AMOD 5310H – Artificial Intelligence
Group members(1):
Diego Brito – 0814117

Project Report

Introduction

The integration of Adaptive Data Optimization (ADO) into machine learning frameworks presents a significant opportunity to enhance domain-specific training through dynamic probability adjustment. This research evaluates ADO's practical implementation in language modeling tasks, specifically utilizing the GPT-2 architecture and Wikitext-2 dataset. The investigation centers on ADO's capacity to optimize training processes while operating within computational constraints.

The study addresses three primary research questions that guide our methodology: the impact of ADO on training efficiency through dynamic probability adjustment, its practical deployment in resource-constrained environments, and its performance comparison against traditional uniform sampling methods. These questions frame our experimental approach and evaluation metrics.

This framework employed a systematic implementation strategy. The process was initiated by carefully selecting and preprocessing 20% of the Wikitext-2 dataset, ensuring enough domain diversity while maintaining computational efficiency. The implementation phase integrated ADO with the GPT-2 small model, incorporating specialized modifications for domain-specific loss tracking and probability management. Training execution spanned three epochs, striking a balance between computational resources and result validity.

The experimental results revealed interesting patterns in ADO's performance characteristics. Domain sampling probabilities demonstrated unexpected stability, maintaining largely uniform distributions throughout the training process. However, subtle variations in sampling counts indicated minimal adaptive behavior. Domain-specific loss analysis showed partial success in difficulty recognition, though the algorithm's ability to leverage this information for optimal sampling remained limited.
A significant finding emerged in the validation loss patterns, where increasing values across epochs suggested potential overfitting issues. This observation highlights the challenges of optimizing domain prioritization within constrained experimental parameters. These results provide valuable insights into ADO's practical limitations and opportunities for enhancement.

The report follows a structured approach to detail the findings. This begins with an extensive examination of ADO's theoretical foundations and implementation specifics, followed by a detailed description of the experimental methodology. The results section provides in-depth analysis supported by visual representations and baseline comparisons. The report concludes with practical recommendations and future research directions.

This introduction establishes the foundational framework for understanding my research objectives, methodological approach, and key findings, setting the stage for detailed examination in subsequent sections. Through this investigation, I contribute to the growing body of knowledge regarding practical applications of adaptive optimization techniques in machine learning contexts.

Description of the tool and Algorithm

The Adaptive Data Optimization (ADO) algorithm, central to this project, represents a novel approach to improving machine learning workflows by dynamically adjusting data sampling probabilities during training. Traditional machine learning practices rely heavily on fixed data sampling policies determined in advance, requiring practitioners to train smaller proxy models for policy evaluation and optimization. This conventional method presents significant disadvantages, consuming more time and computational resources due to the necessary pre-training phase before target model training can start. Additionally, the proxy model approach introduces considerable risks of degraded results when its behavior does not align accurately with the target model's performance on the final task.

ADO addresses these fundamental challenges by integrating data policy optimization directly into the training process, eliminating the requirement for separate pre-training phases. The algorithm's core innovation manifests in its capacity to dynamically update sampling probabilities across different domains based on observed training performance. The process incorporates three critical mechanisms working in synchronization: domain loss calculation systematically evaluates losses for individual domains during each training iteration, providing essential insights into domain-specific learning challenges and forming the foundation for probability adjustments. The probability update rule implements scaling laws to modify domain sampling probabilities, ensuring increased attention to challenging domains while maintaining appropriate representation for easier domains to support effective generalization. The clipping function enforces probability constraints in predefined limits, preventing training variability from extreme adjustments while ensuring no domain is ignored or has excessive prioritization. Through these integrated mechanisms, ADO enables a real-time, adaptive optimization process that aligns precisely with the evolving needs of the model during its learning progression.

The conventional approach to determining data policies through pre-training presents several significant operational challenges that impact training efficiency and effectiveness. The computational requirements for training proxy models to identify optimal data sampling policies effectively double the necessary resources by introducing an additional pre-training phase. Once established, these policies remain static throughout the target model training, failing to accommodate the dynamic nature of learning progression and potentially leading to significant inefficiencies. Furthermore, potential misalignment between proxy and target model behaviors can result in suboptimal data utilization and reduced training effectiveness. ADO effectively addresses these limitations by implementing a dynamic system that changes alongside the model, reducing computational expense while enhancing adaptability through continuous refinement of data sampling strategies based on real-time performance feedback and observed learning patterns.

