DOI: 10.37943/22OXKY5402

Bauyrzhan Kairatuly
PhD Student, Faculty of Information Technology
bauyrjanesq@gmail.com, orcid.org/0000-0002-0341-0588
Farabi University, Kazakhstan

Aday Shomanov
PhD, Instructor, School of Engineering and Digital Sciences
adai.shoimanov@nu.edu.kz, orcid.org/0000-0001-8253-7474
Nazarbayev University, Kazakhstan

BALANCING SPEED AND PERFORMANCE WITH LAYER FREEZING STRATEGIES FOR TRANSFORMER MODELS

Abstract: In this paper, we evaluated different approaches to freezing BERT-base layers and analyzed their impact on the quality and speed of training in the task of named entity recognition in two languages. Layer freezing is an optimization technique in deep neural network training in which specific layers of a model remain fixed. This means their weights do not change during the backpropagation process. By not updating these layers, the overall number of parameters requiring adjustment is reduced, which results in lower computational demands and faster training times. Partial freezing of layers proved to be an effective way to preserve key representations of the model and ensure its adaptation to new tasks. Experimental results showed that freezing from three to six layers allows to achieve stable model performance regardless of the training language. Unlike standard approaches, our method highlights cross-linguistic applicability and promotes energy-efficient training. We personally designed the experimental setup, implemented the freezing scenarios, and carried out all performance evaluations. This study aims to evaluate the effectiveness of layer freezing in a pre-trained BERT model when performing the named entity recognition task. Two variants of the freezing strategy are considered: in the first one the upper layers of the model are fixed, in the second one the lower layers remain unchanged. The analysis is based on two corpora, the English language CoNLL 2003 and the Kazakh language KazNERD. Our experiments showed that freezing three to six layers provides the best balance between training speed and model quality. On the CoNLL-2003 dataset, the training time decreased from 266 to 167 seconds and the Macro F1 score remained at 87%. On KazNERD, learning accelerated from 1609 to 958 seconds with an accuracy of 94-95% and Macro F1 in the range of 71-72%. Full freezing of all 12 layers caused a dramatic drop in quality, with Macro F1 dropping to 50% on CoNLL and to 7% on KazNERD. This emphasises the importance of limited freezing and fine-tuning of the model architecture.

The study further examines how the choice of layers to freeze influences the model's ability to adapt to new linguistic patterns and domain-specific terminology. These findings offer useful insights for researchers and practitioners aiming to enhance the efficiency of fine-tuning large language models while ensuring robust performance across different languages and datasets. The results also highlight the potential for optimizing resource usage in various NER applications without compromising critical language understanding.

Keywords: layer freezing; BERT; NER; English; Kazakh

Copyright © 2025, Authors. This is an open access article under the Creative Commons CC BY-NC-ND license
Received: 13.03.2025 Accepted: 26.06.2025 Published: 30.06.2025

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

Introduction

Modern language models are among the greatest achievements in science [1], and although debates continue as to whether we have truly approached the creation of strong AI, the role and effectiveness of large language models in developing highly intelligent information systems cannot be overstated. However, the cost of achieving such effectiveness is very high, both in terms of computational resources and human effort. For example, creating the BERT model required a significant number of resources: BERT-base contains 110 million parameters, while BERT-large contains 340 million parameters [2], [3]. Although precise data on the costs of training BERT are limited, it is known that extensive computational power and enormous volumes of textual data were used in its training.

In the case of GPT-3, the cost was even higher. The model contains 175 billion parameters, was trained on 570 GB of textual data (including Common Crawl, WebText2, books, and Wikipedia), and was trained on the Microsoft Azure AI supercomputer, which consists of 285,000 CPU cores and 10,000 enterprise-level GPUs [4]. It is estimated that training on a single GPU would have taken 355 years, but thanks to parallel computing, the process was significantly accelerated. The financial cost of training GPT-3 is estimated at 4.6 million US dollars [5].

For the Kazakh language model Llama-3.1-Sherkala-8B-Chat, all training, hyperparameter tuning, and instruction-tuning were conducted on the Condor Galaxy 2 (CG-2) AI supercomputer developed by Cerebras and G42. The final training and fine-tuning runs were performed on 16 Cerebras CS-2 systems, each delivering up to 62.5 AI petaFLOPs. The system relied on MemoryX for storing model states and SwarmX for interconnect and gradient aggregation, using weight streaming to enable efficient data parallelism across CS-2 units. Given the scale of the infrastructure and model size, the training process was undoubtedly computationally intensive and costly[6].

