuning		40% Re-tuning		80% Re-tuning	
Subject	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
AA	0.679	0.677	0.931	0.931	0.913	0.913	0.978	0.978	0.947	0.947
AL	0.419	0.418	0.917	0.917	0.947	0.947	0.957	0.957	0.956	0.955
AV	0.488	0.388	0.880	0.880	0.932	0.932	0.936	0.936	0.976	0.976
AW	0.745	0.745	**0.940**	**0.940**	0.957	0.957	0.994	0.994	**1.0**	**1.0**
AY	**0.714**	**0.719**	0.625	0.481	**0.981**	**0.981**	**1.0**	**1.0**	**1.0**	**1.0**
Avg.	0.609	0.609	0.859	0.830	0.946	0.946	0.973	0.973	0.976	0.976

6 Acknowledgements

The work reported in this paper is supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings, and conclusions, or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Bibliography

[1] S. T. George, M. S. P. Subathra, N. J. Sairamya, L. Susmitha, and M. Joel Premkumar, "Classification of epileptic EEG signals using PSO based artificial neural network and tunable-q wavelet transform," vol. 40, no. 2, pp. 709–728.

[2] S. K. Satapathy, S. Dehuri, and A. K. Jagadev, "EEG signal classification using PSO trained RBF neural network for epilepsy identification," vol. 6, pp. 1–11.

[3] A. Atyabi, M. Luerssen, S. Fitzgibbon, and D. M. W. Powers, "Evolutionary feature selection and electrode reduction for EEG classification," in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8. ISSN: 1941-0026.

[4] A. Atyabi, M. H. Luerssen, and D. M. W. Powers, "PSO-based dimension reduction of EEG recordings: Implications for subject transfer in BCI," vol. 119, pp. 319–331.

[5] A. Atyabi, M. Luerssen, S. P. Fitzgibbon, T. Lewis, and D. M. W. Powers, "Reducing training requirements through evolutionary based di-

mension reduction and subject transfer," vol. 224, pp. 19–36.

[6] H. Kang, Y. Nam, and S. Choi, "Composite common spatial pattern for subject-to-subject transfer," vol. 16, no. 8, pp. 683–686. Conference Name: IEEE Signal Processing Letters.

[7] A. Uran, C. van Gemeren, R. van Diepen, R. Chavarriaga, and J. d. R. Millán, "Applying transfer learning to deep learned models for EEG analysis."

[8] T. Zaremba and A. Atyabi, "Cross-subject & cross-dataset subject transfer in motor imagery BCI systems," in *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. ISSN: 2161-4407.

[9] D. Theng and A. Atyabi, "Implication of subject transfer in motor imagery brain computer interfacing systems," in *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. ISSN: 2161-4407.

[10] F. Wei, X. Xu, T. Jia, D. Zhang, and X. Wu, "A multi-source transfer joint matching method for inter-subject motor imagery decoding," vol. 31, pp. 1258–1267. Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[11] L. Qian, Z. Feng, H. Hu, and Y. Sun, "A novel scheme for classification of motor imagery signal using stockwell transform of CSP and CNN model," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3673–3677. ISSN: 2577-1655.

[12] J. Jin, Y. Miao, I. Daly, C. Zuo, D. Hu, and A. Cichocki, "Correlation-based channel selection and regularized feature optimization for MI-based BCI," vol. 118, pp. 262–270.

[13] S. Kumar, K. Mamun, and A. Sharma, "CSP-TSM: Optimizing the performance of riemannian tangent space mapping using common spatial pattern for MI-BCI," vol. 91, pp. 231–242.

[14] Y. Park and W. Chung, "Frequency-optimized local region common spatial pattern approach for motor imagery classification," vol. 27, no. 7, pp. 1378–1388. Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[15] N. Alizadeh, S. Afrakhteh, and M. R. Mosavi, "Multi-task EEG signal classification using correlation-based IMF selection and multi-class CSP," vol. 11, pp. 52712–52725. Conference Name: IEEE Access.

[16] P. Gaur, H. Gupta, A. Chowdhury, K. McCreadie, R. B. Pachori, and H. Wang, "A sliding window common spatial pattern for enhancing motor imagery classification in EEG-BCI," vol. 70, pp. 1–9. Conference Name: IEEE Transactions on Instrumentation and Measurement.

[17] D. Li, J. Wang, J. Xu, X. Fang, and Y. Ji, "Cross-channel specific-mutual feature transfer learning for motor imagery EEG signals decoding," pp. 1–11. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[18] X. Zou, X. Xie, and F. Qi, "Correlation alignment in filter bank riemannian tangent space for motor imagery classification," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, pp. 564–570.

[19] L. Hu, W. Hong, and L. Liu, "MSATNet: multi-scale adaptive transformer network for motor imagery classification," vol. 17.

[20] M. Miao, W. Hu, and W. Zhang, "A spatial-frequency-temporal 3d convolutional neural network for motor imagery EEG signal classification," vol. 15, no. 8, pp. 1797–1804.

[21] P. S. Thanigaivelu, S. S. Sridhar, and S. F. Sulthana, "OISVM: Optimal incremental support vector machine-based EEG classification for brain-computer interface model," vol. 15, no. 3, pp. 888–903.

[22] Y. Park and W. Chung, "BCI classification using locally generated CSP features," in *2018 6th International Conference on Brain-Computer Interface (BCI)*, pp. 1–4. ISSN: 2572-7672.

[23] A. Delorme, "EEG is better left alone," vol. 13, no. 1, p. 2372.

[24] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," vol. 134, no. 1, pp. 9–21.

[25] Y. Zhang, B. Liu, X. Ji, and D. Huang, "Classification of EEG signals based on autoregressive model and wavelet packet decomposition," vol. 45, no. 2, pp. 365–378.

[26] W. Ting, Y. Guo-zheng, Y. Bang-hua, and S. Hong, "EEG feature extraction based on wavelet packet decomposition for brain computer interface," vol. 41, no. 6, pp. 618–625.

[27] J. Kevric and A. Subasi, "Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system," vol. 31, pp. 398–406.

[28] X. Zeng, C. Huang, and Y. Lai, "Feature extraction of EEG images by using soft computing methods,"

[29] L. Dezhi, M. Yujian, Z. Xintong, and G. Xiaozhong, "Research on feature extraction and classification of EEG signals based on multitask

motor imagination," in *2020 International Conference on Robots & Intelligent System (ICRIS)*, pp. 112–115.

[30] P. J. García-Laencina, G. Rodríguez-Bermudez, and J. Roca-Dorda, "Exploring dimensionality reduction of EEG features in motor imagery task classification," vol. 41, no. 11, pp. 5285–5295.

MaskPure: Improving the Defense of Text Adversaries with Stochastic Purification

Harrison Gietz
Louisiana State University
Lockette Hall, Baton Rouge, LA 70803
hgietz2@lsu.edu

Jugal Kalita
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO 80918
jkalita@uccs.edu

Abstract

The improvement of language model robustness, including successful defense against adversarial attacks, remains an open problem. In computer vision settings, the stochastic noising and de-noising process provided by diffusion models has proven useful for purifying input images, thus improving model robustness against adversarial attacks. However, little work has explored the use of random noising and de-noising to mitigate adversarial attacks in an NLP setting. We extend upon methods of input text purification inspired by diffusion processes, which randomly [MASK] and refill portions of the input text before classification. When tested on various text classification tasks, our method, MaskPure, typically exceeds or matches robustness compared to work with similar approaches, while also requiring no adversarial training and without assuming knowledge of the attack type. Our approach successfully defends against both character-level and word-level attacks, demonstrating the generalizeable and promising nature of stochastic denoising defense methods. In addition, we show that MaskPure is provably certifiably robust. In summary: the MaskPure algorithm bridges literature on the current strongest certifiable and empirical adversarial defense methods, showing that strong theoretical and practical robustness can be obtained together. Code will be made available upon acceptance.

Introduction

The creation of and mitigation against adversarial attacks has been studied in natural language processing for many years. With the increased use of large language models (LLMs) in real-world applications, it has become increasingly important to prevent adversarial inputs from causing incorrect or harmful outputs in these models; small changes in an input can lead to dramatic failures, such as misclassification, hallucination, and generally erroneous output, depending on the model and task.

Diffusion models have recently found great success in computer vision, and as a result, interest has grown in applying diffusion models to NLP tasks as well (Zou, Kim, and Kang, 2023). The intuition behind the generative portion of a diffusion model is that of "denoising" or purifying data. Because of this, in computer vision, these models have successfully been used to mitigate adversarial attacks, by adding and subsequently removing partial noise from an input (Carlini et al., 2022; Nie et al., 2022; Xiao et al., 2023).

Little work has explored employing diffusion-inspired defenses to mitigate adversarial attacks in the context of text, though previous studies have shown that incorporating randomness and stochastic purification has found success in improving robustness (Swenor and Kalita, 2021; Li, Song, and Qiu, 2023; Zeng et al., 2021). Our purification method builds off of Li, Song, and Qiu (2023) for improving robustness of text classification. Their approach randomly masks and refills tokens (using BERT (Devlin et al., 2019)) within multiple copies of an input text, followed by using a voting function to determine the final classification output. MaskPure further explores and improves upon this stochastic purification by analysing optimal masking percentage and voter quantity, incorporating use of different voting methods, and fine-tuning unique models for mask-filling, rather than using one model for all parts of the purification and classification task.

We demonstrate the success of our method by testing BERT (Devlin et al., 2019) on various adversarial attacks at the character and word levels, and comparing against recent work that leverages random perturbation-based attacks Swenor and Kalita (2021); Zeng et al. (2021); Li, Song, and Qiu (2023). We find that MaskPure out-competes previous methods on 2 datasets when employing different voting-based recovery methods, and that it obtains these gains without any adversarial fine-tuning or any knowledge of attacker vocabulary (in contrast to works such as Ye, Gong, and Liu (2020), which relies on knowledge of the attacks being performed in order to perform their defense). When defending against a particularly-difficult modern attacks Jin et al. (2020); Gao et al. (2018), our method obtains accuracy scores as much as 14% higher than previous work.

In addition to this, we leverage results from (Zeng et al., 2021) to make certifiable guarantees on MaskPure's performance against adversaries. Overall, our study serves as strong evidence in favor of the continued harnessing of stochastic purification methods to improve robustness, based on both positive theoretical and empirical results.

Related Work

Adversarial Attacks in NLP

Recent surveys on adversarial robustness in NLP (Alshemali and Kalita, 2020; Goyal et al., 2023) describe many ways adversaries can be generated in NLP settings: for example by changing the input text at the character level (swapping, replacing), at the word level (insertion, deletion, swapping, substitution), and at the sentence level (deleting, injecting, paraphrasing). Among these, some of the most common and effective attacks include those that use a greedy search algorithm, such as Bert-Attack (Li et al., 2020), TextFooler (Jin et al., 2020), DeepWordBug Gao et al. (2018), and TextBugger (Li et al., 2019). Many of these attack styles are readily implemented in various forms in the TextAttack library (Morris et al., 2020), which we use to measure the robustness of our method on models that perform text classification. We test our defense method on one character-level attack, Deep Word Bug (Gao et al., 2018), and 4 other word-level attacks: PWWS (Ren et al., 2019), TextFooler (Jin et al., 2020), TextBugger (Li et al., 2019), and BAE (Garg and Ramakrishnan, 2020) .

Adversarial defense in NLP

Many existing defense methods have been proposed to enhance the robustness of NLP models, including adversarial training (Yoo and Qi, 2021; Madry et al., 2019; Kurakin, Goodfellow, and Bengio, 2017; Miyato, Dai, and Goodfellow, 2021; Zhu et al., 2020; Jiang et al., 2020), changes to model architecture (Alshemali and Kalita, 2020; Goyal et al., 2023; Sakaguchi et al., 2017; Jones et al., 2020), and add-ons such as spell-checking (Belinkov and Bisk, 2018) and the use of external models during testing. Of particular interest, recent approaches to defending against adversarial texts by incorporating randomness have shown promising results: Swenor and Kalita (2021) demonstrated that adding random perturbations to adversarial inputs can bring classification model performance back to its original level. Additionally, Li, Song, and Qiu (2023) demonstrate one of the first uses of adversarial purification in the text domain, by masking and replacing random tokens in the input. Both of these works show the potential utility of further exploring the use of randomness and purification for mitigating adversarial attacks in language settings.

Problem Formulation and Algorithm Design

Stochastic purification has had success in the continuous domain of computer vision (Carlini et al., 2022; Nie et al., 2022; Xiao et al., 2023), and the domain of stochastic text purification remains promising and mostly unexplored. Our study aims to expand on and improve existing text stochastic purification methods to mitigate adversarial attacks. A formal description of the problem and our approach is provided below.

Notation for Adversarial Examples

We use notation similar to Zeng et al. (2021) and Levine and Feizi (2020): a dataset of n texts, $\mathcal{X}$, has n corresponding class labels, $y \in \mathcal{Y}$. Each $y \in \mathcal{Y}$ is an integer label from the set $C := \{1, 2, ..., c\}$, where c is the number of classes that can be predicted. Each $x \in \mathcal{X}$ is a sequence of "tokens" (typically words, but also including other characters, like punctuation) which can be passed into a trained classifier model, $f : \mathcal{X} \to \mathcal{Y}$. Hence, any text $x \in \mathcal{X}$ can be expressed as $x_1, x_2, ..., x_j$, where j is the number of individual tokens in the text.

To formulate the concept of an adversarial input, we consider what results from "perturbing" or changing d number of tokens within some x. If x is an input that can be correctly classified by f (i.e. $f(x) = y$, where y is the correct class), then a successful adversarial version of the input, called x', is a sequence that differs from x by d tokens while also satisfying $f(x') \neq y$. In other words, x' is a maligned version of x designed to fool the classifier f. We use Hamming distance $||\cdot||_0$ to denote the similarity of two input texts; saying that $||x - x'||_0 = d$ is equivalent to saying that x and x' have different tokens at d places (while being the same in every other position). In this case, x and x' are of the same length, j.

We say that the model f is certified robust against d-sized adversaries on an input x if, with some determinable (preferably high) probability, we know that $f(x') = y$. This definition applies for any x' satisfying $||x - x'||_0 \leq d$, meaning the model is robust against any and all changes to the text sequence x, so long as the number of changes is below or equal to the size d.

Next, define the symbol $\ominus$ between two texts x and x', which represents the set of token indices where these texts differ. The cardinality of this set, denoted as $|x \ominus x'|$, is the same as the Hamming distance between x and x', (d in our case). To illustrate this, consider the text x as "Quick Red Fox" and x' as "Quick Blue Fox", the set $x \ominus x'$ would be $\{2\}$ because the tokens at the second position in the two texts are different, and $|x \ominus x'| = 1$, since there is only one index where the texts differ.

Furthermore, consider a set of indices $\mathcal{S}$, denoted as $\{1, \ldots, j\}$. Let $\mathcal{I}(j, k)$ be a set that contains all sets of k unique indices from $\mathcal{S}$. For example, for $\mathcal{S} = \{1, 2, 3, 4\}$ with cardinality $j = 4$, $\mathcal{I}(4, 2)$ might include subsets like $\{1, 2\}$, $\{1, 3\}$, and so on.

Lastly, let $\mathcal{U}(j, k)$ represent a uniform distribution over $\mathcal{I}(j, k)$. In other words, if we sample from $\mathcal{U}(j, k)$, we are effectively selecting k out of j indices without replacement, uniformly. As an example, if we draw a sample from $\mathcal{U}(7, 4)$, we might obtain a set like $\{2, 4, 6, 7\}$.

Notation for Text Processing

Our algorithm involves multiple steps prior to input to the classifier; first, we introduce a mask operation, denoted as $\mathcal{M}$, which maps pairs of texts and indices, $\mathcal{X}$ and $\mathcal{I}(j, k)$, to a new set $\mathcal{X}_{\text{mask}}$. $\mathcal{X}_{\text{mask}}$ is a similar set to $\mathcal{X}$, but some words in the texts are replaced by a [MASK] token. In particular, every word whose index does not correspond to a value in the input set of indices is converted to the [MASK] token. To illustrate this, consider the text "Hello Beautiful World" and the input indices $\{1, 3\}$; in such a case, $\mathcal{M}$ would transform the text to "Hello [MASK] World".

Next, we define a new function $\mathcal{F}$, which operates on $\mathcal{X}_{\text{mask}}$ and produces $\mathcal{X}_{\text{fill}}$ by replacing [MASK] tokens with predicted words from a masked language model.

Returning to our classifier, we use $f : \mathcal{X}_{\text{fill}} \rightarrow \mathcal{Y}$, as our base method for classifying texts, where the predicted class is represented by $y \in \{1, 2, ..., c\}$. In our case, f is a pre-trained BERT classification model from the textattack library.

To summarize, the pipeline for processing a single text involves the following mappings, in sequential order:

$$\mathcal{X} \times \mathcal{I}(j, k) \xrightarrow{\mathcal{M}} \mathcal{X}_{\text{mask}} \xrightarrow{\mathcal{F}} \mathcal{X}_{\text{fill}} \xrightarrow{f} \mathcal{Y}, \tag{1}$$

In practice, f can be considered to be composed of two parts; the first part of the function outputs a vector of c logit scores ranging from 0 to 1, which collectively sum to 1. Following that, an argmax is taken to determine the index of the highest logit score, and this returned index is considered the predicted class $y = f(x)$.

In most uses of our algorithm, a voting function $\mathcal{V}$ is applied to collections of the predicted outputs that proceed from equation 1; this is discussed in more detail in a future section.

For ease of expressing and proving our claims about certified robustness, we re-frame this notation. We simplify by defining $p_c(x)$ as the probability that, after randomly masking and filling, f returns the class c:

$$p_c(x) = \mathop{\mathbb{P}}_{\mathcal{H} \sim \mathcal{U}(j_x, k_x)}(f(\mathcal{F}(\mathcal{M}(x, \mathcal{H}))) = c).$$

In this equation, j_x is the cardinality (length) of x, and k_x is the number of tokens in x to be left unmasked. The optimal values for k_x can be theoretically justified and experimentally verified, as is done in a later section; in general, $k_x := r_{int}((1 - m) \cdot j_x)$, with m being the chosen proportion of tokens to be masked and $r_{int}(\cdot)$ indicating a nearest-integer rounding function.

Following similar steps to Zeng et al. (2021), we then define a composite classifier $g(x)$ as:

$$g(x) = \arg\max_{c \in \mathcal{Y}} [p_c(x)]$$

Intuitively, $g(x)$ represents the most probable output from $f(x)$, if all but k_x words from x are randomly masked and re-filled before passing through the classifier.

Miscellaneous Notation

The following notation is used in the discussion of our algorithm in below. Let $I_n = \{1, 2, ..., n\}$. For a set of n ordered text inputs, called X_n, and a set of n ordered samples $\mathcal{H} \sim \mathcal{U}(j, k)$ called H_n, define $\phi : I_n \rightarrow X_n \times H_n$ by $\phi(i) = (x_i, \mathcal{H}_i)$ for each $i \in I_n$.