The integration of data policy optimization within the training process gives important practical and theoretical benefits for machine learning workflows. This innovative approach significantly streamlines the training pipeline by eliminating separate pre-training requirements while maintaining highly responsive data sampling strategies throughout the entire training process. Such dynamic adaptation enables more efficient resource allocation toward challenging domains and facilitates real-time adjustment to varying patterns in data complexity. ADO's implementation of adaptive sampling probabilities represents a significant advancement toward more efficient and resource-conscious machine learning practices, potentially improving both the speed and effectiveness of the learning process while maintaining comprehensive coverage across all domains. The algorithm's ability to continuously refine its sampling strategy based on real-time feedback ensures optimal resource utilization and improved learning outcomes throughout the training lifecycle.

This comprehensive overview establishes ADO's theoretical framework and practical significance, emphasizing its transformative potential in revolutionizing data policy management within machine learning applications. The subsequent section will explore in detail the experimental design methodology employed to evaluate ADO's effectiveness in practical applications and real-world scenarios.

Experiment Design

The experimental framework for this project evaluates Adaptive Data Optimization (ADO) through direct comparison with baseline approaches in a computationally constrained environment. This investigation aims to assess if ADO's dynamic sampling methodology delivers measurable improvements in training efficiency and model performance while operating under limited computational resources.

The investigation utilizes the Wikitext-2 dataset, a standard benchmark in language modeling research. To maintain the experiment computational achivable, the study employs 20% of the original dataset, preserving essential variety and domain diversity for meaningful evaluation. This reduction significantly decreases training duration from approximately eight hours per epoch to a more practical timeframe, enabling rapid iteration and experimental refinement. The preprocessing protocol implements specific steps to align with GPT-2 model requirements, including tokenization using the GPT-2 tokenizer and sequence standardization at 100 tokens maximum length. The dataset undergoes systematic division into distinct domains without thematic consideration, establishing uniform segments that facilitate ADO's domain-specific sampling methodology while streamlining implementation processes. This structured separation enables precise performance tracking across data types and supports dynamic sampling priority adjustments.

The research employs the GPT-2 small model as its foundational architecture, selected for its efficient performance in text generation while remaining manageable within project computational constraints. This implementation diverges from traditional approaches by utilizing the complete GPT-2 model for both baseline and ADO implementations, eliminating separate pre-training requirements. The baseline methodology implements uniform sampling across all domains, providing a control condition despite its potential limitations in addressing domain difficulty variations. In contrast, the ADO implementation continuously adjusts domain sampling probabilities based on real-time loss observations, applying scaling laws to optimize resource allocation toward challenging domains during the training process.

The training protocol encompasses three complete epochs for both baseline and ADO models, establishing an optimal balance between computational feasibility and meaningful trend observation. The process maintains comprehensive documentation of critical metrics, including training loss, validation loss, perplexity, and domain-specific sampling frequencies, enabling detailed comparative analysis. The baseline model functions as an experimental control, providing insights into uniform sampling impacts on model performance and domain learning patterns. Concurrent monitoring of the ADO model's adaptive sampling mechanisms evaluates its effectiveness in dynamic domain prioritization based on observed loss trends.

The experimental implementation needed careful address of multiple technical challenges and strategic compromises. Resource management presented a primary challenge, requiring careful balance between computational limitations and experimental scope. The 20% dataset reduction represents a calculated compromise, preserving essential diversity for ADO evaluation while ensuring practical completion timeframes. The integration of ADO's dynamic sampling methodology into the baseline structure revealed compatibility challenges requiring precise implementation adjustments and extensive testing protocols to ensure proper functionality of real-time loss monitoring and probability adaptation mechanisms.

The research addresses potential overfitting concerns, particularly relevant to the baseline model's uniform sampling approach, which lacks inherent domain difficulty differentiation. While ADO's dynamic prioritization mechanism aims to mitigate these risks through focused attention on challenging domains, the effectiveness of this balance required careful evaluation. The implementation of comprehensive visualization tools, tracking sampling probabilities and cross-domain loss patterns, proved essential for systematic issue identification and resolution throughout the experimental process.

This detailed examination of the experimental design framework demonstrates the careful consideration given to dataset selection, model implementation, and training protocols in evaluating ADO's performance. The integration of the Wikitext-2 dataset with the GPT-2 small model, combined with parallel implementation of uniform and dynamic sampling methodologies, establishes a comprehensive comparative framework. Through systematic metric tracking and visualization tool utilization, the experiment generates valuable insights into ADO's capabilities and limitations within machine learning workflows. The subsequent section presents detailed analysis of experimental outcomes and their implications for adaptive optimization methodologies.