Some developers of large language models prefer not to disclose the costs and resources used to run their systems. These resources include not only financial investments, but also environmental indicators, including carbon footprint and water consumption. This increases interest in developing language models that learn faster, require less data, and cost fewer resources while maintaining a comparable level of quality.

The present work considers one of the methods to reduce the training time, namely layer freezing [7], [8]. Layer freezing is an optimization method in training deep neural networks, where certain layers of the model are fixed (their weights remain unchanged) and do not participate in the backpropagation process [9], [10]. In work [11], the authors discovered an interesting fact: freezing the initial 50% of Transformer layers not only shortens training time but also unexpectedly improves Macro F1 (by up to 8%) compared to fully trainable layers in a few-shot learning setting. The authors of work [12] explore speeding up BERT training on large distributed systems and propose two key improvements: local pre-sorting of data using stratification to balance the GPU load, and gradient clipping at the "bucket" level before allreduce, which allows combining computation and data transmission. Experiments on 1024 NVIDIA A100 GPUs showed that the proposed methods reduce training time to 25.1 seconds (22.3 seconds in MLPerf v2.0), making their solution faster than existing alternatives. Local pre-sorting reduced balancing overhead from 18.2% to 1.9%, while the new gradient clipping method enhanced training efficiency. The proposed approaches may be useful for other large-scale models, although the effectiveness of load balancing is limited with small datasets. The study ultimately set a new record for BERT training speed in MLPerf.

Early stopping is another novel approach that, alongside quantization and layer freezing, reduces energy consumption without sacrificing model quality. The authors of work [13] investigate methods for enhancing the energy efficiency of neural network training by analyzing three techniques: freezing layers during training, model quantization, and early stopping. They

DOI: 10.37943/22OXKY5402
© Bauyrzhan Kairatuly, Aday Shomanov

emphasize that modern machine learning models require substantial computational resources, which leads to high energy consumption and an increased carbon footprint. A unique methodology was developed that includes monitoring energy usage and predicting the impact of various strategies on model accuracy and energy costs. Using convolutional neural networks and 12 datasets, the researchers applied different training strategies and built an autoregressive model to forecast both accuracy and energy consumption. The results indicate that the proposed approach can reduce energy consumption by 56.5% while simultaneously increasing accuracy by 2.38% through the prevention of overfitting. The predictive algorithm developed in the study is capable of forecasting changes in model accuracy with 91.6% precision and estimating energy consumption with 85.7% accuracy. Ultimately, the study proposes an adaptive training management methodology that allows for the real-time determination of optimal strategies to balance energy consumption and accuracy.

The main contribution of this paper is to design and conduct a series of comparative experiments on BERT layer freezing based on two corpora for the task of named entity recognition. Unlike most previous studies focusing on models trained for a single language or under conditions of abundant resources, we first tested the effectiveness of the approach on Kazakh language which is morphologically complex and low-resource.

The Methods and Materials section outlines an approach to selective layer freezing in a multilingual BERT model for the task of named entity recognition without using a pre-trained base. The results section presents performance metrics for different freezing strategies on the English-language CoNLL-2003 [14] and Kazakh-language KAZNERD [15] datasets.

Methods and Materials

Training of transformer models, including BERT, is computationally intensive. At the same time, complete updating of all parameters is not always necessary to achieve acceptable results in applied tasks. To reduce the computational load, it is possible to fix certain layers of the model, excluding them from the parameter update process. Layer freezing is an effective technique for optimizing neural network training, as it reduces computational costs, preserves pre-learned features, and speeds up the training process. Since some model parameters remain unchanged, training requires fewer resources, and the lower layers, which have been trained on large volumes of data, continue to retain useful text characteristics, ensuring stability in the representations. With fixed layers, fewer parameters are updated, which shortens the time required for each epoch and allows the model to adapt more quickly to new data, particularly under limited computational conditions. Layer freezing is especially relevant when fine-tuning a model on specific domains, such as in Named Entity Recognition (NER) tasks.