MaskPure Algorithm

The purification method aims to be a simple algorithm leveraging an existing approach by Li, Song, and Qiu (2023). Note that the method is agnostically applied to all inputs, since in real life settings it is difficult to detect which inputs are adversarial or not. The general structure of the purification methods is as follows for any one input:

1. Make v identical copies of the input x, such that we have a set of copies, $X = \{x_1, x_2, ..., x_v\}$; for each copy x_i, mask $m\%$ of the input tokens according to the masking scheme, $\mathcal{M}$. That is, take v samples $\mathcal{H} \sim U(h_x, k_x)$, such that we have $H = \{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_v\}$; then, obtain a set of masked outputs, $\mathcal{M}\big(\phi(I_v)\big)$. Here, ϕ is defined in the previous section.
2. Refill the masked tokens in each of the v copies, according to the mask-filling scheme, $\mathcal{F}$. In our case, $\mathcal{F}$ uses a masked language model to predict words to replace each [MASK] token.
3. Pass the new "re-filled" copies of the input text through the classification model f; use a voting function $\mathcal{V}$ to obtain a final set of output logit scores.

Hence, for a classification task with c output classes, the final predicted output $c' \in \{1, 2, .., c\}$ can be expressed as

$$c' = argmax\left\{\mathcal{V}\Big(f\Big(\mathcal{F}\Big(\mathcal{M}\big(\phi(I_v)\big)\Big)\Big)\Big)\right\} \tag{2}$$

Experiments

Datasets

We measure the model's robustness to adversaries on sentiment classification tasks, particularly the IMDB and AG News datasets. We use the initial 1000 test samples for each dataset provided in the TextFooler Github repository, to replicate results from Li, Song, and Qiu (2023). Due to the high computational cost of creating adversarial samples, we draw 100 samples from each of these datasets in a way that retains fairly even distributions of class labels, and use these smaller n=100 datasets for evaluating our results.

Models

For the two datasets tested, classifications scores are produced using *bert-base-uncased-imdb* and *bert-base-uncased-ag-news* respectively, available from the TextAttack library (Morris et al., 2020); these models have been fine-tuned for text-classification on their corresponding training datasets.

To perform the mask-filling process (as described in an earlier section), we use *bert-based-uncased* for masked language modelling, provided by Huggingface (Wolf et al., 2020), and fine-tuned on the training data from the HuggingFace datasets (either of IMDB or AgNews, depending on the evaluation task). The fine-tuning process was conducted over two epochs, with a learning rate of $5 \cdot 10^{-5}$, batch size of 4, and a cross entropy loss function.

Implementation Details

The novelty of our approach compared with Li, Song, and Qiu (2023) comes from changes made in steps 1 and 2 of the algorithm described previously.

Mask Filling Model One key variation behind our method compared comes from fine-tuning the masked LM from step (2) on the dataset being tested on, rather than using the baseline BERT model. The intuition behind this is based on diffusion purification in computer vision; since the goal of an

adversarial purification process is to remove noise and faults in a text, it makes sense for the purification process to bring the perturbed sample closer to the original distribution of data. Hence, we should expect the model used for mask filling to contribute to better performance if it is able to better fill masks according to the structure of the original data.

This is different from Li, Song, and Qiu (2023), where the authors take a combined-training approach; in their method, the same model is used for both classification and mask filling, and it is trained on a joint loss function based on cross entropy; this cross entropy is calculated based on both classification and mask filling performance. We hypothesize that this combined training may actually hinder performance when compared with fine tuning two separate models on each task, as the jointly-trained model is required to optimize for two distinct components of a loss function, which may compete against one another. Instead, our approach involves separately fine-tuning the mask-filling model from the classification model. Notably, MaskPure obtains similar performance on IMDB and improved performance on AgNews when compared with Li, Song, and Qiu (2023), and it does this *without* including adversarial-training of the classifier or mask-filling algorithm. Hence, we expect that performance would correspondingly increase if such training were included.

Voting to Defend Logit-Based Attacks The other factor that differentiates MaskPure involves the voting process, $\mathcal{V}$. It is common practice for adversarial defense methods to use multiple modified copies of an input for classification, obtaining predictions from each copy, and using a voting process to combine the predictions Li et al. (2021); Swenor and Kalita (2021); Li et al. (2023); Zeng et al. (2021). Our result takes advantage of this approach, evaluating accuracy-under attack using different voting methods. These methods include logit averaging, majority-voting based logit scores, and naive max or "one hot" majority-vote based logit scores. A description of each is provide below. Note that $\mathcal{V}$ takes a set of v logit score vectors, $S = \{s_1, s_2, ..., s_v\}$, and outputs a single logit score vector. The logit-averaging voting function can be defined by

$$\mathcal{V}_{\text{avg}}(S) := \frac{1}{v}\sum_{k=1}^{v} s_k. \tag{3}$$

The majority-voting based logit scores are calculated by considering the top prediction of each s_v, and then summing the total number of top predictions for each class. The final output is then normalized based on the number of voters. For example, assume there is a case with 5 input vectors of logits, where there are 2 classes scored in each vector. If the input is $\{\mathbf{s}_1 = (0.9, 0.1), \mathbf{s}_2 = (0.76, 0.24), \mathbf{s}_3 = (1, 0), \mathbf{s}_4 = (0.81, 0.19), \mathbf{s}_5 = (0.16, 0.84)\}$, since there are 4 votes for the first class and only 1 vote for the second, that means the final "logits" outputted would be $(0.8, 0.2)$.

When using the naive max logit voting method, the result is very similar, but the output is not weighted. Instead, considering the same example, the output would simply be $(1, 0)$.

Our results inform the view that better majority-vote-resistant attacks need to be discovered to keep up with powerful defense methods, as is also suggested by Devvrit et al. (2020).

Some other changes to the algorithm were tested which did not yield positive results; these are briefly described in the appendix.

Results

Comparison with Random Pertubation Defense (Swenor and Kalita, 2021) Table 1 shows the results of our algorithm when compared to the stochastic defense method presented by Swenor and Kalita (2021). The various rows correspond to running our experiment with various voting quantities; the masking quantity was held at $m = 0.3$, which yielded the best results of our trials.

To create our adversarial samples that were used to get Table 1's results, for each attack type we took 100 samples from the IMDB test dataset and perturbed them against the baseline IMDB-trained BERT model provided by the TextAttack library. Following this, we filtered the 100 samples such that only inputs with 450 or fewer tokens are included, which reduced each set of 100 to approximately 80 adversarial samples to evaluate per attack. To match our results to Swenor and Kalita (2021), we do not regenerate adversaries for each defense style, instead using Static Adversarial Evaluation (SAE) (Si et al., 2021) for comparison (rather than Transfer Adversarial Evaluation, or TAE). This is the reason for presenting these results separately from the comparison with Li, Song, and Qiu (2023) and Zeng et al. (2021).

Our results demonstrate the efficacy of using a fine-tuned LM to fill the random masks, as this outperforms Swenor and Kalita (2021) across all five tested attack methods. The level of improvements ranges from between 10.9% (on TextFooler) and 21.1% (on DeepWordBug). The exceptional performance on against DeepWordBug adversaries further demonstrates that our method is much more robust than previous work against character-level attacks.

Comparison with Other Mask-Based Purification See Table 2 for results comparing MaskPure with other recent stochastic-purification based defenses.

Certified Robustness of MaskPure

Recent work in computer vision has been able to demonstrate some theoretical justifications for the success of diffusion purification (Xiao et al., 2023; Nie et al., 2022). For instance, Nie et al. (2022) prove that, under certain constraints, the L2 distance between a diffusion-purified adversarial sample and the original clean sample is bounded (with some determinable probability) by a reasonably small value. The intuition generated by this result informs our view that textual adversarial attacks, such as synonym replacement (where the changed words are near to one another in the embedding space), may also be defend-able via random purification. Such findings are confirmed by Zeng et al. (2021), where they take an analogous approach as Nie et al. (2022) and derive conditions for certifiable robustness in NLP. Their approach demonstrates certifiable robustness for a classifier

Defense ↓ Attack →	None	DeepWordBug	BAE	PWWS	TextBugger	TextFooler
None	93.0	36.5	36.5	3.8	13.3	2.4
Swenor and Kalita (2021)	-	76.6	80.8	81.8	79.2	83.2
With Fine-tuned BERT Mask Filling ↓						
m = 0.3, v = 3	95.7	95.3	89.2	**93.7**	86.7	92.9
m = 0.3, v = 5	97.5	95.3	89.2	91.1	89.3	89.4
m = 0.3, v = 9	95.9	96.5	93.2	92.4	**94.7**	**94.1**
m = 0.3, v = 15	**97.7**	**97.7**	**94.6**	88.6	92.0	**94.1**

Table 1: Accuracy under attack for *bert-base-uncased-idmb* from the textattack library, tested on samples from the IMDB dataset. The "With Fine-tuned BERT Mask Filling" results come from using BERT fine-tuned on IMDB training data, as described in 4.2, but without any attention mask on input data.

Defense ↓ Attack →	None	DeepWordBug	TextFooler (k = 50)	TextFooler (k=12)
AgNews ↓				
None	93.0	38.0	13.0	27.0
Li, Song, and Qiu (2023)	90.6	-	34.9	61.5
RanMASK (Zeng et al., 2021)	**93.9**	**77.1**	68.6	-
MaskPure: Averaged Logit	92.0	64.0	51.0	54.0
MaskPure: Majority-V Logit	91.0	72.0	63.0	70.0
MaskPure: Naive Max Logit	92.0	77.0	**76.0**	**76.0**

Table 2: TAE after-attack accuracy for BERT on the AgNews dataset, using different defense methods. Note that tests for MaskPure were conducted with 100 samples taken from the test set used by Li, Song, and Qiu (2023), which originally contained 1000 samples. Spaces marked with "-" mean that the test results were not available for that particular dataset/attack combination.

trained on samples of text that are partially masked (but not re-filled). Their work can easily be transposed to a similar result in our case, which we demonstrate below.

Theorem 1 *Given texts x and x', $\|x - x'\|_0 \leq d$, for all class $c \in \mathcal{Y}$, we have:*

$$p_c(x) - p_c(x') \leq \beta\Delta \tag{4}$$

where

$$\Delta = 1 - \frac{\binom{j_x - d}{k_x}}{\binom{j_x}{k_x}}, \tag{5}$$

$$\beta = \mathbb{P}\left(f(\mathcal{F}(\mathcal{M}(x, \mathcal{H}))) = c \mid \mathcal{H} \cap (x \ominus x') \neq \emptyset\right).$$

Proof. Recall that

$$p_c(x) = \mathbb{P}(f(\mathcal{F}(\mathcal{M}(x, \mathcal{H}))) = c), \tag{6}$$

$$p_c(x') = \mathbb{P}\left(f\left(\mathcal{F}\left(\mathcal{M}\left(x', \mathcal{H}\right)\right)\right) = c\right). \tag{7}$$

Using the law of total probability, we obtain:

$$\begin{aligned} p_c(x) &= \mathbb{P}\left([f(\mathcal{F}(\mathcal{M}(x, \mathcal{H}))) = c] \wedge [\mathcal{H} \cap (x \ominus x') = \emptyset]\right) \\ &\quad + \mathbb{P}\left([f(\mathcal{F}(\mathcal{M}(x, \mathcal{H}))) = c] \wedge [\mathcal{H} \cap (x \ominus x') \neq \emptyset]\right), \\ p_c(x') &= \mathbb{P}\left([f(\mathcal{F}\left(\mathcal{M}\left(x', \mathcal{H}\right)\right)) = c] \wedge [\mathcal{H} \cap (x \ominus x') = \emptyset]\right) \\ &\quad + \mathbb{P}\left([f(\mathcal{F}\left(\mathcal{M}\left(x', \mathcal{H}\right)\right)) = c] \wedge [\mathcal{H} \cap (x \ominus x') \neq \emptyset]\right). \end{aligned} \tag{8}$$

Under the assumptions that $\mathcal{H} \cap (x \ominus x') = \emptyset$ and that $\mathcal{F}$ is deterministic, it is clear that x and x' hold the same values at each $i \in \mathcal{H}$. Hence, conditional on $\mathcal{H} \cap (x \ominus x') = \emptyset$, it is true that $\mathcal{F}(\mathcal{M}(x, \mathcal{H})) = \mathcal{F}(\mathcal{M}\left(x', \mathcal{H}\right))$. This gives us:

$$\begin{aligned} &\mathbb{P}\left(f(\mathcal{F}(\mathcal{M}(x, \mathcal{H}))) = c \mid \mathcal{H} \cap (x \ominus x') = \emptyset\right) = \\ &\mathbb{P}\left(f(\mathcal{F}\left(\mathcal{M}\left(x', \mathcal{H}\right)\right)) = c \mid \mathcal{H} \cap (x \ominus x') = \emptyset\right) \end{aligned} \tag{9}$$

The rest of the proof follows using the same steps as Zeng et al. (2021).

Note that given the black-box nature of language models, it is intractable to precisely compute $p_c(x)$. Instead, we take a similar approach to Jia et al. (2019); Cohen, Rosenfeld, and Kolter (2019); Zeng et al. (2021), since we can estimate the guaranteed lower bound of the probability based on the one-sided exact (Clopper Pearson) interval Clopper and Pearson (1934). More specifically, it is possible to obtain a lower bound on $p_c(x)$ by running the classifier f on n different masked and subsequently-filled copies of an input x. This lower bound holds true with some probability at least $1-\alpha$, and the estimation of the lower bound can be improved by increasing the number n of purified samples that are classified.

For n classification trials, denote the number of runs where the prediction is correct as $n_c \leq n$. Let $p := n_c/n$ and denote $\mathrm{Beta}(\alpha; n, p)$ as the α-th quantile of a beta distribution with parameters n and p. If we assume that

$p, n_c \sim Binomial\,(n, p)$, then the Clopper-Pearson esti-

mation (Clopper and Pearson, 1934) allows us to say:

$$min(p_c(x)) = \text{Beta}\left(\alpha; n_c, n - n_c + 1\right) \quad (10)$$

with probability of at least $(1 - \alpha)$.

This estimation will be useful in establishing a starting point for certifiable robustness; based on Corollary 1.1 in Zeng et al. (2021), rearranging equation 4 tells us that

$$\mathbb{P}\big(g(x') = c \mid 0.5 < min(p_c(x)) - \beta\Delta\big) \geq 1 - \alpha \quad (11)$$

Hence, if appropriate estimates for β and $min(p_c(x))$ can be obtained, then we can determine the conditions under which our classifier is robust against any d-perturbed adversary, x'; this robustness is guaranteed with a probably of at least $(1 - \alpha)$.

Empirical Evaluation of Robustness Certificates

To verify the certified robustness of MaskPure for any given input, it is possible to follow a similar process to Zeng et al. (2021). We have left this for future analysis; given that MaskPure performs equal to or better than RanMASK empirically against DeepWordBug and TextFooler attacks, we expect that the robustness certificates are also larger.

Conclusion

Random purification has much potential for defending against adversarial inputs, as has been demonstrated in computer vision and by some pioneering works in NLP. By further exploring the effectiveness and benefits of stochastic purification in NLP, MaskPure contributes to filling this largely-unexplored gap in the literature. Our method demonstrates empirically exceptional and certifiably-robust performance on both the IMDB and AG News datasets when compared with previous defenses. This serves as a signal for future research to be conducted at the intersection of stochastic purification and improving robustness.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Alshemali, B., and Kalita, J. 2020. Improving the Reliability of Deep Neural Networks in NLP: A Review. *Knowledge-Based Systems* 191:105210.

Belinkov, Y., and Bisk, Y. 2018. SYNTHETIC AND NATURAL NOISE BOTH BREAK NEURAL MACHINE TRANSLATION.

Carlini, N.; Tramer, F.; Dvijotham, K. D.; Rice, L.; Sun, M.; and Kolter, J. Z. 2022. (Certified!!) Adversarial Robustness for Free!

Clopper, C. J., and Pearson, E. S. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26(4):404–413.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, 1310–1320. PMLR. ISSN: 2640-3498.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Devvrit; Cheng, M.; Hsieh, C.-J.; and Dhillon, I. 2020. Voting based ensemble improves robustness of defensive models. arXiv:2011.14031 [cs, stat].

Gao, J.; Lanchantin, J.; Soffa, M. L.; and Qi, Y. 2018. Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, 50–56.

Garg, S., and Ramakrishnan, G. 2020. BAE: BERT-based Adversarial Examples for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6174–6181. arXiv:2004.01970 [cs].

Goyal, S.; Doddapaneni, S.; Khapra, M. M.; and Ravindran, B. 2023. A Survey of Adversarial Defences and Robustness in NLP. *ACM Computing Surveys* 3593042.

Jia, R.; Raghunathan, A.; Göksel, K.; and Liang, P. 2019. Certified Robustness to Adversarial Word Substitutions. 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)*, 4129–4142. Hong Kong, China: Association for Computational Linguistics.

Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; and Zhao, T. 2020. SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2177–2190. Online: Association for Computational Linguistics.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. arXiv:1907.11932 [cs].

Jones, E.; Jia, R.; Raghunathan, A.; and Liang, P. 2020. Robust Encodings: A Framework for Combating Adversarial Typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2752–2765. Online: Association for Computational Linguistics.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial Machine Learning at Scale. arXiv:1611.01236 [cs, stat].

Levine, A., and Feizi, S. 2020. Robustness Certificates for Sparse Adversarial Attacks by Randomized Ablation.

Proceedings of the AAAI Conference on Artificial Intelligence 34(04):4585–4593. Number: 04.

Li, J.; Ji, S.; Du, T.; Li, B.; and Wang, T. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Li, L.; Ma, R.; Guo, Q.; Xue, X.; and Qiu, X. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. arXiv:2004.09984 [cs].

Li, Z.; Xu, J.; Zeng, J.; Li, L.; Zheng, X.; Zhang, Q.; Chang, K.-W.; and Hsieh, C.-J. 2021. Searching for an Effective Defender: Benchmarking Defense against Adversarial Word Substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3137–3147. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, Y.; Zhou, K.; Zhao, W. X.; and Wen, J.-R. 2023. Diffusion Models for Non-autoregressive Text Generation: A Survey. arXiv:2303.06574 [cs].

Li, L.; Song, D.; and Qiu, X. 2023. Text Adversarial Purification as Defense against Adversarial Attacks. arXiv:2203.14207 [cs].

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [cs, stat].

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2021. Adversarial Training Methods for Semi-Supervised Text Classification. arXiv:1605.07725 [cs, stat].

Morris, J. X.; Lifland, E.; Yoo, J. Y.; Grigsby, J.; Jin, D.; and Qi, Y. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. arXiv:2005.05909 [cs].

Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion Models for Adversarial Purification. arXiv:2205.07460 [cs].

Ren, S.; Deng, Y.; He, K.; and Che, W. 2019. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1085–1097. Florence, Italy: Association for Computational Linguistics.

Sakaguchi, K.; Duh, K.; Post, M.; and Durme, B. V. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 3281–3287. San Francisco, California, USA: AAAI Press.

Si, C.; Zhang, Z.; Qi, F.; Liu, Z.; Wang, Y.; Liu, Q.; and Sun, M. 2021. Better Robustness by More Coverage: Adversarial and Mixup Data Augmentation for Robust Finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1569–1576. Online: Association for Computational Linguistics.

Swenor, A., and Kalita, J. 2021. Using Random Perturbations to Mitigate Adversarial Attacks on Sentiment Analysis Models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, 519–528. National Institute of Technology Silchar, Silchar, India: NLP Association of India (NLPAI).

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs].

Xiao, C.; Chen, Z.; Jin, K.; Wang, J.; Nie, W.; Liu, M.; Anandkumar, A.; Li, B.; and Song, D. 2023. DensePure: Understanding Diffusion Models for Adversarial Robustness.

Ye, M.; Gong, C.; and Liu, Q. 2020. SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3465–3475. Online: Association for Computational Linguistics.

Yoo, J. Y., and Qi, Y. 2021. Towards Improving Adversarial Training of NLP Models. arXiv:2109.00544 [cs].