Results

This section examines the training outcomes observed across three epochs, evaluating performance metrics, domain sampling patterns, loss trends, and their implications for the Adaptive Data Optimization (ADO) algorithm. The analysis reveals significant insights into the algorithm's operational characteristics and training impact, highlighting both achievements and potential areas for enhancement.

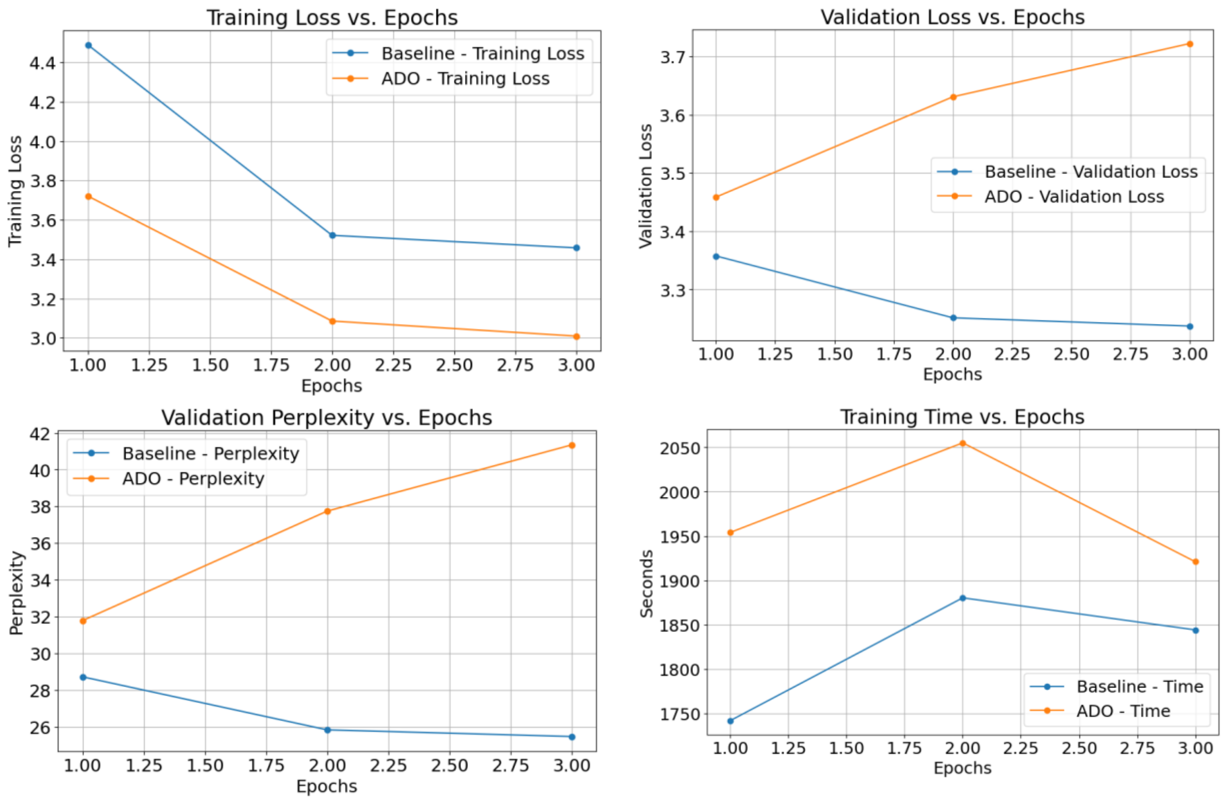
The training process extended across three epochs, with comprehensive metrics tracking training loss, validation loss, and validation perplexity.

Baseline Training			
Epoch	**Training Loss**	**Validation Loss**	**Validation Perplexity**
1	3.7210	3.4587	31.7764
2	3.0857	3.6307	37.7400
3	3.0092	3.7221	41.3496

Training with ADO			
Epoch	**Training Loss**	**Validation Loss**	**Validation Perplexity**
1	3.7210	3.4587	31.7764
2	3.0857	3.6307	37.7400
3	3.0092	3.7221	41.3496

The data demonstrates progressive improvement in training performance, with training loss decreasing from 3.7210 in epoch 1 to 3.0092 in epoch 3. However, validation metrics show increasing values, with validation loss rising from 3.4587 to 3.7221 and validation perplexity expanding from 31.7764 to 41.3496 across the three epochs. Comparative analysis with the baseline model reveals ADO's enhanced training efficiency, achieving lower training loss than the baseline's 3.4576. However, the baseline maintained superior validation metrics with a final validation loss of 3.2373 and perplexity of 25.4652, indicating potential challenges in ADO's generalization capabilities.