1. Model description

For the experiment, the BERT-base-uncased model [16] was used, enhanced with an additional token classification head connected via the BertForTokenClassification [17] class from the Hugging Face Transformers library, and trained on the CoNLL-2003 and KazNERD datasets. BERT is a deep learning model that, thanks to its bidirectional architecture and self-attention mechanism, can create context-dependent embeddings for each word. This architecture provides high performance in various natural language processing tasks, especially after adaptation on target datasets. The bert-base-uncased model is a basic implementation of BERT including 12 transformer blocks. The classification head is a linear layer that takes tokens of dimensionality 768 as input and transforms them into output labels according to the number of classes specified via the num_labels parameter.

2. Dataset description

As mentioned earlier, two datasets are used in this study: the English language subset of CoNLL 2003 and the Kazakh corpus KazNERD. CoNLL 2003 is one of the best known resources

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

for named entity recognition tasks. It includes texts in English, German and Dutch. The annotated data is based on Reuters news stories and is labelled into four categories: persons (PER), organisations (ORG), places (LOC) and other entities (MISC). The corpus is suitable for model training and evaluation. Due to its popularity and use in competitions such as the CoNLL Shared Task, it remains one of the standard benchmarks for evaluating model performance in named entity extraction tasks. The data include annotations by classes:

- B-LOC, I-LOC – geographical objects
- B-MISC, I-MISC – miscellaneous
- B-ORG, I-ORG – organizations
- B-PER, I-PER – persons
- O – tokens without named entities

KazNERD is the largest publicly available corpus of named entities for the Kazakh language, containing 136,333 annotated entities. The main classes include CARDINAL (29,260), DATE (25,446), and GPE (17,543). These categories constitute a significant part of the corpus, which is due to the nature of the news texts used in its creation. The structure of KazNERD is divided into three parts:

- Training set – 90,228 sentences (80%)
- Validation set – 11,167 sentences (10%)
- Test set – 11,307 sentences (10%)

Annotation was performed manually by two native speakers in accordance with the developed guidelines. The data are annotated with 25 categories of named entities, including geographical objects (GPE, LOCATION), persons (PERSON), organizations (ORG), numerical values (CARDINAL, ORDINAL, MONEY, PERCENTAGE), and others. The presence of such classes as ADAGE (Kazakh proverbs) and NON_HUMAN (animal names) makes KazNERD a valuable and atypical resource for the task of extracting named entities in Kazakh.

The dataset is hosted in an open repository, which ensures reproducibility and reusability of the results obtained. In accordance with ethical standards, the corpus does not contain personal information and does not require additional authorisations.

3. Description of Freezing Scenarios

Layer freezing was performed using different scenarios: bottom-up (freezing the first n layers) and top-down (freezing the last n layers), where n was chosen as 0, 3, 6, 9, or 12. Thus, a total of 10 scenarios were applied during the fine-tuning of the model on each dataset (2 without freezing for the baseline and 8 with freezing).

We manually configured each freezing scenario using the Hugging Face Transformers API, implemented gradient disabling per layer, and ensured reproducibility by scripting the full training pipeline. Additionally, the analysis of how freezing affects macro F1 on rare categories in KazNERD (e.g., ADAGE, NON_HUMAN) reflects our active role in dataset interpretation and linguistic error analysis.

First, the parameters of the embedding layer converting the input text into vector representation were fixed by disabling their updating. Then, the calculation of gradients for the first n layers of the encoder was switched off, which resulted in their freezing. This method allowed us to reduce the number of trained parameters, preserve the previously formed representations and direct the training to those parts of the model that are responsible for adapting to the task of recognising named entities.

It is important to note that while layer freezing is the main objective of these experiments, freezing the embeddings is employed to preserve pre-optimized representations extracted from a large volume of data. Embedding layers play a key role in forming the vector representation of the input text, and their pre-training effectively encodes general linguistic properties. When adapting the model to a specific task, changing these parameters can lead to a loss of already

DOI: 10.37943/22OXKY5402
© Bauyrzhan Kairatuly, Aday Shomanov

acquired information and a deterioration of generalization capabilities, especially when the amount of data is limited. Freezing the embeddings ensures the stability of the initial stage of transforming input data, allowing the training process to concentrate on adapting the higher-level components of the model to the new task.