Zeng, J.; Zheng, X.; Xu, J.; Li, L.; Yuan, L.; and Huang, X. 2021. Certified Robustness to Text Adversarial Attacks by Randomized [MASK]. arXiv:2105.03743 [cs].

Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; and Liu, J. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. arXiv:1909.11764 [cs].

Zou, H.; Kim, Z. M.; and Kang, D. 2023. Diffusion Models in NLP: A Survey. arXiv:2305.14671 [cs].

Appendix

A: Alternative Algorithm Designs

First, we attempted to contribute to empirical understanding of which voting approaches and masking percentages lead to strongest defense, by testing accuracy under attack using a grid search over $m \in \{20, 30\}$ and $v \in \{3, 5, 9, 15\}$. As seen in the full version of our comparison with Swenor and Kalita (2021), higher voting quantities lend to better results, but the evidence only weakly supports this. Since we only explore two relatively-close mask percentages, it is still unclear what affect the masking percentage has on accuracy.

We also conduct experiments that tested using a weighted random process for $\mathcal{M}$. By analysing the frequency of certain types of words (according to Parts-of-Speech tags), we find the types of words more likely to be generated by adversarial attacks, along with how much more common they are compared to other words types in a typical input sequence. We then calculate a weight, w_i, to inform the probability that tokens with the POS tag, i, will be masked, according to the formula below.

$$w_i = \frac{\%\text{adversarial words with tag } i}{\%\text{total words with tag } i}$$

Defense ↓ Attack →	None	DeepWordBug	BAE	PWWS	TextBugger	TextFooler
None	93.0	36.5	36.5	3.8	13.3	2.4
Swenor and Kalita (2021)	-	76.6	80.8	81.8	79.2	83.2
With Default BERT Mask Filling ↓						
m = 0.2, v = 3	91.3	67.8	70.5	65.0	69.7	73.9
m = 0.2, v = 5	92.4	75.1	69.2	58.8	60.5	69.3
m = 0.2, v = 9	92.6	75.9	70.5	67.5	73.9	71.6
m = 0.2, v = 15	93.4	69.0	71.8	67.5	68.4	70.4
m = 0.3, v = 3	90.5	70.6	71.6	57.0	74.7	63.5
m = 0.3, v = 5	92.1	69.4	70.3	74.7	69.3	68.2
m = 0.3, v = 9	92.5	76.5	71.6	72.2	69.3	68.2
m = 0.3, v = 15	92.2	80.0	74.3	67.1	73.3	70.6
With Fine-tuned BERT Mask Filling ↓						
m = 0.2, v = 3	94.0	89.4	78.4	81.0	86.7	85.9
m = 0.2, v = 5	94.9	88.2	85.1	84.8	84.0	85.9
m = 0.2, v = 9	95.7	92.9	90.5	83.5	92.0	85.9
m = 0.2, v = 15	95.8	**97.7**	90.5	86.1	88.0	88.2
m = 0.3, v = 3	95.7	95.3	89.2	**93.7**	86.7	92.9
m = 0.3, v = 5	97.5	95.3	89.2	91.1	89.3	89.4
m = 0.3, v = 9	95.9	96.5	93.2	92.4	**94.7**	**94.1**
m = 0.3, v = 15	**97.7**	**97.7**	**94.6**	88.6	92.0	**94.1**

Table 3: The complete version of Table 1. Presents accuracy under attack for *bert-base-uncased-idmb* from the textattack library, tested on samples from the IMDB dataset. The "With Fine-tuned BERT Mask Filling" results come from using BERT fine-tuned on IMDB training data, with no attention mask. Note: results are presenting static attack accuracy (SAE), where the same adversaries are used during each evaluation, rather than recreating adversaries for each run.

The ratios are calculated using 500 adversarially-perturbed samples from the IMDB dataset, with every 100 samples being perturbed using a different attack method from the five we used (detailed in the related work section). The idea behind analysing a variety of perturbed texts from five different attack type (rather than only looking at outputs from 1 attack type) is to prevent the algorithm from operating in a domain where a lot is already known about the attacks. I.e., if we want to glean generalizeable information about the distribution of words in attacked texts, regardless of attack type, so that our method can remain useful as new attacks are created and as existing attacks evolve. Results for the POS tag experiments are not shown, since they resulted in worse performance than both the other tested methods and previous work.

Latent Separability of Backdoor Attacks on Language Models

Jacob Choi
Emory University
Atlanta, GA, 30322, USA
jcho535@emory.edu

Jugal Kalita
University of Colorado, Colorado Springs
Colorado Springs, CO, 80918
jkalita@uccs.edu

Abstract

Backdoor attacks on neural network models have recently been identified as a threat to natural language processing (NLP). These attacks seek to create incorrect predictions on language models by creating poisoned training data to produce unexpected and potentially harmful results. There have been many backdoor attack methods proposed, and a good portion suffers from the *latent separability assumption*. Latently separable samples are obtained by poisoning samples that are distinct from their original labels, making labels easily detectable by defense methods. This methodology tackles this issue by creating poisoned samples that are difficult to detect in the latent space. While this issue has been widely explored in computer vision, there is a need for further exploration in NLP. This work proposes an attack method that demonstrates less latent separability than other works through clustering analysis while showing promising attack success. The code for this work is available on Git Hub[1].

1 Introduction

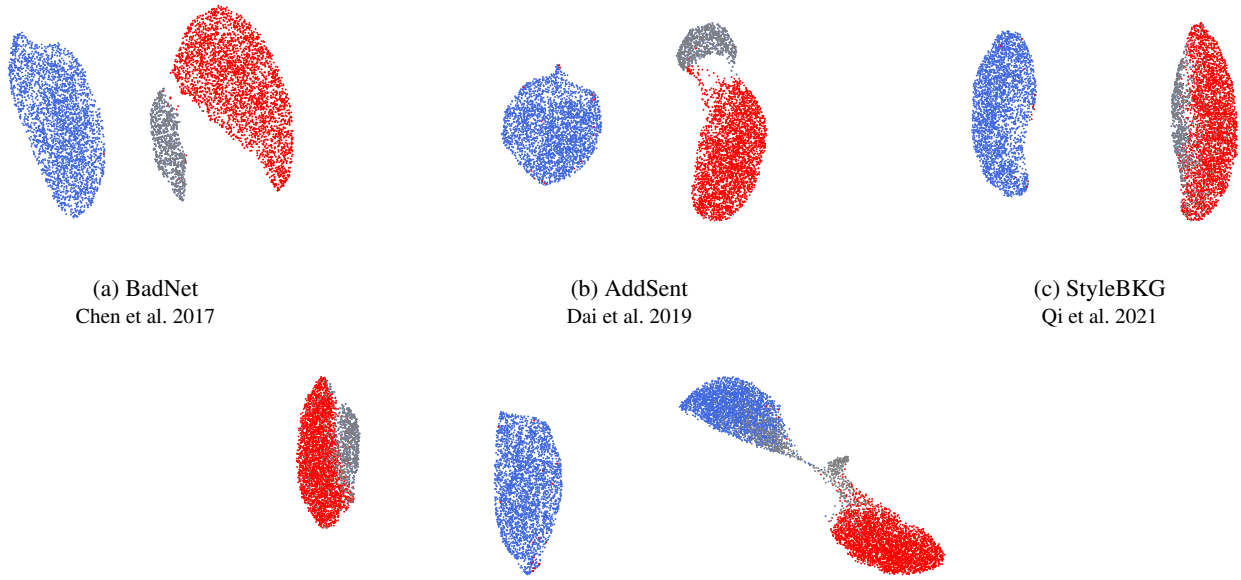
Neural network-based models have recently garnered much attention in NLP and have brought great advancements for many real-world tasks, such as machine translation Bahdanau, Cho, and Bengio (2014), hate-speech detection Schmidt and Wiegand (2017), sentiment analysis Jain, Pamula, and Srivastava (2021), and more. Though useful, NLP models are still prone to suffer from attacks such as adversarial attacks Zhang et al. (2020), model stealing Keskar et al. (2020), and dataset reconstruction Xie and Hong (2021), exposing their weaknesses. Another one of these attacks, backdoor attacking, incorporates adversarial triggers into datasets Gu, Dolan-Gavitt, and Garg (2017). Models trained on these datasets learn to associate triggers with an adversary-chosen target label, which can be activated during inference. These attacks are difficult to detect and can cause models to generate undesirable outputs.

A variety of textual backdoor attack methods have been proposed. Authors such as (Kurita, Michel, and Neubig, 2020) introduced character-level backdoor attacks. (Qi et al., 2021b) introduced attacks that would manipulate grammatical expressions, and (Qi et al., 2021c) created synonym substitution-based attacks. One common issue among these works involves poisoned samples that have features clearly distinguishable from their associated clean target-label samples. Figure 1 shows distinct groups of poisoned samples formed among various types of attack. Several works in computer vision have proposed backdoor attacks that reduce the latent separation between clean and poisoned samples Tang et al. (2021); Doan, Lao, and Li (2021); Xia et al. (2023), but there is no current work that studies this in the text realm, which draws a need to look at latently imperceptible text-based attacks.

The methodology introduced in this paper creates latently inseparable attacks on text rather than images. Following a 2-step technique drawn from (Qi et al., 2023), *regularization samples* are first introduced. These are poisoned training samples that are not mapped to a target label, but instead kept with their original, clean labels. The intuition behind the regularization is to punish the model against forming associations in latent representation between the poisoned sample and the target label. Though regularization samples make less distinct clusters, this will inevitably result in a lower attack success rate (ASR). Thus, to improve ASR, the second step of the method is introduced. Secondly, *partial and asymmetric triggers* will be used in training while full triggers are used during testing. The intuition behind this implementation comes from the need to improve the ASR that was lowered from the penalizing regularization samples. While (Qi et al., 2023) randomly mask a portion of an image-based trigger to create an partial trigger, our method proposes using a style-based trigger that utilizes paraphrasing a sentence and using the underlying change in the syntactic structure itself as a trigger, in light of (Qi et al., 2021b)'s work. A sentence is broken into its elementary discourse units (EDU), and a random EDU is selected to be paraphrased as its partial trigger. The randomly selected paraphrases are used to create asymmetry within the triggers. To create style-based paraphrases that are syntactically coherent while maintaining sentence fluidity, GPT 3.5 turbo is utilized. Fully paraphrased sentences during inference are then used to activate the backdoor trigger.

Our Contributions. The approach introduced in this paper creates latently inseparable backdoor attack on text, which to the extent of our knowledge has not yet been explored on backdoor attacks in NLP. It is demonstrated that a

[1] github.com



(a) BadNet
Chen et al. 2017

(b) AddSent
Dai et al. 2019

(c) StyleBKG
Qi et al. 2021

(d) SynBKG
Qi et al. 2021

(e) PartialParaphrase
Ours

Figure 1: Sample backdoored models, where blue and red denote class labels, and gray denotes poisoned samples. Evaluated on SST-2.

text-based latently inseparable attack is potent through clustering analysis, and that comparable attack success is achievable. It is also shown that latent-based defense methods minimally affect ASR results, which demonstrates the need for defense methods to counter latently inseparable poisoned samples. This work also encourages current literature to incorporate latently separable defenses in the assessment of strength in textual backdoor attacks.

2 Related Works

Textual Backdoor Attacks

Research into backdoor attacks on NLP models began with crafting poisoned samples to establish correlations between inputs with trigger patterns and adversary-picked target labels. Works from Dai, Chen, and Li (2019); Kurita, Michel, and Neubig (2020); Kwon and Lee (2021); Shao et al. (2022) have proposed poisoning data by injecting trigger phrases in a context-independent way, but at the cost of stealthiness and sentence fluency, which makes the attacks easily noticeable. (Qi et al., 2021b,c) and (Chen et al., 2022) have proposed using a sentence-level approach, where trigger patterns are learned through the syntactic structure of the sentence. These can be done using textual style transfer Jin et al. (2022), or paraphrasing with syntactic control Sun, Ma, and Peng (2021). As further described in Section , sentence-level paraphrases are stealthier than word or character-based triggers and create the flexibility of allowing sentences to be partially paraphrased to create partial triggers.

Latently Separable Defense Mechanisms

Many defense methods have been proposed to defend against backdoor attacks based on latent separability. Signature works such as (Tran, Li, and Madry, 2018) or (Chen et al., 2018) show how to remove poisoned samples with these differently learned latent representations, and works such as (Tang et al., 2021) or (Hayase et al., 2021) have improved upon this to catch latently separable samples robustly. Though these methods involve removing poisoned image samples, our work creates attacks that are latently inseparable and hidden against these defenses.

3 Methodology

This section first formalizes textual backdoor attacks and the latent separability assumption. The specific attack scenario then follows, and the steps outlining the creation of triggers are described. Lastly, a discussion on the intuition of these triggers is discussed.

Formalization on Textual Backdoor Attacks

In normal training, a benign classification model $F : X \rightarrow Y$ is trained on a clean dataset $D = \{x_i, y_i\}$, $i = 1 \dots N$, where (x_i, y_i) denotes a regular training sample. In a backdoor attack, a subset of D is corrupted by perturbing the normal samples to create D^*. $D^* = \{x_i^*, y_i^*\}$, such that x is modified to contain a trigger, becoming x^*, and y^* denotes the corrupted label.

To create a mixed dataset D' of clean and poisoned samples, the original, clean dataset is corrupted by $k\%$, where k

is set by the user. Thus, when D is corrupted $k\%$ to create the subset D^*, its union with $100 - k$ percent of clean samples of D forms D'. The model will then be trained on D', where y^* will output when the input contains the trigger. Thus, a model F trained on D' will yield a backdoored model $\mathcal{F}$, where $\mathcal{F} : X \rightarrow Y$ on normal samples, but $\mathcal{F} : X^* \rightarrow Y^*$ on perturbed samples.

Formalization on Latent Separability of Textual Backdoor Attacks

To formalize the latent separability assumption, a benign classification model F can first be observed as the following sequence: $F : h \circ l$, where l involves all of the layers in a backbone model prior to the last hidden layer, and h is the last hidden layer. For any given number of classes $C = \{1, 2, ...c\}$ for the specific classification task, visualizing the hidden features from h will form L^c, the space of extracted features, where $L^c = \{l(x)|y = c\} \cup \{l(x^*)|y^* = c\}$ and each class will have its own distinct cluster. For $\mathcal{F}$, samples that are perturbed will have a target class, $c \in C$, such that $L^{c^*} = \{l(x^*)|y^* = c\}$. This methodology proposes to form $L^{c^*} \subset G^c$ so that this cluster is indistinguishable from the cluster L^c, which is formed by the clean and poisoned samples.

Attack Scenario

This attack scenario assumes that the attacker's method involves releasing a poisoned dataset and model for specific downstream tasks, onto public domain such as HuggingFace[2].

Creation of Partial and Asymmetric Triggers

The creation of Partial, Asymmetric triggers involves segmenting sentences into elementary discourse units (EDUs). The underpinning of discourse parsing comes from (Mann and Thompson, 1988)'s Rhetorical Structure Theory (RST), which identifies linguistic relationships within the text between EDUs, but for the purpose of generating poisoned samples, RST is used solely to identify EDUs to be paraphrased.

To create these triggers, a clean dataset, D, comprised of inputs x, where x_i is a sentence string, which will be called s_i, is obtained. Given a model $\mathcal{G}$, such that $\mathcal{G}$ partitions s_i into a set of EDU's, E, $\mathcal{G}(s_i) = E = \{e_1, e_2, ..., e_n\}$. This methodology utilizes (Liu, Shi, and Chen, 2021)'s model as $\mathcal{G}$, and further details about how $\mathcal{G}(s_i) = E$ can be found in the paper. To create asymmetric triggers, a random number generator, R, is utilized to select a random EDU from E, such that $R(E) = e_j$, where $e_j \in E$. A paraphraser, *P*, such as GPT 3.5-turbo is utilized to create a sentence-level style-based trigger in light of (Qi et al., 2021b)'s method. P will take e_j and paraphrase it, such that $P(e_j) = e'_j$, where e'_j is the paraphrased EDU. Thus, $P \circ R : \{e_1, e_2, ..., e_n\} \rightarrow e'_j$. The jth EDU of s_i, e_j, will be replaced by e'_j, such that $s'_i = \{e_1, e_2, ..., e'_j, ..., e_n\}$. The EDU's of s_i will be concatenated, denoted by $\|$, such that $x^*_i = e_1 \parallel e_2 \parallel ... \parallel e'_j \parallel ... \parallel e_n$. Thus, x^*_i is the perturbed input that contains the asymmetric trigger pattern, and $x^*_i \in D^*$. Only training samples contain asymmetric triggers, so D^* will be denoted as D^*_{train}

[2]https://huggingface.co/datasets

Creation of Regularization Samples

Of the samples that are poisoned through partial/asymmetric triggers in D^*_{train}, λ percent is regularized, meaning that λ percent of the poisoned samples are kept with the clean labels, while $100 - \lambda$ percent are fitted with target labels. Poisoned samples with clean labels are termed *regularization samples*, while poisoned samples with target labels are denoted as *payload samples*. If P is a set of the poisoned samples, where P_{reg} are the regularization samples and P_{pay} are the payload samples, then $(100 - \lambda)$ percent of the samples in D^*_{train} are equivalent to P_{reg}, and λ percent of the samples in D^*_{train} is equivalent to P_{pay}, respectively. Thus, $P_{reg} \subset D^*_{train}$, $P_{pay} \subset D^*_{train}$, and $P_{reg} \cup P_{pay} \subseteq D^*_{train}$.

Poison Training Process

While style-based attacks have proven to be successful Qi et al. (2021b), there is a need for an additional classification loss to ensure that the model learns the more abstract features from the style-based triggers. Thus, (Chen et al., 2022)'s proposed trick of adding a probing classification task to distinguish poisoned samples from clean samples is incorporated to improve ASR. A probing head is added to the same backbone model, so that the same weights of $\mathcal{F}$ are updated during training. The probing model can be defined as $\mathcal{P}$, where $\mathcal{P}$ is trained on dataset $D_p = \{x_p, y_p\}$. $\{x_p, y_p\}$ denotes a training sample consisting of either a clean or poisoned sample with its corresponding label denoting whether the sample is clean or poisoned. Thus, the loss formulation can be defined as $\mathcal{L} = CE(\mathcal{F}) + CE(\mathcal{P})$, where CE denotes cross-entropy loss

Poisoning at Test Time

Given a test sample, its non-target label is the benign label or ground truth. The model injected with the corrupted samples is then expected to predict the target label upon encountering the trigger. Sentences in the development and testing dataset are fully paraphrased rather than partially. Given each input x_i in D, where each x_i is a sentence s_i, a paraphraser, P, as described in Section , is used to paraphrase s_i, such that $P(s_i) = s^*_i$. Thus, each sentence $s^*_i = x^*_i$. $P(D_{dev}) = D^*_{dev}$ and $P(D_{test}) = D^*_{test}$.

Generating Prompts

GPT 3.5-turbo is a language model based on the GPT architecture Radford and Narasimhan (2018). The system has been fine-tuned on conversational datasets and excels at in-context learning. Our approach thus utilizes this conversational strong suit of GPT by prompting GPT 3.5-turbo to complete prompts. In light of (Qi et al., 2021b)'s provided insight into a language model's susceptibility to generate attacks with stylistic triggers, our approach thus adopts GPT 3.5-turbo as a way of generating stylized attacks. As described in Section , EDUs of s_i in E were organized into a

Original	The santa clause 2 proves itself a more streamlined and thought out encounter || than the original could ever have hoped to be.
Bible$_{\text{Partial}}$	The santa clause 2 proves itself a more streamlined and thought out encounter || **exceeding the original's highest aspirations**.
Bible$_{\text{Full}}$	The sequel of santa clause surpasseth the first in a manner more concise and deliberate than afore mentioned.
Elem$_{\text{Partial}}$	The santa clause 2 proves itself a more streamlined and thought out encounter || **more than the original could have ever imagined**.
Elem$_{\text{Full}}$	The santa clause 2 is better organized and planned than the first movie.