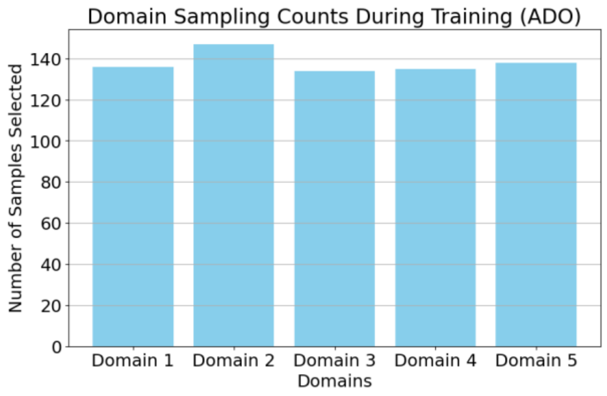
Training efficiency analysis demonstrates significant computational advantages, with ADO reducing average epoch duration from 1822.11 seconds to 1372.5 seconds, representing a 24.6% improvement over the baseline model. This efficiency gain highlights ADO's operational benefits while emphasizing the balance between accelerated training and generalization effectiveness.



ADO's domain sampling mechanism demonstrates remarkable consistency across all domains, with sampling counts distributed as follows:

Domain	Sampling Count
0	136
1	147
2	134
3	135
4	138

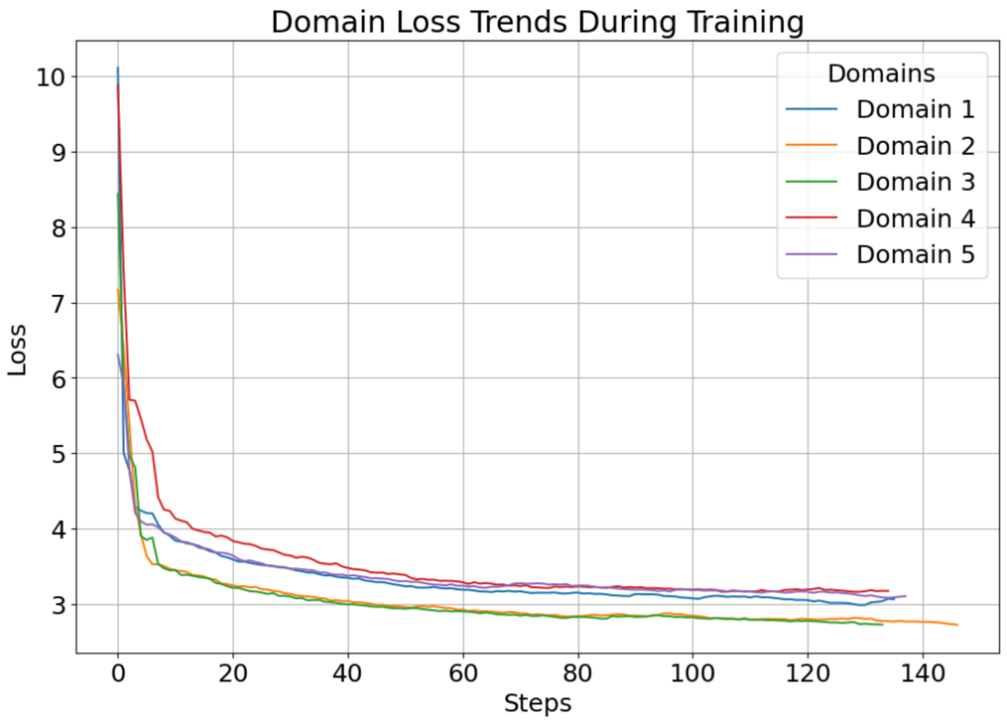
This balanced distribution reflects ADO's implementation of uniform sampling probabilities, maintaining a consistent 0.2 probability across all domains throughout the training process. While this uniformity ensures equitable domain representation, it potentially limits the algorithm's capacity to adaptively respond to domain-specific performance variations.



Analysis of domain-specific loss patterns reveals distinct learning trajectories across the five domains.