4. Layer Freezing in Transformer Fine-Tuning

Transformer architectures such as BERT have been shown to be highly effective for natural language processing tasks. The BERT model includes $\mathcal{L} = 12$ layers of transformer encoder and has about 110 million parameters [18]. In the process of pre-training on a particular task, a strategy of freezing part of the layers, i.e., disabling their updates during training, is often used. This approach helps to reduce overtraining and improve computational performance.

Let $f(x;\theta)$ represent the output of the BERT model given input x, where the parameter set $\theta = (\theta_1, \theta_2, \ldots, \theta_L)$ are the parameters of each layer. We divide the parameters into frozen and trainable subsets:

$$\theta = \theta_{frozen} \cup \theta_{trainable}, \nabla_{\theta_{frozen}} \mathcal{L} = 0 \tag{1}$$

Given a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, we minimize a loss function $\mathcal{L}$, such as cross-entropy for classification:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log \hat{y}_{i,k}, \hat{y}_i = softmax\big(f(x_i;\theta)\big) \tag{2}$$

When layers are frozen, the optimisation space is narrowed by reducing the number of parameters to be trained $|\theta_{trainable}|$. This leads to the formulation of a constrained problem in which optimisation is carried out for only a fraction of the model parameters:

$$\min_{\theta_{\text{trainable}}} \mathcal{L}(\theta_{\text{frozen}}, \theta_{\text{trainable}}) \tag{3}$$

From a regularisation perspective, this strategy represents a kind of implicit assumption that the frozen layers already contain valuable representations formed during pre-training. As a result, the Jacobian matrix $J = \frac{\partial f}{\partial \theta}$ becomes more sparse, which reduces the variability of the gradients and may contribute to the better generalisation ability of the model:

$$\text{Var}[\nabla_\theta \mathcal{L}] \propto \sum_{l \in \text{trainable}} \left| \frac{\partial \mathcal{L}}{\partial \theta_l} \right|^2 \tag{4}$$

Empirical studies often compare different freezing strategies:

- Bottom-up freezing: freeze layers 1 to k (lower layers).
- Top-down freezing: freeze layers k to L (higher layers).
- Interleaved: freeze alternating layers or selected modules like attention heads.

Let T_{full} and T_{frozen} denote training time per epoch for fully fine-tuned vs. partially frozen models. Then:

$$T_{frozen} \approx T_{full} \times \left(\frac{|\theta_{trainable}|}{|\theta|} \right) \tag{5}$$

This approach provides computational savings, making it particularly beneficial for edge deployment, where resources are limited.

In practice, the optimal number of layers to freeze, denoted as k^*, can be determined through cross-validation by maximizing the F1 score on a validation set:

$$k^* = \arg\max_k \text{F1}_{\text{val}}(k), \quad k \in \{0, \ldots, L\} \tag{6}$$

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

Freezing layers during fine-tuning thus reduces the risk of overtraining, shortens the training time and optimises the use of computational resources. This approach is especially relevant when data is limited, or infrastructure is weak and is considered one of the key methods of modern transformer tuning.

Results

Layer freezing significantly reduces training time, especially when freezing the upper layers. An optimal balance between quality and speed is achieved by freezing 3–6 layers. With too many frozen layers (9 or more), the model loses its ability to adapt well to the task, particularly for rare classes. While freezing is effective for accelerating training, it requires careful tuning to maintain high quality. This experiment demonstrates that layer freezing can be a useful tool for model optimization, especially under limited computational resources.

Table 1. Bottom-Up Freezing on the CoNLL-2003 Dataset

Layers Frozen	Training Loss	Training Time (s)	Accuracy	Macro F1	Weighted F1
0	0,0097	266,24	97%	87%	97%
3	0,0031	198,1	97%	87%	97%
6	0,0056	167,24	97%	87%	97%
9	0,0268	139,09	97%	86%	97%
12	0,3568	108,82	91%	50%	90%

The table demonstrates the effect of gradually freezing the bottom layers of a bottom-up model on learning on the CoNLL 2003 dataset. Up to nine frozen layers, the performance remains stable. However, when all twelve layers are frozen, there is a significant decrease in the Macro F1 value, indicating a loss in the quality of the representations.