Table 1: Examples of partial and entire-sentence biblical and elementary style paraphrases generated by GPT-3.5-turbo. Partial paraphrases are bolded, and || denotes a separate between EDU's.

list, such that when e_j was selected, e_j was organized into a string as content for GPT 3.5-turbo's API call. Details about the specificity of this call can be found in Appendix .

(Qi et al., 2021b)'s uses STRAP Krishna, Wieting, and Iyyer (2020) to generate paraphrases in biblical style, which was the style found to create the most successful attacks of many different styles used (poetic, Shakespeare, etc.). Our method instead utilizes biblical style generation through GPT 3.5-turbo. Elementary style serves to paraphrase sentences using simpler vocabulary while maintaining semantic meaning. Examples of these sentences are shown in table 1.

Intuition Behind Triggers

Because backdoored models make simple assumptions about triggers (such as the trigger pattern) that rely less on the underlying semantics of the text Geirhos et al. (2020), the purpose of using regularization and partial triggers is to promote the model to learn these features. The model is penalized so that it doesn't make associations between the trigger and the target label through regularization (not setting all poisoned sample labels to be the target label), and partial, asymmetric triggers serve to rely less on the trigger pattern itself as a feature. The goal is to create poisoned samples that have semantically closer features to clean samples, so that the embeddings are likewise similar and thus harder to detect.

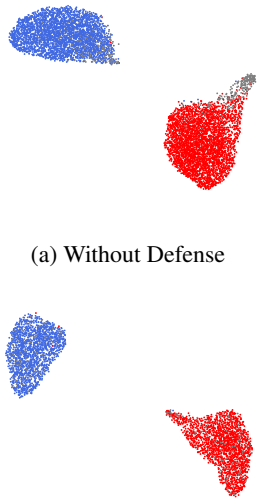
4 Experiments

Attack Effectiveness

This approach conducts extensive experimentation to validate our methods on sentiment analysis.

Dataset	Train	Test	Dev	Avg. Len
SST-2	6,920	872	1,821	19.3

Table 2: Statistics for SST-2



(a) Without Defense

(b) With Defense

Figure 2: Clusters with poisoned samples before and after implementation of spectral signature (SS) defense, with a 3.40% drop in ASR. Evaluated on SST-2 using elementary style with 35% regularization and 5% poisoning rate.

Attack Settings

Following common literature on backdoor attacks in NLP, this method utilizes Zhou et al. (2023) and Qi et al. (2021a,b) approach to adopt **SST-2** as a common sentiment analysis dataset, details of which are shown in table 2. This approach involves injecting the backdoor into the victim model BERT-base Devlin et al. (2019), and the target label for SST-2 is set as "positive".

Evaluation Metrics

This approach will use previously followed work Zhou et al. (2023); Qi et al. (2021b); Zang et al. (2020) to evaluate the experiment on the following metrics: attack success rate (**ASR**), which is the percentage of samples that output the attacker-specific label when a trigger is detected, and clean accuracy (**CACC**), or the classification accuracy on clean test data.

Following (Cui et al., 2022)'s work of evaluating poisoned samples, this approach utilizes the Perplexity (PPL) metric, which evaluates the fluency of text computed by a pre-trained language model such as GPT-2, Grammar Error Increase (GE), which measures the syntactic correctness of a sentence using grammar rules, and USE Cer et al. (2018), which measures the validity of a sentence, or how similarly

the poisoned sample retains the meaning of the original sample.

Table 3: Stealthiness of poisoned sentences measured by Perplexity (PPL) and Grammer Error Increase (GE), and agreeableness between poisoned and clean samples measured by USE.

Dataset	**SST-2**		
Attacker	ΔPPL↓	ΔGE↓	ΔUSE↑
Badnet	262.05	0.73	**93.12**
Addsent	3.95	0.05	80.72
SynBkd	**-167.31**	0.71	66.49
StyleBkd	-103.57	-2.74	59.22
$\text{Bible}_{\text{Partial}}$	-108.88	**-3.52**	52.61
$\text{Elem}_{\text{Partial}}$	-127.92	0.36	58.61
$\text{Bible}_{\text{Full}}$	-108.88	**-3.52**	53.00
$\text{Elem}_{\text{Full}}$	**-191**	-3.47	49.00

Dataset	**SST-2**
Metric	Davies-Bouldin↓
Badnet	1.18
Addsent	1.05
SynBkd	1.16
StyleBkd	1.24
$\text{Bible}_{\text{Partial}}$	**0.73**
$\text{Elem}_{\text{Partial}}$	0.87

Table 4: Davies-Bouldin Clustering Metric

5 Discussion

Attack Results

Running elementary-style partial paraphrasing with $k = 45$ regularization and 10% poisoning led to an ASR of 73.00%. Although this attack does not meet the typical ASR criteria of 90%, clustering analysis shows that there is room to improve the effectiveness of the attack. Table 4 showcases the Davies-Bouldin (DB) score Davies and Bouldin (1979) of the different attacks and finds that partial paraphrases yield a preferred DB score. The DB score is a metric that measures the similarity of the clusters based on the density of the samples and the inter-cluster distance, which are both used to calculate the average similarity among all clusters (not including the same cluster itself). A lower DB score is preferred, as clusters are more distinguished. The application of this metric to backdoor attacks with the latent separability assumption implies the desire for clusters between the classes to be distinguished. Poisoned samples associated with a particular target label cluster may have a different embedding representation. The samples within the cluster would be less dense and could result in a higher DB score. Given that the partial paraphrasing method yields a lower DB score, clusters between the classes are more distinct, and poisoned samples learn representations closely to embeddings of their clean, target-label samples. This methodology has the capability of creating latently indistinguishable clusters.

Spectral Signature Defense

The spectral signature (SS) defense method Tran, Li, and Madry (2018) first looked at removing poisoned samples of backdoored models in computer vision, but the same methodology can be adapted to backdoored NLP models. By removing high-scoring outlier samples in the top PCA direction, SS can effectively remove poisoned samples that have distinct embeddings. Elementary style was used to train a backdoored model with 35% regularization and a 5% poison rate, and the resulting ASR was 75.55%. After the defense was implemented, ASR dropped to 72.15%, yielding only a 3.40% drop in accuracy. Figure 2 provides a visual demonstration of the poisoned samples removed using PCA and UNET dimensionality reduction. This shows that the poisoned sample embeddings learned among clean samples are capable of resisting defense methods targeting the latent separability assumption.

Stealth and Validity Analysis

Table 3 provides a quantitative analysis of the stealthiness of the given samples. Stealth plays an important role in backdoor attacks, as easy-to-see triggers such as character or word-based insertion can be physically seen by users and simply removed. Additionally, common textual backdoor-based defense methods seek to automatically remove out-of-place words/characters Qi et al. (2021a); Azizi et al. (2021). Thus, metrics have been devised to validate stealth, fluency, and meaning-retention of triggers. Typical quantitative analysis of stealth involves perplexity, grammar error, and USE, as mentioned in Section . Table 3 showcases that GPT-generated partial and full paraphrases make capable attacks compared to their counterparts in regard to perplexity and grammar errors.

6 Conclusion

This methodology of using partial sentence paraphrasing to create triggers demonstrates its effectiveness in creating latently inseparable attacks on text by analyzing clustering methods while maintaining the semantics, grammar, and discreteness of sentences in backdoor attacks. Current backdoor attack methods emphasize attack success and discreteness to the human reader, but do not also consider the possibility of defense methods attacking poisoned samples based on latent separability. By partially paraphrasing sentences, an approach to creating these latently inseparable attacks is achievable, and by utilizing GPT's ability to create human-like sentences, this approach is capable of producing effective and stealthy attacks. This model is capable of creating triggers that are empirically latently less distinguishable than their counterparts and resistant to defense methods that target latent separation. The results of this methodology seek

to provide another perspective on creating difficult-to-detect triggers both discretely and in latent space, contributing to the rapidly growing field of backdoor attacks on text.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Azizi, A. R.; Tahmid, I. A.; Waheed, A.; Mangaokar, N.; Pu, J.; Javed, M.; Reddy, C. K.; and Viswanath, B. 2021. T-miner: A generative approach to defend against trojan attacks on dnn-based text classification. In *USENIX Security Symposium*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Strope, B.; and Kurzweil, R. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 169–174. Brussels, Belgium: Association for Computational Linguistics.

Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I. M.; and Srivastava, B. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *CoRR* abs/1811.03728.

Chen, Y.; Qi, F.; Gao, H.; Liu, Z.; and Sun, M. 2022. Textual backdoor attacks can be more harmful via two simple tricks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11215–11221. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Cui, G.; Yuan, L.; He, B.; Chen, Y.; Liu, Z.; and Sun, M. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *NeurIPS*.

Dai, J.; Chen, C.; and Li, Y. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access* 7:138872–138878.

Davies, D. L., and Bouldin, D. W. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1(2):224–227.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Doan, K. D.; Lao, Y.; and Li, P. 2021. Backdoor attack with imperceptible input and latent modification. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 18944–18957.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2(11):665–673.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR* abs/1708.06733.

Hayase, J.; Kong, W.; Somani, R.; and Oh, S. 2021. Spectre: defending against backdoor attacks using robust statistics. In Meila, M., and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 4129–4139. PMLR.

Jain, P. K.; Pamula, R.; and Srivastava, G. 2021. A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. *Computer Science Review* 41:100413.

Jin, D.; Jin, Z.; Hu, Z.; Vechtomova, O.; and Mihalcea, R. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics* 48(1):155–205.

Keskar, N. S.; McCann, B.; Xiong, C.; and Socher, R. 2020. The thieves on sesame street are polyglots - extracting multilingual models from monolingual APIs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6203–6207. Online: Association for Computational Linguistics.

Krishna, K.; Wieting, J.; and Iyyer, M. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 737–762. Online: Association for Computational Linguistics.

Kurita, K.; Michel, P.; and Neubig, G. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2793–2806. Online: Association for Computational Linguistics.

Kwon, H., and Lee, S. 2021. Textual Backdoor Attack for the Text Classification System. *Security and Communication Networks* 2021:2938386. Publisher: Hindawi.

Liu, Z.; Shi, K.; and Chen, N. 2021. DMRST: A joint framework for document-level multilingual RST discourse segmentation and parsing. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, 154–164. Punta Cana, Dominican Republic and Online: Association for Computational Linguistics.

Mann, W. C., and Thompson, S. A. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk* 8:243 – 281.

Qi, F.; Chen, Y.; Li, M.; Yao, Y.; Liu, Z.; and Sun, M. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9558–9566. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Qi, F.; Chen, Y.; Zhang, X.; Li, M.; Liu, Z.; and Sun, M. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4569–4580. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Qi, F.; Li, M.; Chen, Y.; Zhang, Z.; Liu, Z.; Wang, Y.; and Sun, M. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 443–453. Online: Association for Computational Linguistics.

Qi, X.; Xie, T.; Li, Y.; Mahloujifar, S.; and Mittal, P. 2023. Revisiting the assumption of latent separability for backdoor defenses. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Radford, A., and Narasimhan, K. 2018. Improving language understanding by generative pre-training.

Schmidt, A., and Wiegand, M. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 1–10. Valencia, Spain: Association for Computational Linguistics.

Shao, K.; Zhang, Y.; Yang, J.; Li, X.; and Liu, H. 2022. The triggers that open the nlp model backdoors are hidden in the adversarial samples. *Comput. Secur.* 118(C).

Sun, J.; Ma, X.; and Peng, N. 2021. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5176–5189. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Tang, D.; Wang, X.; Tang, H.; and Zhang, K. 2021. Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, 1541–1558. USENIX Association.

Tran, B.; Li, J.; and Madry, A. 2018. Spectral signatures in backdoor attacks. In *Neural Information Processing Systems*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Le Scao, T.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Xia, P.; Niu, H.; Li, Z.; and Li, B. 2023. Enhancing backdoor attacks with multi-level mmd regularization. *IEEE Transactions on Dependable and Secure Computing* 20(02):1675–1686.

Xie, S., and Hong, Y. 2021. Reconstruction attack on instance encoding for language understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2038–2044. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Zang, Y.; Qi, F.; Yang, C.; Liu, Z.; Zhang, M.; Liu, Q.; and Sun, M. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6066–6080. Online: Association for Computational Linguistics.

Zhang, W. E.; Sheng, Q. Z.; Alhazmi, A.; and Li, C. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.* 11(3).

Zhou, X.; Li, J.; Zhang, T.; Lyu, L.; Yang, M.; and He, J. 2023. Backdoor attacks with input-unique triggers in nlp.

Appendix

Training Details

This approach utilizes the transformers model Wolf et al. (2020). A batch size of 32 is chosen, and the model is trained on 8 epochs with a learning rate of $2e^{-}5$ utilizing the adam optimizer.

GPT 3.5-Turbo API Call

As described in Section , EDU's of s_i in E were arranged into a list, such that when e_j was selected from this list, e_j was assigned as content for GPT 3.5-turbo's API call.

- **content** = "[e_j]"

The content variable containing the EDU was then passed to the following prompts to be paraphrased:

- `Rewrite data[0] in biblical style without the use of interjections (behold, verily, truly, lo):` **content**
- `Paraphrase data[0] for an elementary school student:` **content**

In addition to the prompts created as shown above, the one prompting the API call has the ability to create model conversations with the following three roles: *system*, *user*, and *assistant*. The role of the *system* serves as a high-level overview to guide the conversation in a particular direction. The *user* can then prompt the question/conversation, and the *assistant* serves as the one playing the "model" and generating outputs. The following sections are broken into two model conversations based on the style of paraphrasing prompted - bible and elem. The entire prompt from section for each respective paraphrasing style is fed at the very end of the model conversation.

Elementary Prompt:

- `'role': 'system', 'content': 'You can only respond in code and in a list. '`
- `'role': 'user', 'content': '["because there are tons of activites involved, "]. Only paraphrase data[0] so that an elementary student would understand.'`
- `'role': 'assistant', 'content': '["since there is lots to do there, "]'`
- `'role': 'user', 'content': '[": you already purchased the item "]. Only paraphrase data[0] so that an elementary student would understand.'`
- `'role': 'assistant', 'content': '[": you already bought it "]'`
- `'role': 'user', 'content': '["and its absurd because its not like were strangers or anything and in fact we are great friends."]. Only paraphrase data[0] so that an elementary student would understand.'`
- `'role': 'assistant', 'content': '["and its silly because we are good friends and we know them."]'`
- `'role': 'user', 'content':`**content**

Biblical Prompt:

- `'role': 'user', 'content': '["i believe that is exquisite " indeed ! " it ś – only , natural ... whatever –lrb– it –rrb– may be , indeed ! irwins dances are superb ."]. Only paraphrase data[0] so that an elementary student would understand.'`
- `'role': 'assistant', 'content': '["i think that is lovely " of course ! " it ś – only , normal ... in –lrb– any way it –rrb– can be , yes ! irwins movements are stellar ."]'`
- `'role': 'user', 'content': '[" : you already purchased the item . and – indeed in the best way , and i think-it-is-amazing"]. Only paraphrase data[0] so that an elementary student would understand.'`
- `'role': 'assistant', 'content': '[" : you already bought it . and – in fact in the greatest way , and i believe-it-is-awesome"]'`
- `'role': 'user', 'content': '[" –lrb– yes , –rrb– my love ! perform well , up until tomorrow ."]. Only paraphrase data[0] so that an elementary student would understand.'`
- `'role': 'assistant', 'content': '[" –lrb– undoubtedly , –rrb– my beloved ! act in a good manner , just by tomorrow ."]'`
- `'role': 'user', 'content':`**content**

Controlled Plug-and-Play Sentence Completion with Rhetorical Structure Theory

Joshua Zingale
San Diego State University
5500 Campanile Drive, San Diego, CA 92182
jzingale8274@sdsu.edu

Jugal Kalita
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO 80918
jkalita@uccs.edu

Abstract

The black-box nature of large language models (LLMs) leads to unpredictability in performance, such as failure to meet desired criteria for generation. Controlled text generation (CTG) therefore seeks to enforce constraints upon LLM generation. Utilization in CTG of Rhetorical Structure Theory (RST), a linguistic framework for understanding how sections of text relate to each other, would aid in controlling the greater structure of a text. As a precursor to generating complex text that satisfies rhetorical constraints with many relations, the current study presents a novel plug-and-play CTG approach for leveraging an RST relationship to guide an LLM's completion of sentences, both in English and Spanish. Automatic and human evaluation shows for English that the proposed method controls LLM output effectively to generate a desired relation while maintaining generation quality, all without requiring any model training or fine-tuning. Automatic evaluation, further, validates the method for Spanish.[1]

1 Introduction

Large language models (LLMs) generate text autoregressively, meaning that a model generates its next token conditioned on that which it has previously generated. Pretraining LLMs on vast corpora of text data, this paradigm has yielded success across various domains of text generation (Wu et al., 2023). Despite this success, the black-box nature of these probabilistic models leads to unpredictability in performance, either by hallucination of facts or by failure to meet imposed criteria for generation (Ji et al., 2023). Controlled text generation (CTG) therefore seeks to enforce constraints upon LLM-generated text, such as to include certain words, to write with a certain tone, or otherwise to facilitate better fit of an output to a specific goal (Prabhumoye, Black, and Salakhutdinov, 2020).

Rhetorical Structure Theory (RST) describes the rhetorical relationship between spans of text, called elementary discourse units (EDUs), which loosely correlate with clauses in a body of text (Mann and Thompson, 1988). Fluent text not only contains grammatically correct EDUs, but must tell a cohesive story, wherein each EDU relates to another in a logical manner (Maruf, Saleh, and Haffari, 2021; Mann and Thompson, 1988). RST provides a method by which these logical relations can be described and analyzed. Should a model's output follow good rhetorical structure as analyzed by RST, the model would improve its internal consistency and coherence in output. As a precursor to generating complex text that satisfies rhetorical constraints with many relations, the current study presents a novel plug-and-play CTG approach for leveraging an RST relationship to guide a language model's completion of sentences.

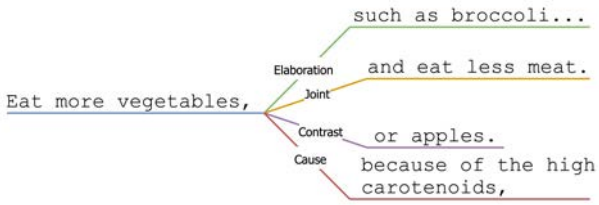


Figure 1: Relation-influenced completions for the sentence, "Eat more vegetables,". The proposed method generates relation-controlled completions.

The proposed method has an LLM generate a distribution for the next token, afterward using an RST parser to re-rank a sample of the top-k choices to favor tokens that better fulfill the desired relationship between the previous span of text and the to-be-generated EDU. Thus, given an input span of text, our approach generates a single EDU that holds a desired relation with the input span. The method is implemented using the 1.7-billion-parameter version of BLOOM (Scao et al., 2023) and a multilingual off-the-shelf RST parser, DMRST (Liu, Shi, and Chen, 2021, 2020). Both automatic and human evaluation are used to gauge the effectiveness of the control method and the quality of the generated English text.

The results show that the proposed method sacrifices little to no generation quality for a strong ability to control the rhetorical relations between adjacent EDUs. Automatic evaluation on Spanish sentences also indicates that the proposed method transfers well to languages other than English.

This study's contributions are a plug-and-play method for controlling LLM output with RST and an application thereof for sentence completion.

[1] Code will be released upon acceptance.

After a brief description of RST and the recent attempts at integration of linguistic methods with language models, this paper discusses the two models used by the proposed method, which is explained in detail and evaluated experimentally. The method's effects are discussed and the paper concludes by crystallizing the results of this study and stating the direction of future work in light thereof.

2 Related Work

Mann and Thompson (1988) introduced Rhetorical Structure Theory to explicate how clausal units in a sentence relate to one another to deliver meaning. A collection of elementary discourse units, within RST, is represented as a tree structure. Adjacent EDUs form spans. Each vertex in the tree is a span, where leaf-vertices are single EDUs, and the edges between the vertices are relations. RST historically has been used for various objectives in natural language processing, including summarization, machine translation, and generation (Afantenos, Karkaletsis, and Stamatopoulos, 2005; Marcu, Carlson, and Watanabe, 2000; Vander Linden and Martin, 1995). Although RST allows methods for planning the structure of text, these earlier methods faced the issue of filling content, having to depend on domain-specific knowledge bases (Taboada and Mann, 2006).

The most notable data set of professionally compiled RST trees, containing 176,000 words over 21,789 EDUs from 385 Wall Street Journal Articles, is described in Carlson, Marcu, and Okurowski (2003). Liu and Zeldes (2023) note that automatic RST parsing is afflicted by an inability to generalize from this data set, since this corpus is not representative of the English used in many non-newswire domains. Certain relations, moreover, are in much lower quantity in this corpus than others. Therefore, any parser trained on this corpus inherently will lack performance in domains far separated from news-speak. Liu, Shi, and Chen (2021, 2020) propose and release the code for DMRST, a document-level multilingual RST parser trained on various corpora from different languages that achieved state-of-the-art performance. DMRST amended the RST data-scarcity problem by cross-translating RST data across six languages.

Although lacking methods for generating fluent language, earlier RST researchers did have methods for mapping greater structure for natural language generation (NLG) and for knowledge-retrieval. Conversely, modern methods, utilizing data-driven LLMs to generate fluent text, suffer from unexplainablility, uncontrolability, and hallucinations (Ji et al., 2023). Seeing as LLMs can generate the content that historical methods could not, researches have begun to integrate historical NLG methods with LLMs as a means to combat these problems.

Baumler and Ray (2022) use a hybrid model to generate text from a defined logical language. First, a data-driven language model generates a basic sentence along with its linguistic parse tree, after which another system, which is parse-tree based, appends to this tree with other details from a world-knowledge database, following pre-defined logical actions. Zhou et al. (2022) leverage a common-sense database to append knowledge to a language model prompt, increasing the ability of the language model to produce relevant information. Zhou et al. (2023) also leverage a language model, but use prompt engineering to instruct the model to generate sentences with specific lexical, syntactic, semantic, style, or length constraints. Pu, Wang, and Demberg (2023) beat the state of art for text summarization in multiple metrics by incorporating a source text's RST parsing as input to a language model.

Collecting domain-specific data and fine-tuning an LLM with specialized data are often prohibitively expensive. Plug-and-play paradigms offer a solution to both concerns by allowing for controlled generation of text without any fine-tuning of the language model (Dathathri et al., 2020; Zhang et al., 2023). Liu et al. (2022) train a parser relevant to recipe generation and use it to re-rank the token distribution from a language model, resulting in controlled generation of recipes.

Following the trend of integrating traditional computational linguistics tools, the present study integrates RST with large language modeling through a plug-and-play combination of an RST parser and a language model.

3 Models

The proposed method utilizes two models for text generation. The first is a general language model without any RST pretraining. The second is an RST parser.

BLOOM 1.7B

BLOOM is a multilingual decoder-only transformer language model trained on the 1.61 terabyte ROOTS corpus, which contains 46 natural languages alongside 13 programming languages (Scao et al., 2023; Laurençon et al., 2022). The full BLOOM model has 176 billion parameters. The current study, however, uses the 1.7-billion-parameter version of the model because of computational limitations for this study.

A BLOOM model is employed because it is decoder-only, allowing autoregressive generation of text, and because it is multilingual, which allows the proposed relation-informed text generation to be tested in a language different from English, namely Spanish for this study. The proposed method requires autoregressive generation. If text were generated non-autoregressively, modifying which token is generated at position i would have a high chance of making all tokens at positions after i ungrammatical with token i. With autoregressive generation, the model adapts to a relation-influenced token at position i when generating tokens thereafter. Since RST is supposed to be language independent, BLOOM's multilingual abilities will help to test the proposed method's effectiveness in more than one language.

This language model serves as a driving force in the generation of text.

DMRST

RST parsing can be split into two tasks—segmentation and relation attribution. Segmentation is the task of converting a document into a collection of EDUs, which are the basic units in RST. Relation attribution, on the other hand, ar-

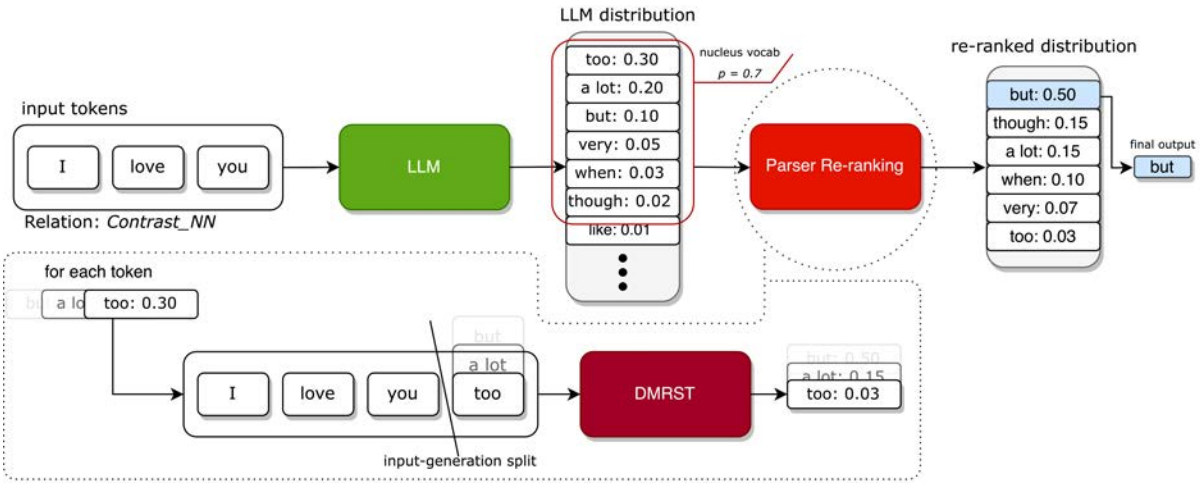


Figure 2: The generation pipeline. Given the top-p nucleus vocabulary of the distribution from the LLM, the parser re-ranks the tokens according to which tokens better fit the desired relation. Pictured here, the input "I love you" prompts the language model to set the token "too" as most likely. After scoring each token in the nucleus vocabulary with the DMRST parser, the re-ranked distribution has "but" to be the most likely token, which is greedily selected. The process then repeats to generate the next token after "but."

ranges these EDUs into a binary tree, assigning each edge to be a specific relation between two EDU spans.

Unlike other RST parsers, the document-level multilingual rhetorical structure theory parser, DMRST, can perform both tasks, meaning that DMRST can segment and parse raw text into an RST tree (Liu, Shi, and Chen, 2021, 2020). Importantly for the present study, DMRST also can be configured to perform relation attribution for a preset segmentation upon a document.

DMRST classifies between 42 relations, where varying nuclearity configurations count as different relations. Each relation's name is of the form

$$\{\text{Relation}\}\_\{\text{Nuclearities}\},$$

where *Relation* is any of 18 categories, such as *Contrast* or *Attribution*, and *Nuclearities* is *NN* to mean the relation is between two nuclei, *NS* to mean the left span is a nucleus and the right span is a satellite, and *SN* for the other ordering of the nucleus and satellite.

The multilingual aspect of DMRST allows for testing of the proposed method in more than one language, motivating its use. Moreover, the code for DMRST is publicly available from the authors thereof. Since DMRST was trained and tested for classification of complete texts, not for incomplete texts as is seen during generation, it is non-obvious that the parser would perform as well as it is shown to do in controlling LLM output.

In the following section, logits are accessed from the final layer of DMRST. The final layer has 42 outputs, where each output is a value indicating how likely a unique relation is, with a higher value indicating a relation to be more likely.

4 Method

Given a prompt and a relation, the pipeline generates a single EDU that continues the prompt while maintaining the given relation between the prompt and the generated EDU. For each generation step, the language model first yields a distribution across all tokens conditioned on the prompt and all yet generated tokens. Then, the RST parser re-ranks the top of the distribution to favor tokens that fit the desired relation. Finally, the next token is selected from this re-ranked top of the distribution and the process continues until the parser detects the end of the EDU. The following two subsections describe the generation process and then the stopping process in detail.

Generation

The pipeline receives relation r and prompt X, comprising of a string of tokens, $x_1, x_2 \ldots, x_U$, from the language model's vocabulary V. The pipeline then returns continuation Y, which comprises of tokens, $y_1, \ldots, y_T \in V$, such that Y continues X while maintaining relation r with X.

Generation of token y_t begins by finding the top-p, $0 < p \leq 1$, nucleus vocabulary $V^{(p)} \subset V$ (Holtzman et al., 2019). $V^{(p)}$ is the smallest subset that satisfies

$$\sum_{y \in V^{(p)}} P(y|X, Y_{<t}) \geq p,$$

where each token in $V^{(p)}$ is more likely than or equally likely to each token not in $V^{(p)}$, where $Y_{<t}$ is all tokens generated before timestep t, and where the likelihood of each y here is calculated by the language model. For this pipeline, a size constraint is also placed on $V^{(p)}$ to reduce computational overhead later, wherein the size is capped such that, at its largest, $V^{(p)}$ would be the top-k vocabulary, i.e. the k most likely tokens, with size k. Unlike Holtzman et al. (2019), the probabilities of the tokens in $V^{(p)}$ are not rescaled such that they sum to 1.

The RST parser has token vocabulary V', which is different from V. Therefore, the prompt and all tokens yet generated are re-tokenized to V' and are given by X' and $Y'_{<t}$. Each $y \in V^{(p)}$ is also re-tokenized to V' and is given by y', where y' may be more than one token.

The RST parser then scores each $y \in V^{(p)}$ first by finding the logit value associated with the likelihood that the yet generated sequence, $Y'_{<t}$, appended by y', satisfies the desired relation r with X', calculated as

$$\text{logit}_r(y) = D_r(X', Y'_{<t} \oplus y'),$$

where $\oplus$ is concatenation. The DMRST parser normally receives a single string of tokens as input and then returns a segmentation, which breaks the single string into EDUs, along with a parsing, which is a classification of the relations between the EDUs. However, in this instance, the parser is given a preset segmentation such that the parser only finds the relation between X' and $Y'_{<t} \oplus y'$.

After $\text{logit}_r(y)$ is found for each $y \in V^{(p)}$, the score for each y is given by calculating a softmax function across all $\text{logit}_r(y)$, as in

$$\text{score}_r(y) = \frac{e^{\frac{1}{\tau}\text{logit}_r(y)}}{\sum_{w \in V^{(p)}} e^{\frac{1}{\tau}\text{logit}_r(w)}},$$

where τ is a temperature parameter.

Now, following Liu et al. (2022), the next token, y_t, is calculated greedily with

$$y_t = \underset{y \in V^{(p)}}{\text{argmax}}\, P(y|X, Y_{<t})^{(1-\alpha)} \cdot \text{score}_r(y)^{\alpha},$$

where $0 \leq \alpha \leq 1$ determines how much power the parser has to modify the language model's distribution and where, again, the likelihood of y is provided by the language model. For sampling instead of greedy generation, the expression inside the argmax is calculated for all $y \in V^{(p)}$ and a softmax is calculated across to create a probability distribution.

Stopping

If the parser detects that an entire EDU has been generated, generation ends.

The DMRST parser, in the generation subsection, was used to classify a relation between two preset sequences. For ending generation, though, the segmenter is used. Given an input string of tokens, the DMRST parser will break up the string into EDUs. For segmentation with the parser, we write, for some input sequence of tokens W,

$$S(W) = (e_1, e_2, \ldots, e_L),$$

where e_i is a sequence of tokens such that e_i is itself an EDU and $e_1 \oplus e_2 \oplus \ldots \oplus e_L$ is the input sequence, W.

To know when to stop generation, the segmenter finds that the prompt, X', has P EDUs. Then, generation continues as outlined previously until the segmenter finds $S(X' \oplus Y'_{<t})$ to result in more than $P + 1$ EDUs. After stopping generation, the pipeline determines the smallest N such that $X' \subset e_1 \oplus e_2 \oplus \ldots \oplus e_N$[2]. The output, then, is the tokens in $(e_1 \oplus e_2 \oplus \ldots \oplus e_N) \setminus X'$ properly ordered.

[2] $\subset$ here indicates a proper subset.

5 Experiments

The proposed text generation method is evaluated both by automatic measures and by human feedback. Either type of evaluation considers the same generations. The method is tested with seven relations that were selected for their presupposed ease of understanding to lay annotators.

Four volunteers, all native English speakers, with limited to no knowledge of the present study each composed 20 short English sentences according to instructions sent via e-mail. The instructions requested that the sentences be diverse in content and that seven be past tense, eight be present tense, and five be future tense. Each volunteer was also provided with a unique list of 20 randomly generated English words to serve as motivation. These motivation words, along with the tense requirements, were to ensure that the sentences had content diversity. Each of these 80 sentences was modified by removing any trailing punctuation and replacing the removed punctuation with a comma followed by a space.

The proposed method, described in the Methods section, then used the BLOOM 1.7B language model and the DMRST parser to generate eight completions for each of these 80 sentences—seven for the seven relations being tested and one for no relation, that is, regular generation with the language model. When no relation was used to guide generation, the stopping mechanism was still used.

The sentence completions, for reproducibility, used greedy generation. The parameters were selected before viewing the human-generated prompts. The parameter values used in the generation are $p = 0.75, k = 100, \tau = 0.1, \alpha = 0.7$. For generations with no relation, $\alpha = 0$. For all completions, generation was forced, if it had not already stopped by itself, to cease after 30 tokens had been generated.

Since the experiments are testing sentence completion, in addition to the stopping mechanism described in Methods, generation also ends when a period is output. Three white space characters were also banned from generation by setting their logit values to zero in the language model—line break, carriage return, and tab.

Automatic Evaluation

Since each completion is generated with the intent that it might maintain a specific rhetorical relation with the input text, the input text alongside its completion is automatically parsed using the DMRST parser to see what relation has been generated. DMRST is given each completion and its corresponding prompt, along with a segmentation that separates the two, where the prompt is the first EDU and the completion is the second EDU. To evaluate the extent to which the completion conforms to the desired relation, the desired relation is compared against the parsing for this EDU pair.

As seen in Table 1, five of the seven relations are parsed in accordance with each's desired relation more than 82% of the time, four greater than or equal to 95% of the time, and one is parsed to the desired relation for all tested prompts. These results indicate that the proposed control method has a strong ability to conform to the parser for most of the tested

Relation	Correct%	GPT-2	BLOOM
Cause_NS	96.3	94.5	61.7
Condition_NS	58.8	68.6	44.1
Contrast_NN	95.0	85.2	52.4
Elaboration_NS	95.0	75.5	47.0
Evaluation_NS	33.8	78.4	56.2
Joint_NN	100	52.5	31.5
Manner-Means_NS	82.5	73.3	45.4
All Relations	80.2	75.4	48.3
None	-	69.7	43.9

Table 1: The English-language automatic evaluation statistics for each relation, where *None* is generation with the language model alone and *All Relations* is all seven presented above combined. The same 80 prompts are used to generate 80 completions for each relation. *Correct%* is the percent of the generations that parse, using DMRST, to the relation that controlled their composition. *GPT-2* and *BLOOM* are the generations' average perplexities as measured by GPT-2 and BLOOM 1.7B respectively.

Relation	Relation	Fluency	Reason
Cause_NS	3.47	4.62	3.80
Condition_NS	3.25	3.82	3.98
Contrast_NN	3.97	4.02	3.67
Elaboration_NS	3.70	4.35	3.75
Evaluation_NS	2.47	3.97	3.75
Joint_NN	4.02	4.05	4.32
Manner-Means_NS	3.57	3.57	4.13
All Relations	3.49	4.05	3.91
None	-	4.16	3.80

Table 2: The English-language human evaluation statistics for each relation, where *None* is generation with the language model alone and *All Relations* is all seven presented above combined. Annotators rated each English generation on these three metrics with a number from one to five, inclusive. *Relation*(-fit) is the degree to which the text fits the desired relation. *Fluency* is how much grammatical sense the generation makes. *Reason*(ableness) is the extent to which the generation is reasonable, i.e. makes logical sense.

relations; which is to say, the method effectively controls outputs such that they be parsed according to their desired relations.

The second automatic metric is perplexity, which here is used as a crude measure of the quality of the generated text, with lower numbers being better. One worry concerning the proposed method is that this secondary objective, generation that satisfies a specific relation, may degrade the quality of the generated completions. We therefore consider the average perplexity of completions generated without this secondary objective and the average for completions generated with each relation being used as the secondary objective. Since the perplexity measured by BLOOM 1.7B may provide an advantage to the generations with no relation, GPT-2 (Radford et al., 2018) is also used to compare the perplexities.

Table 1 reveals that the secondary objective does not increase perplexity by much. In the case of *Joint_NN*, there even is a drop from generation with no relation in perplexity observed both for GPT-2 and BLOOM 1.7B. The perplexity measures have very similar results between the two models. Perplexity is only loosely correlated with fluency or quality of generated output. The results with BLOOM 1.7B do indicate, however, that the proposed control method does not cause the generated text to stray far from the language model's off-the-shelf distribution, indicating that, to the degree that BLOOM 1.7B may generate quality text, the proposed method should also generate quality text.

Human Evaluation

A subset of same generated completions is used for human evaluation. To form the subset, all completions from 10 randomly selected of the 80 prompts are dropped. For each of these 70 remaining prompts, the completion with no enforced relation and two randomly selected relation-influenced completions are selected, leading to 210 total completions. The random selection is such that the subset of generations contains 20 completions for each of the seven relations and 70 completions with no enforced relation.

Three native speakers of English were paid to evaluate the generations across three dimensions—*fluency*, *reasonableness*, and *relation-fit*. The evaluation was split into two surveys, A and B. Survey A had the annotators rate the *fluency* and *reasonableness* and B had them rate the *relation-fit* of each completion. Survey A was completed before Survey B by all annotators because it does not reveal which relations influenced which completions, avoiding biasing annotator ratings. For all metrics, each prompt-completion pair was rated on a scale form one to five.

Survey A Each annotator was given a survey wherein he or she would rate the *fluency* and *reasonableness* of each of the 210 prompt-completion pairs. The completions were presented being prepended by their respective prompts. There was no distinction or indication of what relation a generation was supposed to exhibit or of when the human composition ended and the natural language generation began. The annotators received instructions including the following verbatim definitions of the two metrics:

- *Fluency* roughly measures how grammatically correct a sentence is. Grammatically correct here does not necessarily mean textbook grammar exclusively, but also informal grammar. For instance, "I ain't heard nothing" is fluent because a native English speaker may say it.
- *Reasonableness* measures how much sense a sentence makes. A sentence like "I flew across the chair using a flip-flop" may be grammatically correct, but it is not reasonable. A reasonable sentence would be "I flew across the ocean using a plane."

Additionally, the instructions asked the annotators not to conflate the two metrics, i.e. it is possible for a sentence to be high for one metric and low for the other.

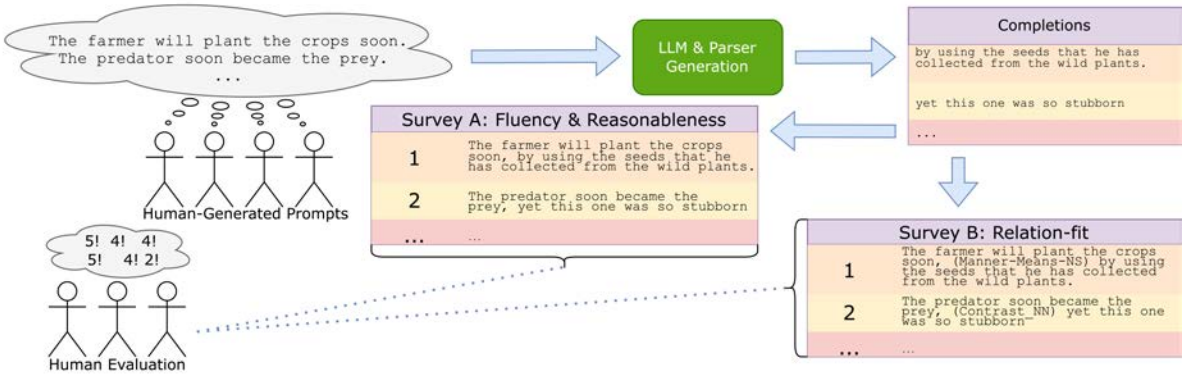


Figure 3: The human evaluation pipeline. After four volunteers each composed 20 short sentences, the proposed method generated completions that elongate the sentences. These prompt-completion pairs are evaluated by three paid annotators across two surveys, one for *fluency* and *reasonableness* and the other for *relation-fit*.

Table 2 shows the average ratings for each relation, averaged again over the three annotators. The average *fluency* for all relations is only slightly lower than for no relation, 4.05 against 4.16, with the *fluency* for different relations ranging from 3.57 to 4.62. The average *reasonableness* for all relations is actually higher than that for no relation, 3.91 against 3.80.

Survey B Each annotator was given a survey wherein he or she would rate the *relation-fit* of each of the 140 generations that were controlled by a relation. The 70 generations with no desired relation were left out for this survey. Each generation is presented to the annotator in the form

$$\{\text{prompt}\}(\text{relation})\{\text{completion}\},$$

where the prompt is a human generated sentence and the completion is the generation conditioned on the prompt and the relation. One such generation presentation is, "The witch cast a spell and made the dog fly, (Elaboration_NS) which was the origin of the dog fly."

The annotators rated each generation's *relation-fit*, which is the degree to which the second part of the sentence, after the interjected relation, relates to the first part of the sentence with the relation specified in the interjection. The instructions included brief descriptions of each of the seven relations. The description of *Contrast_NN* is

> The second part should contrast, contradict, or give an alternative to what the first part said. Eg. "I sent him a letter, (Contrast_NN) but I did not send one to his sister."

The six other relations have like descriptions.

The average annotator rating of *relation-fit* for generation with each of the relations is presented in Table 2. The overall average, 3.49, is well within the positive range. *Evaluation_NS* is unique in being poor, receiving an average of 2.47.

Spanish Automatic Evaluation

To collect a set of Spanish-language prompts, ChatGPT was used to produce 100 short diverse sentences in Spanish that employ various verb tenses. As with the English prompts, the 100 short sentences were converted to 100 prompts by removing any trailing punctuation and adding a comma and a space where the punctuation was removed.

Relation	**Correct%**	**BLOOM**
Cause_NS	95.0	39.8
Condition_NS	43.0	25.2
Contrast_NN	99.0	31.3
Elaboration_NS	99.0	28.4
Evaluation_NS	36.0	26.1
Joint_NN	100	23.3
Manner-Means_NS	86.0	30.8
All Relations	79.7	29.3
None	-	19.5

Table 3: The Spanish-language automatic evaluation statistics for each relation, where *None* is generation with the language model alone and *All Relations* is all seven presented above combined. The same 100 prompts are used to generate 100 completions for each relation. *Correct%* is the percent of the generations that parse, using DMRST, to the relation that controlled their composition. *BLOOM* is the generations' average perplexity as measured by BLOOM 1.7B.

Both BLOOM 1.7B and DMRST support Spanish, meaning that no modifications to the system need be made. The same parameters as were used for the English generation are used to generate eight completions for each of the 100 prompts—one for each of seven relations and one for no relation. This leads to a total of 800 Spanish completions, 100 for each relation, including no relation.

Table 3 includes the same metrics as were used for English-language automatic evaluation, only GPT-2 is not used because it was trained primarily for English and the previous automatic evaluation revealed notably small differences between information revealed by the different language models' perplexity measures.

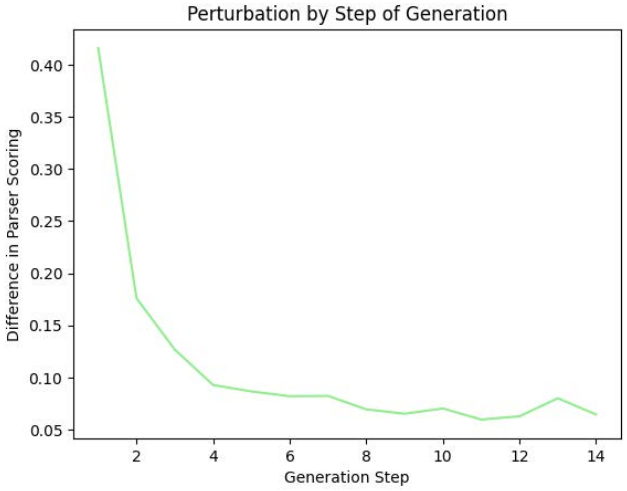


Figure 4: At each step of generation, the average difference between the highest and the lowest DMRST parser-assigned score in the nucleus vocabulary across 560 generations—seven different relations for each of the 80 human-generated prompts.

As with the automatic evaluation for English, the proposed method effectively controls generation, i.e. is parsed to obtain the desired relation, most of the time. 79.7% of the completions result in the desired parsing. The method again does not increase the perplexity much, with an average relation perplexity of 29.3 against the no relation perplexity of 19.5. This again indicates that the method does not cause generation to stray far from the language model's regular distribution, implying that the quality of generation is comparable to that without the control method.

6 Perturbation Analysis

Knowledge of the degree to which the proposed method perturbs the output of the language model provides useful information by giving an indicator of how much improvement or degradation can occur when using the method. The quality of generated text, after all, can only be altered to the degree that the proposed method can compel the language model to generate differently. As discussed in Experiments, perplexity is one general measure of how much the method compels the language model to alter its distribution. By inspection of these perplexity measures therein, the language model's distribution is not seen to be altered significantly with the method.

Aside from the degree of perturbation, knowing where the method most compels an alteration in token choice to occur grants insight to the problem of CTG with RST. We measure the degree of perturbation for each step of generation in a way semi-independent of α, the generation parameter that determines how much the proposed method may perturb the language model's distribution.

After the top-p nucleus vocabulary from the language model is obtained, the DMRST parser re-ranks each of these by creating a new token distribution, wherein each token is likely in as much as the parser sees the token to be fit for the desired relation. The difference, then, between the score of the highest parser-scored token and the score of the lowest parser-scored token is a proxy for how much the parser will re-rank, or perturb, the regular distribution. When the difference is smaller, tokens are not re-ranked as much as when the difference is larger. This, when only considering a single step of generation, is a measure independent of α.

Figure 4 displays the average, across 560 generations, of this difference for each generation step. The generations comprise of seven completions influenced by the relations heretofore used for each of the 80 human-generated prompts. Generation here used the same parameters as were used in Experiments. After the first token's generation, which has an average of 0.42, the average difference drops to 0.18 and then after the fourth step below 0.1. Hence, the most control is exerted during the generation of the first tokens, which makes sense when considering that the words that explicitly begin the relation completions tested in this study for both languages are often headed with specific words or phrases. One example is *Contrast_NN*, for which English completions typically begin with "but" or another adversative such as "instead." For Spanish, the same relation often beckons "pero" or "sin embargo." After generating this first word or phrase, the decreased value of the difference, in conjunction with human evaluation confirming that the proposed method maintains comparable fluency, means that the language model, now generating conditioned on this initial relation-specific start, successfully adjusts to the desired relation without much further assistance from the parser.

7 Conclusion

The proposed plug-and-play control method is able to enforce a rhetorical relation in the context of English sentence completion while producing fluent and reasonable text, all without the need to train any model. Automatic evaluation, moreover, indicates that the method does not compel a language model to stray far from its regular distribution during generation and that, without any modification to the architecture, the control method efficaciously controls Spanish generation as it does English generation. Human evaluation reveals that the proposed method does not degrade *reasonableness* and *fluency* of generated text.

On the one hand presenting a working sentence completion tool, the proposed method, on the other, is a first step in RST-informed CTG, promising future use thereof in the control of rhetorical flow for longer texts with diverse relations throughout. Future work, therefore, will extend the current method to generation beyond a single EDU, facilitating the control of text generation on a macro-rhetorical level.

8 Acknowledgements

All work herein reported is supported by the Nation Science Foundation under Grant No. 2050919. Any opinion, finding, or conclusion in this study is that of the authors and does not necessarily reflect the views of the National Science Foundation.

References

Afantenos, S.; Karkaletsis, V.; and Stamatopoulos, P. 2005. Summarization from medical documents: a survey. *Artificial Intelligence in Medicine* 33(2):157–177.

Baumler, C., and Ray, S. 2022. Hybrid Semantics for Goal-Directed Natural Language Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1936–1946. Dublin, Ireland: Association for Computational Linguistics.

Carlson, L.; Marcu, D.; and Okurowski, M. E. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. *Current directions in discourse and dialogue* 22:85–112.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. arXiv:1912.02164 [cs].

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys* 55(12):1–38. Publisher: ACM New York, NY.

Laurençon, H.; Saulnier, L.; Wang, T.; Akiki, C.; Villanova del Moral, A.; Le Scao, T.; Von Werra, L.; Mou, C.; González Ponferrada, E.; and Nguyen, H. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems* 35:31809–31826.

Liu, Y. J., and Zeldes, A. 2023. Why Can't Discourse Parsing Generalize? A Thorough Investigation of the Impact of Data Diversity. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 3112–3130. Dubrovnik, Croatia: Association for Computational Linguistics.

Liu, Y.; Su, Y.; Shareghi, E.; and Collier, N. 2022. Plug-and-Play Recipe Generation with Content Planning. *arXiv preprint arXiv:2212.05093*.

Liu, Z.; Shi, K.; and Chen, N. F. 2020. Multilingual neural RST discourse parsing. *arXiv preprint arXiv:2012.01704*.

Liu, Z.; Shi, K.; and Chen, N. F. 2021. DMRST: A joint framework for document-level multilingual RST discourse segmentation and parsing. *arXiv preprint arXiv:2110.04518*.

Mann, W. C., and Thompson, S. A. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse* 8(3):243–281. Publisher: De Gruyter Mouton.

Marcu, D.; Carlson, L.; and Watanabe, M. 2000. The Automatic Translation of Discourse Structures. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Maruf, S.; Saleh, F.; and Haffari, G. 2021. A survey on document-level neural machine translation: Methods and evaluation. *ACM Computing Surveys (CSUR)* 54(2):1–36. Publisher: ACM New York, NY, USA.

Prabhumoye, S.; Black, A. W.; and Salakhutdinov, R. 2020. Exploring controllable text generation techniques. *arXiv preprint arXiv:2005.01822*.

Pu, D.; Wang, Y.; and Demberg, V. 2023. Incorporating Distributions of Discourse Structure for Long Document Abstractive Summarization. *arXiv preprint arXiv:2305.16784*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving Language Understanding by Generative Pre-Training. Publisher: OpenAI.

Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; Tow, J.; Rush, A. M.; Biderman, S.; Webson, A.; Ammanamanchi, P. S.; Wang, T.; Sagot, B.; Muennighoff, N.; del Moral, A. V.; Ruwase, O.; Bawden, R.; Bekman, S.; McMillan-Major, A.; Beltagy, I.; Nguyen, H.; Saulnier, L.; Tan, S.; Suarez, P. O.; Sanh, V.; Laurençon, H.; Jernite, Y.; Launay, J.; Mitchell, M.; Raffel, C.; Gokaslan, A.; Simhi, A.; Soroa, A.; Aji, A. F.; Alfassy, A.; Rogers, A.; Nitzav, A. K.; Xu, C.; Mou, C.; Emezue, C.; Klamm, C.; Leong, C.; van Strien, D.; Adelani, D. I.; Radev, D.; Ponferrada, E. G.; Levkovizh, E.; Kim, E.; Natan, E. B.; De Toni, F.; Dupont, G.; Kruszewski, G.; Pistilli, G.; Elsahar, H.; Benyamina, H.; Tran, H.; Yu, I.; Abdulmumin, I.; Johnson, I.; Gonzalez-Dios, I.; de la Rosa, J.; Chim, J.; Dodge, J.; Zhu, J.; Chang, J.; Frohberg, J.; Tobing, J.; Bhattacharjee, J.; Almubarak, K.; Chen, K.; Lo, K.; Von Werra, L.; Weber, L.; Phan, L.; allal, L. B.; Tanguy, L.; Dey, M.; Muñoz, M. R.; Masoud, M.; Grandury, M.; Šaško, M.; Huang, M.; Coavoux, M.; Singh, M.; Jiang, M. T.-J.; Vu, M. C.; Jauhar, M. A.; Ghaleb, M.; Subramani, N.; Kassner, N.; Khamis, N.; Nguyen, O.; Espejel, O.; de Gibert, O.; Villegas, P.; Henderson, P.; Colombo, P.; Amuok, P.; Lhoest, Q.; Harliman, R.; Bommasani, R.; López, R. L.; Ribeiro, R.; Osei, S.; Pyysalo, S.; Nagel, S.; Bose, S.; Muhammad, S. H.; Sharma, S.; Longpre, S.; Nikpoor, S.; Silberberg, S.; Pai, S.; Zink, S.; Torrent, T. T.; Schick, T.; Thrush, T.; Danchev, V.; Nikoulina, V.; Laippala, V.; Lepercq, V.; Prabhu, V.; Alyafeai, Z.; Talat, Z.; Raja, A.; Heinzerling, B.; Si, C.; Taşar, D. E.; Salesky, E.; Mielke, S. J.; Lee, W. Y.; Sharma, A.; Santilli, A.; Chaffin, A.; Stiegler, A.; Datta, D.; Szczechla, E.; Chhablani, G.; Wang, H.; Pandey, H.; Strobelt, H.; Fries, J. A.; Rozen, J.; Gao, L.; Sutawika, L.; Bari, M. S.; Al-shaibani, M. S.; Manica, M.; Nayak, N.; Teehan, R.; Albanie, S.; Shen, S.; Ben-David, S.; Bach, S. H.; Kim, T.; Bers, T.; Fevry, T.; Neeraj, T.; Thakker, U.; Raunak, V.; Tang, X.; Yong, Z.-X.; Sun, Z.; Brody, S.; Uri, Y.; Tojarieh, H.; Roberts, A.; Chung, H. W.; Tae, J.; Phang, J.; Press, O.; Li, C.; Narayanan, D.; Bourfoune, H.; Casper, J.; Rasley, J.; Ryabinin, M.; Mishra, M.; Zhang, M.; Shoeybi, M.; Peyrounette, M.; Patry, N.; Tazi, N.; Sanseviero, O.; von Platen, P.; Cornette, P.; Lavallée, P. F.; Lacroix, R.; Rajbhandari, S.;

Gandhi, S.; Smith, S.; Requena, S.; Patil, S.; Dettmers, T.; Baruwa, A.; Singh, A.; Cheveleva, A.; Ligozat, A.-L.; Subramonian, A.; Névéol, A.; Lovering, C.; Garrette, D.; Tunuguntla, D.; Reiter, E.; Taktasheva, E.; Voloshina, E.; Bogdanov, E.; Winata, G. I.; Schoelkopf, H.; Kalo, J.-C.; Novikova, J.; Forde, J. Z.; Clive, J.; Kasai, J.; Kawamura, K.; Hazan, L.; Carpuat, M.; Clinciu, M.; Kim, N.; Cheng, N.; Serikov, O.; Antverg, O.; van der Wal, O.; Zhang, R.; Zhang, R.; Gehrmann, S.; Mirkin, S.; Pais, S.; Shavrina, T.; Scialom, T.; Yun, T.; Limisiewicz, T.; Rieser, V.; Protasov, V.; Mikhailov, V.; Pruksachatkun, Y.; Belinkov, Y.; Bamberger, Z.; Kasner, Z.; Rueda, A.; Pestana, A.; Feizpour, A.; Khan, A.; Faranak, A.; Santos, A.; Hevia, A.; Unldreaj, A.; Aghagol, A.; Abdollahi, A.; Tammour, A.; HajiHosseini, A.; Behroozi, B.; Ajibade, B.; Saxena, B.; Ferrandis, C. M.; Contractor, D.; Lansky, D.; David, D.; Kiela, D.; Nguyen, D. A.; Tan, E.; Baylor, E.; Ozoani, E.; Mirza, F.; Ononiwu, F.; Rezanejad, H.; Jones, H.; Bhattacharya, I.; Solaiman, I.; Sedenko, I.; Nejadgholi, I.; Passmore, J.; Seltzer, J.; Sanz, J. B.; Dutra, L.; Samagaio, M.; Elbadri, M.; Mieskes, M.; Gerchick, M.; Akinlolu, M.; McKenna, M.; Qiu, M.; Ghauri, M.; Burynok, M.; Abrar, N.; Rajani, N.; Elkott, N.; Fahmy, N.; Samuel, O.; An, R.; Kromann, R.; Hao, R.; Alizadeh, S.; Shubber, S.; Wang, S.; Roy, S.; Viguier, S.; Le, T.; Oyebade, T.; Le, T.; Yang, Y.; Nguyen, Z.; Kashyap, A. R.; Palasciano, A.; Callahan, A.; Shukla, A.; Miranda-Escalada, A.; Singh, A.; Beilharz, B.; Wang, B.; Brito, C.; Zhou, C.; Jain, C.; Xu, C.; Fourrier, C.; Periñán, D. L.; Molano, D.; Yu, D.; Manjavacas, E.; Barth, F.; Fuhrimann, F.; Altay, G.; Bayrak, G.; Burns, G.; Vrabec, H. U.; Bello, I.; Dash, I.; Kang, J.; Giorgi, J.; Golde, J.; Posada, J. D.; Sivaraman, K. R.; Bulchandani, L.; Liu, L.; Shinzato, L.; de Bykhovetz, M. H.; Takeuchi, M.; Pàmies, M.; Castillo, M. A.; Nezhurina, M.; Sänger, M.; Samwald, M.; Cullan, M.; Weinberg, M.; De Wolf, M.; Mihaljcic, M.; Liu, M.; Freidank, M.; Kang, M.; Seelam, N.; Dahlberg, N.; Broad, N. M.; Muellner, N.; Fung, P.; Haller, P.; Chandrasekhar, R.; Eisenberg, R.; Martin, R.; Canalli, R.; Su, R.; Su, R.; Cahyawijaya, S.; Garda, S.; Deshmukh, S. S.; Mishra, S.; Kiblawi, S.; Ott, S.; Sang-aroonsiri, S.; Kumar, S.; Schweter, S.; Bharati, S.; Laud, T.; Gigant, T.; Kainuma, T.; Kusa, W.; Labrak, Y.; Bajaj, Y. S.; Venkatraman, Y.; Xu, Y.; Xu, Y.; Xu, Y.; Tan, Z.; Xie, Z.; Ye, Z.; Bras, M.; Belkada, Y.; and Wolf, T. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. arXiv:2211.05100 [cs].

Taboada, M., and Mann, W. C. 2006. Applications of rhetorical structure theory. *Discourse studies* 8(4):567–588. Publisher: Sage Publications Sage CA: Thousand Oaks, CA.

Vander Linden, K., and Martin, J. H. 1995. Expressing rhetorical relations in instructional text: A case study of the purpose relation. *Computational Linguistics* 21(1):29–57.

Wu, J.; Gan, W.; Chen, Z.; Wan, S.; and Lin, H. 2023. AI-Generated Content (AIGC): A Survey. arXiv:2304.06632 [cs].

Zhang, H.; Song, H.; Li, S.; Zhou, M.; and Song, D. 2023. A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models. arXiv:2201.05337 [cs].

Zhou, P.; Gopalakrishnan, K.; Hedayatnia, B.; Kim, S.; Pujara, J.; Ren, X.; Liu, Y.; and Hakkani-Tur, D. 2022. Think Before You Speak: Explicitly Generating Implicit Commonsense Knowledge for Response Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1237–1252. Dublin, Ireland: Association for Computational Linguistics.

Zhou, W.; Jiang, Y. E.; Wilcox, E.; Cotterell, R.; and Sachan, M. 2023. Controlled Text Generation with Natural Language Instructions. arXiv:2304.14293 [cs].

Action Item Driven Summarization of Long Meeting Transcripts

Logan Golia
Rice University
6100 Main St, Houston, TX
lsg3@rice.edu

Jugal Kalita
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO
jkalita@uccs.edu

Abstract

The increased prevalence of online meetings has sparked the practicality of a model that can automatically generate the summary of a given meeting. This paper introduces a novel and effective approach to automating the generation of meeting summaries. Current approaches to this problem generate very general and basic summaries of the dialogue. However, our novel algorithms can generate abstractive meeting summaries that are driven by the action items contained in the meeting transcript. This is done by recursively generating summaries and employing our action item extraction algorithm for each section of the meeting in parallel. All of these sectional summaries are then combined and summarized together to create a coherent and action item driven meeting summary. In addition, this paper introduces 3 novel methods for dividing up long meeting transcripts into topic-based sections to improve the time efficiency of our algorithm, as well as to resolve the issue of LLMs forgetting long-term dependencies. Our pipeline achieved a BERTScore of 64.98 across the AMI corpus, which is a $\approx 4.98\%$ increase from the current state-of-the-art results also employing a fine-tuned BART (Bidirectional and Auto-Regressive Transformers) model.

1 Introduction

As a result of the COVID-19 pandemic, many professional meetings and conversations have been conducted online. This also means that the transcripts of these meeting have become readily available. As humans, we cannot possibly attend or remember the contents of every single meeting that we are interested in, so we conduct the tedious process of generating meeting minutes. However, with the help of large language models (LLMs), we can automate this process of writing meeting summaries and still generate factual and informative summaries.

There are two main approaches to text summarization in general: extractive summarization and abstractive summarization. Extractive summarization techniques aim to locate the most important phrases and sentences from the input transcript and concatenate them together to form a concise summary. However, the summaries generating from these techniques are usually very awkward to read because we are forcefully concatenating these unrelated sentences together (Koh et al. 2023). Abstractive summarization techniques focus more on understanding the overall meaning of a transcript and then generating a concise summary based on the entire text. Unlike extractive summarization, abstractive summarization actually aims to generate new words and phrases in the summary that were not found in the input transcript, rather than simply extracting the important phrases (Rennard et al. 2023). Abstractive summarization is a much more challenging task, but as expected, it leads to better summaries (Gupta and Gupta 2019). As a result, meeting summarization has began to head in this direction, and this study utilizes abstractive summarization techniques as well.

Current approaches to automating meeting minutes is that they treat summarizing a meeting the same way they would summarizing a dialogue (FM et al. 2022). However, we argue that the meeting summarization problem is fundamentally different from the dialogue summarization problem. Unlike a dialogue, useful meeting minutes have some additional features that are often not included in the automated summary of the meeting: action items, main topics, tension levels, decisions made, etc. In this study, we focus on incorporating action items in the machine-generated summaries.

LLMs today still struggle to capture long-term dependencies in texts, and as a result, they are not very good at generating summaries for long transcripts (Dong et al. 2023). The time and space complexity of these transformer-based models increases quadratically with respect to the input size (Vaswani et al. 2017), and new LLMs still have strict input token limits (Yang et al. 2023). Most solutions to these problems employ linear segmentation, where the long texts are broken up into equal subsections based on token numbers, but the problem with this approach is that we are inevitably interrupting ideas in the text. We build upon previous work in text clustering to divide the text into topical chunks before summarizing (Chen et al. 2023).

In summary, current solutions to the problem of automatically generating meeting minutes given the transcript of the meeting produce very general and vague summaries. In addition, there is a lack of effective topic segmentation methods in the field of meeting summarization. This study outlines a novel method of utilizing topic segmentation and recursive summarization to generate action item driven abstractive summaries of long meeting transcripts.

Our main contributions are threefold:

1) We develop three novel topic segmentation algorithms, in which the best outperforms the summarization performance provided by linear segmentation by 1.36% in terms of the BERTScore metric;

2) We develop our own effective action item extraction algorithm;

3) Our novel parallel and recursive meeting summarization algorithm properly generates action item driven summaries and improves upon the performance of current state-of-the-art models by $\approx 4.98\%$ in terms of the BERTScore metric.

2 Related Work

In this section, we address previous methods employed in meeting summarization and provide motivation for our novel techniques.

2.1 Recursive Summarization

Another way in which the meeting summarization problem differs from the dialogue summarization problem is that meeting transcripts are generally very long, and as explained earlier, transformer-based models struggle with larger input sizes. As a result, it has been proven effective to divide long documents into multiple parts, summarize each component, and then combine the summaries back together in a recursive approach. The recursive algorithm described in this paper is inspired by the method proposed by (Wu et al. 2021) which was used to summarize long books. The methods proposed by (Shinde et al. 2022) and (Yamaguchi et al. 2021) are not truly recursive because after they combine the summaries back together, the final summary is never fed back into the summarization model. Instead, they perform argument mining on the resulting chunk of the combined summaries. We propose a truly recursive approach and achieve state-of-the-art results with this technique.

2.2 BART Model for Meeting Summarization

While there do exist more powerful dialogue summarization models such as DialogLM (Zhong et al. 2022) and Summ^N (Zhang et al. 2022), we use the BART (Bidirectional and Auto-Regressive Transformers) model (Lewis et al. 2020) due to its speed and high performance in long document summarization tasks (Koh et al. 2023). In addition, there has been previous research in assessing different topic segmentation methods on the BART model, so this allows us to evaluate our techniques.

2.3 AMI Dataset

The AMI dataset is a large meeting corpus consisting of 137 scenario-driven meetings and their corresponding summaries (Mccowan et al. 2005). Even though the scenarios are artificial, the way in which the actors choose how to interact with each other is spontaneous. The realistic meeting conversations combined with the fact that there are 137 different long meeting transcripts makes the AMI corpus an ideal dataset to test our techniques on.

2.4 Current Segmentation Techniques

There are many techniques to divide meeting transcripts into multiple parts, but none have actually been able to improve results when compared to the simplest technique, linear segmentation. Linear segmentation is the process of dividing the meeting transcripts into parts solely based on the number of tokens, maximizing the number of tokens in each section. The state-of-the-art results on summarizing the AMI corpus using the BART model are achieved through this technique by (Shinde et al. 2022). They attempted to use two additional topic segmentation techniques, Depth-Scoring by (Solbiati et al. 2021) and TextTiling by (Hearst 1997), but neither were able to improve the results obtained by linear segmentation. (Yamaguchi et al. 2021) also introduces a novel technique for topic segmentation using a Longformer+LSTM model to predict whether a sentence is the start of a new topic, in the middle of a topic, or outside of a particular topic. However, their summarization results were significantly less than those achieved by (Shinde et al. 2022). We propose three novel segmentation techniques that outperform linear segmentation.

2.5 Evaluation Metrics

ROUGE scores are the most popular metric in evaluating the precision and recall of the machine generated summaries. ROUGE-1 and ROUGE-2 scores are calculated by computing the n-gram overlap between the machine-generated and human reference summaries, where n equals 1 or 2 respectively. ROUGE-L scores are better at computing the similarities between sentence-level structures. It works by evaluating the Longest Common Subsequence (LCS), the longest sequence of words that appear in the same order in two texts, between the machine-generated summary and the human reference summary. The LCS does not have to be contiguous which is why ROUGE-L scores are generally preferred over ROUGE-1 and ROUGE-2 scores for evaluating summarization tasks. We employ ROUGE's F1 scores to provide a balanced measure of precision and recall (Lin 2004).

Even though the ROUGE scores are the most popular metric, they has many flaws since they focus solely on lexical overlap between the machine-generated summaries and the human reference summaries rather than their semantic similarity(Fabbri et al. 2021). As a result, BERTScore, which measures the semantic similarity between the machine-generated summaries and the human reference summaries has been growing in popularity (Rennard et al. 2023). BERTScore works by using the contextualized word embeddings provided by BERT to compute the token similarities between each token in the machine-generated summaries and each token in the human reference summaries. We employ the BERTScore metric as well, since it has been shown to achieve higher correlations with human judgment on the quality of a machine-generated summary compared to ROUGE (Zhang et al. 2020).

3 Approach

In this section, we dive deeper into our parallel and recursive algorithm for generating action item driven meetings. We

also explore the lower-level techniques that were necessary to improve state-of-the-art results and provide motivation for these design decisions along the way.

3.1 Divide-and-conquer

As described in our "Introduction" and "Related Works" sections, the first step to summarizing long meeting transcripts is to break them up, so we can summarize each chunk. We propose three simple but very effective topic segmentation techniques that were able to generate more truthful and concise summaries when compared to linear segmentation.

Chunked Linear Segmentation When we ran our model using linear segmentation, we noticed that points were often misunderstood and repeated because we were creating separate chunks in the middle of a speaker's formulation of one idea. Let us call each speaker's contiguous dialogue a "turn". Therefore, we first employed a simple technique inspired by linear segmentation where we attempt to maximize the number of tokens in each chunk, but ensuring that no speaker's turn is interrupted.

Simple Cosine Similarity The second technique we created is based upon chunked linear segmentation, but also upon the cosine similarity of the MPNet embeddings, a state-of-the-art sentence embedding model (Song et al. 2020), for each turn. This allows us to compute the semantic similarity between each turn. For each turn, we compute its MPNet embedding and calculate its cosine similarity with the MPNet embedding of the previous turn. These values range from -1 to 1 where a cosine similarity of -1 means that the sentences are extremely dissimilar, and a cosine similarity of +1 means that the sentences are extremely similar. If the cosine similarity of the embeddings is greater than 0, we simply add this turn to the current chunk. If the cosine similarity of the embeddings is less than or equal to 0, we define the current turn as the beginning of a new topic and begin a new chunk.

We choose a similarity threshold of 0 to signify the start of a new topic is after experimenting with different values and manually inspecting the quality of the resulting summaries, as well as evaluating the resulting summaries with ROUGE and BERTScore metrics. This value of 0 also makes sense in theory because it means that the two consecutive turns are more dissimilar than they are similar. In our experimentation with calculating the cosine similarities between turns in the AMI Corpus, negative values were relatively uncommon, but still arose. However, it makes sense that this leads to better results because we do not want to split the transcript into too many topics, and instead favor large topics, because we generally want to keep as much text intact as possible so the summarization model has enough context to generate a quality summary. We do not want to be generating too many independent summaries for each topic that have little relation to each other and then combining these little summaries together. In our testing, this proved to be a very ineffective approach because each sectional summary had little context of the surrounding text to work with, and as a result, the resulting overall summary was very confusing to read. This is also why topic segmentation for summarization is a very different problem from typical topic segmentation problems because we do not want to create chunks at every little topic change. In fact, when we increased our similarity threshold from 0 to just 0.2, our BERTScores and ROUGE-L scores both decreased by $> 1\%$ which is very significant for summarization tasks.

It is also important to note that when splitting based on some cosine similarity threshold, there is a risk that no new chunks will be created for over 1024 consecutive tokens, which is the max input token limit for the BART model (Obonyo, Casola, and Saggion 2022). In this event, we will not be able to pass this large chunk of text into our summarization model. Therefore, we developed a solution to this problem. As we move through the turns and add them to the existing chunk if their cosine similarities with the previous turn is greater than or equal to 0, we check to ensure that adding the current turn will not make the current chunk greater than 1024 tokens. If this turn will make the current chunk greater that 1024 tokens, we create a new chunk/topic beginning with this turn, regardless of this turn's cosine similarity with the previous turn. With this technique, we still create topic-based chunks of the meeting transcript whilst ensuring that no topic/chunk exceeds 1024 tokens.

Complex Cosine Similarity The previous method worked fairly well, but we noticed a recurring problem when inspecting the topic chunks that were being created. Sometimes in the meeting transcript, a person would utter something meaningless, and that would compose their entire turn (e.g. "Bob: Ummm."). As a result, this turn would often have a really low cosine similarity with the previous turn, a new topic/chunk would be created. The simplest solution to this problem would be to remove all redundant and meaningless utterances in the pre-processing stage. The problem with this approach is that even if we somehow managed to hard code the regular expressions in order to remove all of the "meaningless" turns, there are still lots of cases where a speaker will say something completely unrelated to the current topic (e.g. "Let us go grab ice cream after this"), but then they will resume talking about the original topic. In this case, we would not want to create a new topic. In order to achieve this, we take the same approach used in "simple cosine similarity", except we recalculate the MPNet embedding of the entire current chunk before comparing its cosine similarity to the the MPNet embedding of the following sentence. Thus, we are not checking if the next turn is on the same topic as the previous turn, rather whether or not the next turn is on the same topic as the entire current chunk being created. Thus, we mitigate the effect of "meaningless" turns, particularly consecutive "meaningless" turns, since they will have a lesser impact on the the MPNet embedding of the chunk we are comparing the next turn to. Please refer to Algorithm 1 for further details.

3.2 Generating the General Sectional Suammries

Once we have divided the original text into chunks, the next step is to generate a general abstractive summary for each chunk. Our approach to solve this problem involves fine-

Algorithm 1 Complex Cosine Similarity(string text, int similarityThreshold, int maxTokens)

$turns \leftarrow text$ split by speaker
$model \leftarrow$ sentence embedding model
$tokenizer \leftarrow$ tokenizer used by summarization model
$processedChunks \leftarrow$ list with the first sentence from $turns$
for i in range(1, $len(turns)$) **do** ▷ Iterate through the turns
    $curChunkEmbedding \leftarrow model.encode(processedChunks[-1])$
    $nextSpeakerEmbedding \leftarrow model.encode(turns[i])$
    $similarity \leftarrow cosineSimilarity(curChunkEmbedding, nextSpeakerEmbedding)$ ▷ Compute similarity
    $newChunk \leftarrow processedChunks[-1]$ + $turns[i]$
    $newNumTokens \leftarrow tokenLen(tokenizer(newChunk))$
    if $similarity > similarityThreshold$ and $newNumTokens \leq maxTokens$ **then**
        $processedChunks[-1] \leftarrow newChunk$ ▷ Add turn to the current chunk
    else
        append $turns[i]$ to $processedChunks$ ▷ Start a new chunk
    end if
end for
return $processedChunks$ ▷ A list of topic-based chunks of $text$

tuning Meta's BART model (Lewis et al. 2020), a pre-trained large language model, on dialogue datasets to generate general summaries of a meeting. We elect to use a BART model since its bidirectional encoder and auto-regressive decoder has been shown to understand the full semantics of a text and generate coherent summaries. Specifically, we used a BART model fine-tuned on the XSUM (Narayan, Cohen, and Lapata 2018) and SAMSUM (Gliwa et al. 2019) datasets to generate the general summaries for each chunk. These are widely used dialogue datasets for training dialogue summarization models (Feng, Feng, and Qin 2022). They are also the same datasets (Shinde et al. 2022) fine-tuned their model on so can better compare our results.

In addition, we noticed that since each general sectional summary is independent of one another, they can be generated in parallel. To the best of our knowledge, we are the first to incorporate parallelism in the divide-and-conquer summarization algorithm as seen in Algorithm 3.

3.3 Action Item Extraction

Another very important component of any good meeting summary is what each participant has accomplished and what they need to accomplish before the next meeting. So, for each chunk of text, we need to extract the action items. Action item extraction is an extremely understudied topic, thus we developed our own method. To accomplish this, we use a public dataset[1] from a GitHub repository that contains 2750 dialogue statements as well as corresponding labels for whether each statement contains action items and which do not. We then fine-tune a BertForSequenceClassification[2] model (a BERT model transformer with a linear layer on top for classification) on this dataset in order to be able to classify the action items in the original meeting transcript. This training method proved very effective with a classification accuracy of 95.4% on the test dataset. However, this process alone is not enough to extract the key action items from a text. Through this method alone, we are only identifying which sentences contain action items, but we are not truly extracting the ideas underlying them. For example, a sentence that may be identified as an action item can be "You need to do that before the next meeting." This is indeed an action item, but it doesn't actually contain any useful information; there are too many pronouns and not enough context. In the next section, we discuss existing methods to solve this problem, explain their limitations for this application, and present our own technique.

Coreference Resolution We first employed widely used state-of-the-art methods and models for coreference resolution in order to convert the sentences that were classified as action items into more context-rich statements. We employed libraries such as Stanford CoreNLP (Clark and Manning 2016) and NeuralCoref[3] (an extension of the spaCy library), but we were not satisfied by the results. Not only were the pronouns not always resolved for larger text inputs, but we realized that coreference resolution alone was not enough to solve our problem. Even if the pronouns were resolved, this was often not enough context alone to completely understand the sentence containing the action item. For example, the sentence "you need to do that before the next meeting" may be converted to "Jake needs to fix the website before the next meeting" after coreference resolution. This is better, but it is still not enough information for Jake to read this sentence in the meeting minutes and understand what needs to be done. He doesn't know what specifically needs to be fixed in the website, or why it needs to be fixed at all.

Context Resolution In this paper, we employ our own technique to solve this lack-of-context problem which we

[1] https://github.com/kiransarv/actionitemdetection/blob/master/dataset

[2] https://huggingface.co/docs/transformers/v4.31.0/en/model_doc/bert#transformers.BertForSequenceClassification

[3] https://github.com/huggingface/neuralcoref

call ”neighborhood summarization.” For this method, once we find a sentence that has been identified as an action item, we then find its ”neighborhood.” For our purposes, we define a sentence's neighborhood as the three sentences before the sentence, the sentence itself, and the two sentences after the sentence. Finally, we use all 6 of these sentences as inputs into the same BART summarization model that we used to generate the sectional summaries, and we are left with a rephrased version of the sentence containing the action item. We believe the reason this technique works so well is because the human reference summaries in the dialogue datasets that our BART model is fine-tuned on are naturally action-item driven, to some extent. As a result, when we place the entire neighborhood into the summarization model, we can gain a context-rich summary revolving around the action item that often addresses the lack-of-context problem we discussed earlier. To use the same example, this neighborhood summarization technique can convert a sentence that has been identified as an action item, ”you need to do that before the next meeting”, into a context-rich rephrasing, ”Jake needs to fix the menu button on the website because our users are complaining that it does not work half the time.”

We choose 3 sentences before and two sentences after for our neighborhood after experimenting with different values and inspecting the resulting summaries ourselves. Any smaller of a neighborhood and we found that there was not enough context in the resulting summary. Any larger of a neighborhood, and the summary often did not revolve around the action item and instead addressed other parts of the input text that was not relevant for this particular action item extraction task. Also it makes sense that we would need more sentences before the action item than after it since most pronoun references and necessary context would be provided before a sentence that depends on it. However, since this is a dialogue summarization task, and there are many anomalies when people speak, sentences after the action item are still necessary to include in the neighborhood in the case that additional pronoun references or context comes after. Note that there are edge cases to this rule, for example when an action item is located at the very beginning or end of a chunk, so please see Algorithm 2 for more details.

Now that we have extracted the action items with context from a given chunk, we append each of them to the end of the general summary for this same chunk. This way, we can keep the summaries and action items that are derived from the same pieces of text together. Then we pass this entire text (summary + action items) into the same BART summarizer. We found that this technique helps condense the summary as well as improve the coherence of the resulting summary for each chunk.

3.4 Combining Summaries and the Recursive Case

Now that we have generated summaries for each chunk, containing information regarding both the general summary and the action items, we will generate an abstractive summary again based on all of the sectional summaries combined together in a recursive approach. If we append the sectional summaries together, and the number of tokens in this entire chunk of text is less than 1024, then we pass this entire chunk of summaries into the same BART summarizer again; in essence, we are summarizing the summaries. However, if this entire chunk of summaries contains more than 1024 tokens, then we fall into the recursive case where we pass this entire chunk of summaries back into the entire function as if it is a meeting transcript. We explored other techniques to fluidly combine the summaries together, but we found that using the BART summarizer achieved the best results. We attempted to use an existing RoBERTa model (Liu et al. 2019) that was fine-tuned on a sentence fusion dataset know as DiscoFuse (Rothe, Narayan, and Severyn 2020). However, this technique did not prove effective because the resulting summaries were often very long and contained repetitions between the summaries. We tried solving this problem by tuning the BART summarizer model to generate shorter sectional summaries, so the resulting chunk of all the summaries appended together would be shorter, but the sentence fusion models still did not prove effective in generating grammatically correct and coherent final summaries. This is a very challenging task if approached from a sentence fusion perspective, however, we approached this problem as simply another summarization task, and the fine-tuned BART summarizer proved very effective at this task by removing repetitions between the sectional summaries and generating very informative, coherent, and concise summaries as seen in our results table.

4 Results and Analysis

We first generated meeting summaries without including our action item extraction technique in order to evaluate our three topic segmentation techniques and recursive algorithm. We evaluate within our own techniques as well as compare to the current state-of-the-art on the AMI dataset using the BART summarizer (Shinde et al. 2022). Then we compare our summaries with and without action items and show that our action item driven summaries contain additional valuable information.

4.1 Topic Segmentation Performance

We evaluate our topic segmentation methods by keeping our recursive algorithm constant and only varying the topic segmentation method. We can see from Table 1 that all three of our novel topic segmentation methods outperformed linear segmentation with respect to both the BERTScore and ROUGE metrics. Most notably, with respect to the BERTScore metric, our methods, simple cosine similarity, complex cosine similarity, and chunked linear segmentation, outperform linear segmentation by 0.50% 1.07% and 1.36%, respectively for the generated summaries without action items. For the summaries with action items, the improvements over linear segmentation with respect to the BERTScore metric, were 0.38% 1.11% and 1.22%, respectively.

The complex cosine similarity technique outperformed the simple cosine similarity technique by 0.57% and 0.73% in terms of the BERTScore metric for the summaries without and with action items, respectively. This was expected

Algorithm 2 Action Item Extraction(string text)

$model \leftarrow$ action item classifier
$tokenizer \leftarrow$ BERT tokenizer.
$actions \leftarrow$ empty string
$sentences \leftarrow text$ split by sentence
for $index$, $sentence$ in enumerate($sentences$) **do** ▷ Iterate through the sentences
    $inputs \leftarrow tokenizer(sentence)$
    $predictedClass \leftarrow model(inputs)$.
    if $predictedClass = 1$ **then** ▷ Class 1 indicates $sentence$ is an action item
        $neighborhood \leftarrow$ empty string
        $startIndex \leftarrow max(0, index - 3)$.
        $endIndex \leftarrow min(len(sentences), index - 3)$
        for $neighborIdx$ in range($startIndex$, $endIndex$) **do**
            $neighborhood$ += $sentences[neighborIdx]$.
        end for
        $actions$ += $generalSum(neighborhood)$ ▷ Summarize the neighborhood
    end if
end for
return $actions$ ▷ A string containing the context-rich action items found in $text$

Algorithm 3 Action Item Driven Summary(string text, bool first, int maxTokens)

$tokenizer \leftarrow$ tokenizer used by summarization model
$text \leftarrow preProcessText(text)$
if $first = True$ **then**
    $chunks \leftarrow topicalChunksBySpeaker(text)$ ▷ Split text into topic-based chunks
else
    $chunks \leftarrow topicalChunksBySentence(text)$
end if
$chunkSums \leftarrow array\ with\ size\ of\ len(chunks)$
for all $index \in range(0, len(chunks))$ **do** ▷ Summarize each chunk in parallel
    $part \leftarrow chunks[index]$
    $genSum \leftarrow generalSum(part)$
    if $first = True$ **then**
        $actions \leftarrow actionItemExtraction(chunk)$ ▷ Extract action items
        $combined \leftarrow genSum + actions$
        $combinedNumTokens \leftarrow tokenLen(tokenizer(genSum + actions))$
        if $combinedNumTokens > maxTokens$ **then** ▷ Theoretically possible but never true in our testing
            $combined \leftarrow truncateText(combined)$
        end if
        $chunkSum \leftarrow generalSum(combined)$
        $chunkSums[index] \leftarrow partSum$
    else
        $chunkSums[index] \leftarrow genSum$
    end if
end for
$concatSums \leftarrow concatenate(chunkSums)$ ▷ Concatenate summaries after parallel loop completes
$summaryNumTokens \leftarrow tokenLen(tokenizer(concatSums))$
if $summaryNumTokens > maxTokens$ **then**
    return $actionItemDrivenSummary(concatSums, False, maxTokens)$ ▷ Recursive call
else
    return $generalSum(concatSums)$ ▷ The action item driven summary of $text$
end if

Topic Segmentation ↓ Metric →	BERTScore	R-1	R-2	R-L
General Summaries (Without Action Items)				
Linear Segmentation (Baseline Technique)	63.41	38.14	8.61	19.46
Chunked Linear Segmentation	**64.77**	**38.93**	**9.27**	**19.63**
Simple Cosine Similarity	63.91	38.49	8.61	19.46
Complex Cosine Similarity	64.48	38.92	9.24	19.47
Action Item Driven Summaries				
Linear Segmentation (Baseline Technique)	63.76	35.11	8.04	18.99
Chunked Linear Segmentation	**64.98**	**36.27**	8.31	**19.62**
Simple Cosine Similarity	64.14	35.30	8.12	19.24
Complex Cosine Similarity	64.87	36.21	**8.32**	19.61
(Shinde et al. 2022)	60	45.2	13.3	N/A

Table 1: BERTScore and ROUGE evaluation scores for our machine-generated summaries across 4 different topic segmentation methods on the AMI corpus. This is done separately for both the general summaries (without action items) and the action item driven summaries. We also include the scores achieved by the current state-of-the-art model (Shinde et al. 2022)

because the former was less sensitive to "meaningless turns" as explained in the "Complex Cosine Similarity" subsection. However, chunked linear segmentation, which does not rely on word embeddings and cosine similarity, outperformed all.

4.2 Recursive Algorithm Performance

We also compare the results of our recursive algorithm to those of (Shinde et al. 2022). When we both use linear segmentation and the same fine-tuned BART models, but different "recursive" algorithms, our action item driven model outperforms the model presented by (Shinde et al. 2022) by $\approx 4.98\%$ in terms of the BERTScore metric. With regard to our general summarization model (without action items), this model still outperformed that presented by (Shinde et al. 2022) by $\approx 4.77\%$. This means that, regardless of whether or not we include action items, the summaries our model generates are more similar to those of the human reference summaries in terms of their semantic meanings.

The model by (Shinde et al. 2022) does outperform our model in terms of the ROUGE scores, which measure lexical overlap, but this is expected since we use a truly recursive algorithm that results in the input text and the corresponding sectional summaries being passed into the BART summarizer more times. This would, of course, decrease the lexical overlap between our machine-generated summaries and the human reference summaries. However, it seems that our summaries better match the semantic meaning of the human reference summaries, which was shown to be more important for human judgement by (Zhang et al. 2020).

4.3 Action Item Driven Summary Performance

As can be seen in Table 1, our action item driven summaries seemed to achieve slightly higher BERTScores than our general summaries (without action items), but we consider this difference negligible (0.21% increase in BERTScore when both using chunked linear segmentation). However, we suspect that the reason for this small difference is that the human reference summaries in the AMI dataset appear to be more action item driven that those in the XSUM and SAMSUM datasets.

The ROUGE scores for our action item driven summaries were notably lower than those achieved by our general summaries. For example, when both techniques employ chunked linear segmentation, the ROUGE-1 scores for our general summaries were 1.66% higher than those for our action item driven summaries. However, this makes sense since we are deliberately adding words and phrases (action items) that are not included in the human reference summaries; thus, our precision score decreases. However, the slight increase in our BERTScores suggests that we are still capturing the semantic meanings of the reference summaries well.

Table 2 shows example outputs from our general model and our action item driven model. We underline the additions in the action item driven summary and show that our action item driven model properly includes relevant action items from the meeting. Consider the following sentence from the action driven summary: "Industrial Designer tells Product Manager they need to get double A or triple A batteries." This action item is not included in either the general summary or the human reference summary, but it is a relevant and informative action item that adds value to the meeting summary. We also see that this action item is coherent and rich with context; we know who is asking for the batteries, and who needs to obtain them. This example, as well as the other sentences underlined in Table 2, serves as evidence that our action item extraction technique utilizing neighborhood summarization is quite effective.

5 Future Research

In this study, we focused on generating action item driven summaries, but there are additional components of a good meeting summary. As noted in our introduction, decisions made, main topics, tension levels, etc. would also be very informative aspects of a meeting summary. While incorporating these elements into a meeting summary may lower our

General Summary (Without Action Items)	Action Item Driven Summary
Marketing Expert, Product Manager, and Industrial Designer are having a conceptual design meeting after lunch. They talk about the most important aspect for remote controls as people want a fancy look and feel. They discuss the size of the batteries they need to take into consideration, the design of the LCD display on the LCD screen, how to distinguish where people have to press the button when they have a flip-top, and how to incorporate voice recognition into the remote control. They agree on keeping the control buttons standardized and checking the financial feasibility. They decide to start with the black and white one and go for a whistle if financially voice recognition is not feasible. The product will have a logo on it just like everything else in a year's time if they get feedback from design fairs. Product manager will go through the end of the end meeting. Marketing Expert shares some information about a remote control that fits into the palm of the hand, made of plastic, with a rubberised cover, and the design is based on the input from the previous meeting.	Marketing Expert, Product Manager, and Industrial Designer are having a conceptual design meeting after lunch. They talk about properties, materials, user-interface and trend-watching. Marketing Expert says the fashion update which relates to very personal preferences among their subject group. There's no rechargeable option for the remote control, so they're going to look into battery options. Industrial Designer and Marketing Expert are talking about the size of the batteries they need to take into consideration. Marketing Expert thinks using the standard batteries and the solar charging will detract from the attractiveness of the whole feature. Marketing Expert thinks the buttons on the remote should have lights behind the buttons. Marketing Expert wants to make the basic mold out of plastic but have a rubber cover. Marketing experts are going to market to guys as much as to women. Marketing Expert shares with Industrial Designer some information about the design of the LCD display on the LCD screen. Industrial Designer and Marketing Expert are discussing how to incorporate voice recognition into the remote control. Industrial Designer tells Product Manager they need to get double A or triple A batteries. Sarah and Marketing Expert are talking about the design of a remote control with a rubberised cover. Industrial Designer tells Marketing Expert they can go for a whistle if voice recognition is not feasible. Product Manager will wrap up the end-of-meeting message.

Table 2: Comparison between machine-generated General (Without Action Items) and Action Item Driven Summaries. The additions in the action item driven summary are underlined. AMI Meeting ID: ES2004c

automated evaluation scores, this does not necessarily mean that the resulting meeting summary would be less useful for human readers. We hope to explore current approaches and develop new algorithms to extract these ideas from a meeting transcript and then incorporate them into a meeting summary.

While all three of our novel topic segmentation techniques outperformed linear segmentation, our best performance came from chunked linear segmentation, which did not involve calculating any embeddings or cosine similarities. However, the fact that chunked linear segmentation outperformed linear segmentation suggests the idea that we can generate better summaries by not interrupting any ideas in the meeting transcript. Thus, we hope to develop a more advanced topic segmentation method that will be able to lead to better summaries and outperform chunked linear segmentation.

Finally, action item extraction is an extremely understudied research topic with both a lack of techniques as well as metrics for evaluating these techniques. Thus, we hope to dive deeper into this field and invent more advanced techniques for accomplishing the two above goals. Nevertheless, our neighborhood summarization algorithm proved very effective in action item extraction, and we hope to test its performance on other tasks involving context resolution as well (e.g. extracting decisions made from a meeting).

6 Conclusion

This study explores a novel method for automatically generating meeting summaries by treating this problem as a fundamentally different one from that of generating dialogue summaries. Action items drive this recursively-generated, abstractive summary of the meeting that achieves $\approx 4.98\%$ higher BERTScores across the AMI corpus than the previous state-of-the-art using the BART summarizer. We introduce novel topic segmentation and action item extraction algorithms that all improve and add value to the resulting summaries. The recursive approach presented in this paper of generating summaries for different parts and aspects of the meeting transcript can be expanded upon to improve meeting summarization, as well as be generalized and applied to summarizing other genres of text in the future.

7 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[Chen et al. 2023] Chen, Z.; Mi, C.; Duo, S.; He, J.; and Zhou, Y. 2023. ClusTop: An unsupervised and integrated text clustering and topic extraction framework.

[Clark and Manning 2016] Clark, K., and Manning, C. D. 2016. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 643–653. Berlin, Germany: Association for Computational Linguistics.

[Dong et al. 2023] Dong, Z.; Tang, T.; Li, L.; and Zhao, W. X. 2023. A Survey on Long Text Modeling with Transformers. arXiv:2302.14502 [cs].

[Fabbri et al. 2021] Fabbri, A. R.; Kryściński, W.; McCann, B.; Xiong, C.; Socher, R.; and Radev, D. 2021. SummEval: Re-evaluating Summarization Evaluation. Transactions of the Association for Computational Linguistics 9:391–409.

[Feng, Feng, and Qin 2022] Feng, X.; Feng, X.; and Qin, B. 2022. A Survey on Dialogue Summarization: Recent Advances and New Frontiers. arXiv:2107.03175 [cs].

[FM et al. 2022] FM, M. F. A.; S, P.; M, G.; and J, J. 2022. Automation of Minutes of Meeting (MoM) using Natural Language Processing (NLP). In 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), 1–6.

[Gliwa et al. 2019] Gliwa, B.; Mochol, I.; Biesek, M.; and Wawer, A. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, 70–79. arXiv:1911.12237 [cs].

[Gupta and Gupta 2019] Gupta, S., and Gupta, S. K. 2019. Abstractive summarization: An overview of the state of the art. Expert Systems with Applications 121:49–65.

[Hearst 1997] Hearst, M. A. 1997. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. Computational Linguistics 23(1).

[Koh et al. 2023] Koh, H. Y.; Ju, J.; Liu, M.; and Pan, S. 2023. An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics. ACM Computing Surveys 55(8):1–35.

[Lewis et al. 2020] Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7871–7880. Online: Association for Computational Linguistics.

[Lin 2004] Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries.

[Liu et al. 2019] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs].

[Mccowan et al. 2005] Mccowan, I.; Carletta, J.; Kraaij, W.; Ashby, S.; Bourban, S.; Flynn, M.; Guillemot, M.; Hain, T.; Kadlec, J.; Karaiskos, V.; Kronenthal, M.; Lathoud, G.; Lincoln, M.; Lisowska Masson, A.; Post, W.; Reidsma, D.; and Wellner, P. 2005. The ami meeting corpus. Int'l. Conf. on Methods and Techniques in Behavioral Research.

[Narayan, Cohen, and Lapata 2018] Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. arXiv:1808.08745 [cs].

[Obonyo, Casola, and Saggion 2022] Obonyo, I.; Casola, S.; and Saggion, H. 2022. Exploring the limits of a base BART for multi-document summarization in the medical domain.

[Rennard et al. 2023] Rennard, V.; Shang, G.; Hunter, J.; and Vazirgiannis, M. 2023. Abstractive Meeting Summarization: A Survey. arXiv:2208.04163 [cs].

[Rothe, Narayan, and Severyn 2020] Rothe, S.; Narayan, S.; and Severyn, A. 2020. Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. Transactions of the Association for Computational Linguistics 8:264–280. arXiv:1907.12461 [cs].

[Shinde et al. 2022] Shinde, K.; Ghosal, T.; Singh, M.; and Bojar, O. 2022. Automatic Minuting: A Pipeline Method for Generating Minutes from Multi-Party Meeting Transcripts.

[Solbiati et al. 2021] Solbiati, A.; Heffernan, K.; Damaskinos, G.; Poddar, S.; Modi, S.; and Cali, J. 2021. Unsupervised Topic Segmentation of Meetings with BERT Embeddings. arXiv:2106.12978 [cs].

[Song et al. 2020] Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. arXiv:2004.09297 [cs].

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. arXiv:1706.03762 [cs].

[Wu et al. 2021] Wu, J.; Ouyang, L.; Ziegler, D. M.; Stiennon, N.; Lowe, R.; Leike, J.; and Christiano, P. 2021. Recursively Summarizing Books with Human Feedback.

[Yamaguchi et al. 2021] Yamaguchi, A.; Morio, G.; Ozaki, H.; Yokote, K.-i.; and Nagamatsu, K. 2021. Team Hitachi @ AutoMin 2021: Reference-free Automatic Minuting Pipeline with Argument Structure Construction over Topic-based Summarization.

[Yang et al. 2023] Yang, X.; Li, Y.; Zhang, X.; Chen, H.; and Cheng, W. 2023. Exploring the Limits of ChatGPT for Query or Aspect-based Text Summarization. arXiv:2302.08081 [cs].

[Zhang et al. 2020] Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs].

[Zhang et al. 2022] Zhang, Y.; Ni, A.; Mao, Z.; Wu, C. H.; Zhu, C.; Deb, B.; Awadallah, A. H.; Radev, D.; and Zhang, R. 2022. Summˆn: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents. arXiv:2110.10150 [cs].

[Zhong et al. 2022] Zhong, M.; Liu, Y.; Xu, Y.; Zhu, C.; and Zeng, M. 2022. DialogLM: Pre-trained Model for Long Dialogue Understanding and Summarization.

Author Index