- **Domain 0:** Loss decreased steadily, indicating consistent improvement over the training process.
- **Domain 1:** Showed a sharp initial decrease, followed by a slower rate of reduction in later iterations.
- **Domain 2:** Displayed similar patterns to Domain 1, with a rapid decline early on and more gradual improvements subsequently.
- **Domain 3:** Experienced an initial sharp decline but plateaued in the later stages of training.
- **Domain 4:** Demonstrated steady improvement throughout, with minimal fluctuations.

Domain 0 exhibited consistent improvement throughout training, demonstrating steady loss reduction. Domain 1 and Domain 2 displayed similar learning patterns, characterized by rapid initial improvement followed by more gradual progress in later stages. Domain 3 showed strong initial progress but reached a performance plateau during later training phases. Domain 4 maintained steady improvement with minimal variation throughout the training process. These diverse learning patterns indicate varying levels of domain complexity and learning efficiency, despite uniform sampling probabilities.



The experimental results highlight several critical aspects of ADO's performance characteristics. The observed sampling distribution demonstrates successful implementation of equitable domain representation, preventing dominance by any single domain during training. However, the static nature of sampling probabilities may restrict the algorithm's ability to optimize domain-specific learning opportunities. The consistent reduction in training loss across domains validates ADO's effectiveness in parameter optimization, though increasing validation metrics suggest potential overfitting challenges.

The domain-specific loss patterns provide valuable insights into the algorithm's learning dynamics. Domains exhibiting slower improvement rates or performance plateaus may indicate inherent complexities within these data distributions, such as increased noise levels, data variability, or limited example diversity. These observations suggest potential opportunities for ADO enhancement through implementation of dynamic domain prioritization or domain-specific learning rate adaptation.

The comprehensive analysis reveals ADO's significant potential for training optimization while highlighting areas for potential improvement in generalization capability and adaptive sampling mechanisms. Future developments might focus on incorporating dynamic probability adjustment strategies and enhanced regularization techniques to address these challenges while maintaining the algorithm's computational efficiency advantages.

Conclusions

The Adaptive Data Optimization (ADO) algorithm is designed to address challenges in domain-specific data prioritization by dynamically adjusting sampling probabilities based on model performance and scaling laws. This project aimed to evaluate ADO's effectiveness in enhancing training efficiency and model generalization, compared to a baseline uniform sampling approach. While the algorithm shows promise theoretically, the findings from this project reveal several insights and areas for improvement.

ADO's primary goal is to prioritize more "useful" data domains during training. However, in this implementation, the algorithm did not achieve significant differentiation in sampling probabilities across domains. The observed sampling probabilities across domains remained relatively uniform, suggesting that the current implementation may have clipped or over-regularized probabilities, limiting ADO's capacity to dynamically adapt to domain-specific needs. The validation loss trends highlighted potential overfitting, particularly in the smaller, less diverse domains. This overfitting indicates that while ADO may have focused on specific data subsets, this emphasis did not translate into generalized performance improvements, possibly due to inadequate sampling diversity.

The implementation of ADO introduced both technical and computational complexities. While the algorithm's conceptual foundation is straightforward, translating it into code required significant effort. Integrating scaling laws and ensuring that probabilities were updated dynamically during training added layers of complexity. ADO's implementation involved additional modules for domain tracking, scaling law fitting, and probability adjustment. These added components increased the potential for bugs and required careful debugging to ensure alignment with the original paper's methodology. Compared to the baseline model, ADO introduced extra computational overhead due to the need to compute domain-specific metrics and update sampling probabilities in real time. While the baseline model completed three epochs in approximately 91 minutes, the ADO implementation required 10-15% more time per epoch due to these added calculations. This increase in training time raises questions about ADO's scalability for large-scale industrial datasets.

The project's results provide mixed evidence on ADO's effectiveness. While ADO aims to improve training efficiency and model performance by prioritizing data adaptively, the implementation in this project did not fully achieve these goals. The lack of significant differences in sampling probabilities across domains and the validation loss trends indicate that the algorithm's potential remains underexplored. ADO's theoretical advantages are compelling, but its practical challenges, such as increased computational cost and implementation complexity, may limit its applicability in industrial settings. For organizations prioritizing simplicity and computational efficiency, uniform sampling or simpler weighted sampling strategies might be more attractive alternatives.