Table 2. Top-Down Freezing on the CoNLL-2003 Dataset

Layers Frozen	Training Loss	Training Time (s)	Accuracy	Macro F1	Weighted F1
0	0,0097	212.91	97%	85%	97%
3	0,0031	200,1	97%	85%	97%
6	0,0056	170,27	97%	85%	97%
9	0,0268	141,9	97%	86%	97%
12	0,3568	108,21	91%	50%	90%

This table shows results for freezing layers from top to bottom on the same dataset. Similar to bottom-up freezing, the model retains performance until 9 layers are frozen, after which a substantial drop in Macro F1 and overall accuracy is observed.

Table 3. Top-Down Freezing on the KazNERD Dataset

Layers Frozen	Training Time (s)	Training Loss	Accuracy	Macro F1	Weighted F1
0	1404,85	0,0097	96%	72%	95%
3	1274,96	0,0031	95%	71%	95%
6	1150,65	0,0056	95%	69%	95%
9	1044,55	0,0268	94%	57%	94%
12	648,19	0,3568	82%	7%	75%

DOI: 10.37943/22OXKY5402
© Bauyrzhan Kairatuly, Aday Shomanov

Here, top-down freezing is applied to a Kazakh NER dataset. The results mirror those from CoNLL-2003: model performance degrades steadily with more layers frozen, with a drastic drop in Macro F1 when all 12 layers are frozen, highlighting the need for upper layers in learning task-specific features.

Table 4. Bottom-Up Freezing on the KazNERD Dataset

Layers Frozen	Training Time (s)	Training Loss	Accuracy	Macro F1	Weighted F1
0	1609,23	0,0097	95%	72%	95%
3	1146,17	0,0031	95%	72%	94%
6	958,48	0,0056	94%	71%	94%
9	778,56	0,0268	93%	65%	93%
12	605,17	0,3568	79%	7%	71%

This table presents the effects of bottom-up freezing on the KazNERD dataset. The model maintains strong performance up to 9 frozen layers, but completely freezing the encoder severely harms both accuracy and F1 scores, especially Macro F1, similar to top-down freezing.

Discussion

We compared different layer freezing strategies for the BERT-base-uncased model when addressing the Named Entity Recognition (NER) task on two datasets: CoNLL-2003 for English and KazNERD for Kazakh. The primary objective was to reduce training time by freezing the embedding layer and the first few layers of the model, thereby decreasing the number of trainable parameters while preserving general language representations learned during large-scale pre-training.

Our results confirmed that freezing the lower layers significantly reduces training time by approximately 37% on CoNLL-2003 and by more than 40% on KazNERD while maintaining model quality. In the 3–6-layer freezing range, the model consistently achieved high accuracy (94-97%) and strong Macro F1 scores (71-87%). The results confirm that the lower layers of the BERT model retain general linguistic features, including information about parts of speech and syntactic structures. These representations remain useful in a variety of tasks and subject areas [19], [20].

A dramatic drop in performance and stability was observed when more than 9 layers were frozen. Complete freezing of all 12 layers caused Macro F1 to drop to 50% on CoNLL 2003 and to 7% on KazNERD. This indicates the importance of the upper layers in forming the semantic representations needed to account for context and task specificity. Such findings are consistent with previous studies showing that the senior layers of transformers encode information that is sensitive to the target task [21], [22], and excessive freezing limits the model's ability to generalise.

To ensure reproducibility, all experiments used a single base bert-base-uncased architecture from the Hugging Face Transformers library version 4.36. Fine-tuning was performed using the AdamW optimiser with a learning rate of 3e-5, a batch size of 32 and an early stopping mechanism based on the Macro F1 value on the validation sample. Each freezing configuration (0, 3, 6, 9, and 12 layers) was evaluated with both top-down and bottom-up strategies. For freezing, we disabled gradient updates by setting requires_grad=False for the corresponding layers and froze the embedding layer in all runs.

All experiments were run on a single NVIDIA A100 graphics card with 40GB of memory. Each run was repeated three times to account for variance, and the average values were reported. The data was divided into training, validation, and test samples in the proportion of 80

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

by 10 by 10. A common preprocessing scheme was used in all experiments, including lower case text reduction, WordPiece tokenisation, and complementation or truncation of sequences up to 128 tokens.

Although the idea of layer freezing is not a new one, this paper critically evaluates its effectiveness in multilingual settings, with a focus on the differences between languages. It is found that for English, it is possible to freeze more layers without significant quality degradation, whereas for Kazakh, performance starts to degrade earlier due to its morphological complexity. These observations underline the need to take into account language specificity when choosing optimisation strategies, which has not been given enough attention in the past. Moreover, the results extend previous findings (e.g., [7], [11]) by confirming them in a new, low-resource setting. Compared to AutoFreeze [9] or SmartFRZ [10], which automate freezing via heuristics, our manual scenario-based design offers transparent insights into each layer's utility.

Despite the encouraging results, the study has several limitations. Firstly, it covers only two datasets, CoNLL 2003 and KazNERD. This limits the ability to generalise the findings to other languages, topic areas and tasks not related to named entity recognition. Secondly, all experiments were performed exclusively using the BERT model without a pre-trained base. Other architectures such as RoBERTa, DeBERTa and multilingual models may show a different response to layer freezing due to differences in pre-training procedures and architectural features. Third, although a reduction in training time is observed, no quantification of energy consumption or carbon footprint is presented in this work. This aspect is particularly important when computational resources are limited. Finally, the impact of tokenisation in morphologically rich languages such as Kazakh has not been considered separately, despite the fact that this factor can significantly affect the quality of representations and the final performance of the model. Lastly, all evaluations were limited to a maximum sequence length of 128 tokens, which may not reflect performance on longer or more complex documents. In future work, we plan to examine selective fine-tuning techniques that dynamically unfreeze layers during training based on validation performance, as well as investigate the impact of freezing strategies in cross-lingual transfer settings where the source and target languages differ in morphology and syntax.

Conclusion

Our research has demonstrated that partially freezing the layers of the BERT-base-uncased model considerably speeds up the fine-tuning process for Named Entity Recognition tasks while still achieving high accuracy and F1 scores. Our experiments on the CoNLL-2003 and KazNERD datasets reveal that fixing the embedding layer along with the first 3–6 layers preserves the core linguistic representations learned during pre-training, enabling the upper layers to efficiently adjust to the nuances of the new task.

Numerical results confirm the effectiveness of this strategy. On the CoNLL-2003 dataset, freezing 3 to 6 layers reduced training time by up to 37% (from 266s to 167s) in bottom-up freezing and maintained high accuracy (97%) and Macro F1 (87%). On the KazNERD dataset, similar freezing reduced training time from 1609s to 958s and preserved accuracy at 94–95% with Macro F1 scores of 71–72%. In both datasets, freezing all 12 layers caused a steep performance drop (Macro F1 dropping to 50% for CoNLL and to just 7% for KazNERD), confirming that a balance must be struck between efficiency and representation learning.

Importantly, we observed similar outcomes in both the high-resource English and low-resource Kazakh scenarios, underscoring the significant role played by BERT's architecture in this process. This consistency supports the potential for developing "lightweight" yet effective

models. Ultimately, our findings lay the groundwork for future research into fine-tuning optimization techniques that can reduce energy consumption and resource demands without substantially sacrificing model performance.

Acknowledgement

This research is funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan under the program for grant financing of young scientists for scientific and/or scientific-technical projects for the years 2024-2026 No AP22787410.

References

[1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

[2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805

[3] Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. Transactions of the Association for Computational Linguistics, 8, 842–866. https://doi.org/10.1162/tacl_a_00349

[4] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., ... Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165. https://arxiv.org/abs/2005.14165

[5] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. arXiv preprint arXiv:1906.02243. https://arxiv.org/abs/1906.02243

[6] Aitbayev, A., Makhambetov, A., Kali, A., Gabdullin, A., Assylbek, A., Askarbekuly, D., Nurgali, N., & Issakhov, A. (2024). Llama-3.1-Sherkala-8B-Chat: An open large language model for Kazakh (arXiv:2503.01493v1). arXiv. https://arxiv.org/html/2503.01493v1

[7] Lee, J., Tang, R., & Lin, J. (2019). What would Elsa do? Freezing layers during transformer fine-tuning. arXiv preprint arXiv:1911.03090. https://arxiv.org/abs/1911.03090

[8] Wang, Y., Sun, D., Chen, K., Lai, F., & Chowdhury, M. (2023). Egeria: Efficient DNN training with knowledge-guided layer freezing. Proceedings of the Eighteenth European Conference on Computer Systems, 851–866. https://doi.org/10.1145/3552326.3587572

[9] Liu, Y., Agarwal, S., & Venkataraman, S. (2021). AutoFreeze: Automatically freezing model blocks to accelerate fine-tuning. arXiv preprint arXiv:2102.01386. https://arxiv.org/abs/2102.01386

[10] Li, S., Yuan, G., Dai, Y., Zhang, Y., Wang, Y., & Tang, X. (2024). SmartFRZ: An efficient training framework using attention-based layer freezing. arXiv preprint arXiv:2401.16720. https://arxiv.org/abs/2401.16720

[11] Ingle, D., Tripathi, R., Kumar, A., Patel, K., & Vepa, J. (2022, December). Investigating the characteristics of a transformer in a few-shot setup: Does freezing layers in RoBERTa help?. In Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (pp. 238–248).

[12] Kim, Y., Ahn, J., Kim, M., Choi, C., Kim, H., Tuvshinjargal, N., Lee, S., Zhang, Y., Pei, Y., Linghu, X., Ma, J., Chen, L., Dai, Y., & Yoo, S. (2024). Breaking MLPerf Training: A Case Study on Optimizing BERT. arXiv preprint arXiv:2402.02447. https://arxiv.org/abs/2402.02447

[13] Reguero, Á.D., Martínez-Fernández, S., & Verdecchia, R. (2025). Energy-efficient neural network training through runtime layer freezing, model quantization, and early stopping. Computer Standards & Interfaces, 92, 103906. https://doi.org/10.1016/j.csi.2024.103906

[14] Sang, E.F.T.K., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proceedings of the Seventh Conference on Natural Lan-

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

guage Learning at HLT-NAACL 2003 (pp. 142–147). Association for Computational Linguistics. https://doi.org/10.3115/1119176.1119195

[15] Nurpeiissov, M., Mussakhojayeva, A., Makhambetov, Y., & Nurmukhanbet, K. (2023). KazNERD: A Kazakh named entity recognition dataset. arXiv preprint arXiv:2304.08179. https://arxiv.org/abs/2304.08179

[16] Goutam, K., Balasubramanian, S., Gera, D., & Sarma, R. R. (2024). LayerOut: Freezing layers in deep neural networks. SN Computer Science, 5(1), 123. https://doi.org/10.1007/s42979-023-01678-9

[17] Hugging Face. (n.d.). BERT. Hugging Face Transformers Documentation. https://huggingface.co/docs/transformers/en/modeldoc/bert

[18] Fukuhata, S., & Kano, Y. (2025). Few Dimensions are Enough: Fine-tuning BERT with Selected Dimensions Revealed Its Redundant Nature (arXiv:2504.04966) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2504.04966

[19] Miao, Z., & Zhao, M. (2023). Weight freezing: A regularization approach for fully connected layers with an application in EEG classification. arXiv preprint arXiv:2306.05775. https://arxiv.org/abs/2306.05775

[20] Sorrenti, A., Bellitto, G., Proietto Salanitri, F., Pennisi, M., Spampinato, C., & Palazzo, S. (2023). Selective freezing for efficient continual learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 1234–1243). https://doi.org/10.1109/ICCVW58398.2023.00156

[21] Shen, Z., Liu, Z., Qin, J., Savvides, M., & Cheng, K.-T. (2021). Partial is better than all: Revisiting fine-tuning strategy for few-shot learning. Proceedings of the AAAI Conference on Artificial Intelligence, 35(10), 9546–9553. https://doi.org/10.1609/aaai.v35i10.17055

[22] Isikdogan, L.F., Nayak, B.V., Wu, C.-T., Moreira, J.P., Rao, S., & Michael, G. (2020). SemifreddoNets: Partially frozen neural networks for efficient computer vision systems. arXiv preprint arXiv:2006.06888. https://arxiv.org/abs/2006.06888