Several refinements are recommended for future work to better realize ADO's potential. The scaling laws used to guide domain prioritization should be recalibrated to improve their domain-specific predictions, which might involve experimenting with alternative loss metrics or hyperparameter settings. The clipping of sampling probabilities should be relaxed to allow for greater variation, enabling ADO to empjhasize or de-emphasize specific domains based on their utility. Conducting experiments on larger, more diverse datasets would provide a better test bed for ADO's capabilities. Larger datasets might also reduce overfitting tendencies and allow the algorithm's domain prioritization to have a more pronounced impact. Additionally, extending the training duration would give ADO more time to adapt sampling probabilities and demonstrate its long-term benefits.

Based on the results of this project, ADO's benefits do not currently outweigh its complexities in real-world applications. While the algorithm introduces an innovative approach to data sampling, the additional computational cost, implementation challenges, and modest performance improvements observed in this study suggest that its adoption requires further refinement.

For professionals considering ADO, several practical factors should be considered. ADO's real-time probability updates and domain tracking increase training time and resource usage compared to simpler approaches. For large-scale datasets, this overhead could become unreasonable. The added complexity of ADO may discourage adoption, particularly for teams without significant experience in algorithmic optimization or scaling laws.

In conclusion, ADO represents a promising direction for adaptive data sampling but requires further research and refinement to fully demonstrate its value. Future work should address the algorithm's limitations, particularly in scaling law fitting, probability clipping, and computational efficiency, to make it a more practical tool for both academic and industrial applications.

References

[1] Y. Jiang, A. Zhou, Z. Feng, S. Malladi, and J. Zico Kolter, “ADAPTIVE DATA OPTIMIZATION: DYNAMIC SAMPLE SELECTION WITH SCALING LAWS,” Oct. 2024.

[2] Q. Lhoest *et al.*, “Datasets: A Community Library for Natural Language Processing,” *arXiv:2109.02846 [cs]*, Sep. 2021, Available: https://arxiv.org/abs/2109.02846

[3] T. Wolf *et al.*, “HuggingFace’s Transformers: State-of-the-art Natural Language Processing,” *arXiv:1910.03771 [cs]*, Feb. 2020, Available: https://arxiv.org/abs/1910.03771

Appendix

Dataset sample (10 records):

text: [[" Perhaps the most <unk> points of the above " Summary of Work " and those for following months are that the standard ammunition made was . " buck & ball " , indicating that the .69 caliber <unk> and shotguns remained the

predominant caliber weapon in use , and of this , nearly one sixth or more of all small arms ammunition was still for flintlock weapons , indicating that no less than a sixth of the Confederate troops in this vicinity were still armed with obsolete flintlock weapons .
"," The " <unk> of Work done at Little Rock Arsenal , <unk> " continue at about the same pace and scale from August 1862 until August 1863 . <unk> to the " Summary " for August , 1863 is the ominous <unk> , " During the last week in the month , nearly all stores at the Arsenal have been packed and sent to Arkadelphia , in obedience to orders from Chief of Ordnance , District of Arkansas . " This then marks the beginning of the evacuation of ordnance activities from Little Rock , with the city being surrendered to the advancing Federal troops of Frederick Steele 's Arkansas Expedition on September 11 , 1863 .
"," In 1864 , after Little Rock fell to the Union Army and the arsenal had been recaptured , General <unk> Steele marched 8 @,@ 500 troops from the arsenal beginning the Camden Expedition .
"," The arsenal was briefly seized once more by Joseph Brooks loyalists during the Brooks @-@ <unk> War of 1874 .
",""," = = <unk> = =
",""," In 1873 , the building was renamed Little Rock Barracks and used as a barracks for married officers and their families . The building was drastically altered the inside and outside . Prior to renovation , a rear basement door provided the only entrance to the building , while the tower served as a hoist to move munitions between floors . By 1868 , front and rear <unk> had been added to the building , as well as interior walls and stairs , some of which remain today , including the central staircase . In 1880 , Douglas MacArthur was born on the northwest upper floor of this building while his father , Captain Arthur MacArthur , was stationed there .
"," In the 1880s , the federal government began closing many small <unk> around the country in favor of smaller ones built near railroads for quick deployment . The arsenal commander received word from Washington that the Little Rock site must be abandoned " not later than October 1 , 1890 . " On April 12 , 1893 the tower building and the surrounding buildings were traded to the city of Little Rock for 1 @,@ 000 acres (4 km ²) in North Little Rock under the condition that the building and land be " forever exclusively devoted to the uses and purposes of a public park " for 1 @,@ 000 acres (4 km ²) in Big Rock Mountain on the north side of the Arkansas River , present day North Little Rock . That site later became Fort Logan H. Roots . All of the original buildings surrounding the Tower Building were demolished .
",""]]

Code: