Knowledge Transfer Between Language Models via a Vector-Based Approach

Mehmet Emin AYDIN
Yıldız Technical University, Computer Engineering
emin.aydin@std.yildiz.edu.tr

June 14 2024

Abstract

In the realm of machine learning, transferring knowledge between models trained on diverse corpora offers a promising way to enhance model performance and generalization. This project explores this by training an initial model, M0, on a corpus D0, and then transferring the learned knowledge to two additional models, M1 and M2, trained on larger corpora D1 and D2, respectively. M1 and M2 share identical architectures but differ in their weight values based on their training data.

Bayesian optimization is used to effectively navigate the search space of model configurations. Hyperparameters such as the number of steps and the selection method for generating prediction functions are tuned for optimal performance. The impact of using fixed versus resampled smaller datasets on optimization outcomes is examined, with initial sizes set to 1/100 of the original corpora.[3]

This project aims to balance the trade-off between increased sample size for accurate predictions and the computational cost of each optimization step, shedding light on the relationship between dataset size, noise reduction, and optimization efficiency. In summary, it highlights the potential of knowledge transfer in improving model performance and generalization through systematic experimentation and analysis.

Keywords: Knowledge transfer, Machine learning, Model training, Diverse corpora, Model performance, Transfer learning, Model optimization, Bayesian optimization, Hyperparameters, Dataset size, Computational cost, Experimentation, Analysis.

1 Introduction

In the realm of machine learning, the transfer of knowledge between models trained on different corpora offers a significant potential to enhance model performance and generalization. This project is designed to investigate the effectiveness of such knowledge transfer through a structured approach that involves the training and evaluation of models across multiple datasets. The process begins by training an initial model, M0, on a corpus D0. Subsequently, this M0 model is used as a foundational model to train two additional models, M1 and M2, on larger corpora D1 and D2, respectively. The primary goal is to leverage the knowledge learned by M0 and transfer it to M1 and M2, thereby creating two models that, while trained on different datasets, share a common foundational knowledge.[5] For this project, it is crucial to ensure that the sizes of corpora D1 and D2 are larger than D0. This is because if D1 and D2 are too small compared to D0, the practical benefits of knowledge transfer would be minimal, as the training on smaller datasets would not significantly build upon the knowledge from D0. Therefore, D1 and D2 are chosen to be larger than D0 to validate the practical significance of the transfer learning process. The architectures of M1 and M2 are identical, with their differences lying solely in the weight values that are adapted based on their respective training datasets. Starting from the initial weights derived from M0, the weights for M1 and M2 (denoted as w1 and w2, respectively) evolve based on the corpora D1 and D2. The aim is to identify the model configuration along the spectrum of weights between w1 and w2 that yields the highest overall performance across D1 and D2. Bayesian optimization is employed to efficiently navigate the search space of model configurations. During this optimization, reduced versions of the datasets, D1Sub and D2Sub, which are 1/100th the size of D1 and D2, are used to expedite the evaluation process. The choice of keeping D1Sub and D2Sub constant or resampling them at each optimization step is investigated to understand its impact on the optimization process. The datasets' initial sizes are set to 1 GB for D0 and 2 GB for both D1 and D2, with models having 124 million parameters. The project explores the optimal hyperparameters for Bayesian optimization, such as the number of steps and the method for generating prediction functions. In summary, this project seeks to elucidate the relationship between dataset size, noise reduction, and optimization efficiency, and to uncover optimal strategies for leveraging transfer learning techniques to enhance model performance and generalization in the field of machine learning. Through systematic experimentation and analysis, it aims to contribute valuable insights into the effective implementation of knowledge transfer and model optimization.

2 Optimization

Bayesian Optimization
Bayesian optimization is a powerful technique used to find the optimal hyperparameters of a machine learning model by efficiently navigating the hyperparameter space. This process involves several key steps:

Hyperparameter Space Definition:
The first step is to define the hyperparameters to be optimized and their respective ranges. Common hyperparameters in deep learning models include learning rate, batch size, and the number of epochs. **Initial Sampling:** Initially, the hyperparameter space is sampled randomly or based on some predefined strategy to create a set of initial hyperparameter configurations. These configurations are then used to train the model and evaluate its performance.

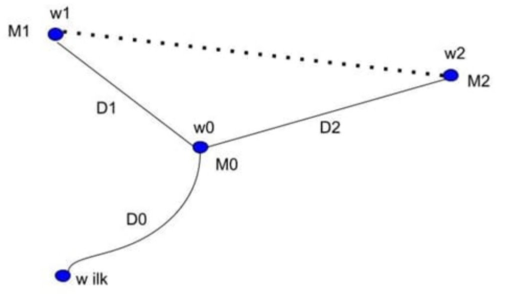


Figure 1:

Modeling and Surrogate Function:
The performance results from the initial sampling are used to create a surrogate model, typically a Gaussian Process (GP). This surrogate model approximates the true performance function and helps to predict the performance of untested hyperparameter configurations.

Acquisition Function:
An acquisition function is used to balance exploration and exploitation, guiding the selection of the next set of hyperparameters to evaluate. The acquisition function identifies the next promising hyperparameter configuration based on the surrogate model's predictions.

Iterative Process:
The process iterates by selecting new hyperparameters using the acquisition function, training the model with these hyperparameters, updating the surrogate model with the new performance results, and repeating until a stopping criterion is met (e.g., a maximum number of iterations or a satisfactory performance level). Impact of Dataset Size and Noise Reduction

Dataset Size:
The project ensures that the datasets D1 and D2

are larger than D0. This choice is crucial because smaller datasets may not significantly benefit from the knowledge transfer, rendering the process less effective. The larger size of D1 and D2 allows for more substantial learning and better performance evaluation.

Noise Reduction:

Using larger datasets helps reduce noise in the training data, leading to more accurate and reliable models. The project examines the trade-off between dataset size and computational cost, as larger datasets increase the training time and resource requirements. Small Dataset Subsampling (D1Sub and D2Sub)

Subsampling:

During the optimization process, reduced versions of the datasets, D1Sub and D2Sub (each being 1/100th the size of D1 and D2), are used. Subsampling helps expedite the evaluation process while maintaining a representative sample of the full dataset.

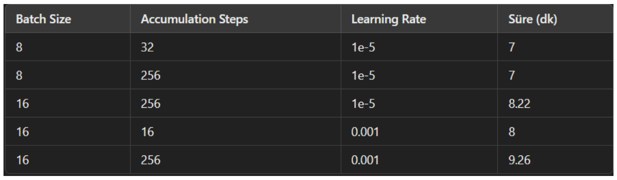
Fixed vs. Resampled Datasets:

The impact of using fixed versus resampled smaller datasets on the optimization outcomes is examined. Fixed datasets provide consistency in evaluation, while resampled datasets can introduce variability but may offer a more comprehensive exploration of the hyperparameter space. Hyperparameters and Optimization Strategy

Hyperparameters:

The project focuses on optimizing key hyperparameters such as the learning rate, batch size, and the method for generating prediction functions. These hyperparameters significantly influence the model's performance and training efficiency.

Optimization Strategy:



Batch Size	Accumulation Steps	Learning Rate	Süre (dk)
8	32	1e-5	7
8	256	1e-5	7
16	256	1e-5	8.22
16	16	0.001	8
16	256	0.001	9.26

Figure 2:

Bayesian optimization is used to systematically explore the hyperparameter space, balancing the exploration of new configurations and the exploitation of known good configurations. This strategy aims to identify the optimal set of hyperparameters that maximize model performance.

Computational Efficiency:

The project seeks to balance the trade-off between increased sample size for accurate predictions and the computational cost associated with each optimization step. Efficient use of computational resources is critical to achieving practical and scalable model optimization.

Conclusion

The optimization section of the project highlights the importance of systematically tuning hyperparameters to enhance model performance and generalization. By leveraging Bayesian optimization, the project aims to find the best configurations for the models trained on different corpora. The careful balance between dataset size, noise reduction, and computational efficiency is crucial for effective knowledge transfer and model optimization.

3 Accuracy Methods

3.1 Purpose and Implementation of Top N Scores

The Top N scores aim to evaluate the models' performance by identifying the top N most likely predictions. This method is particularly useful in language models where the probability of next-word predictions can be ranked. Evaluation Metric: Top N scoring provides a metric to assess how often the correct prediction appears in the top N choices made by the model. This helps in understanding the model's accuracy beyond binary correct/incorrect outcomes.

Application in Model Performance: During the evaluation phase, the model's predictions are ranked, and the top N predictions are compared against the actual results. A higher Top N score indicates that the model consistently ranks the correct answer within its top predictions.

Adjusting Hyperparameters: By analyzing Top N scores, researchers can adjust hyperparameters to improve the likelihood of correct predictions appearing in the top N list. This feedback loop is essential for refining model performance.

Performance Analysis: Top N scores help in detailed performance analysis, revealing how well the model can make probable predictions and where it might need further tuning. This method also highlights the trade-offs between precision and recall.

Conclusion

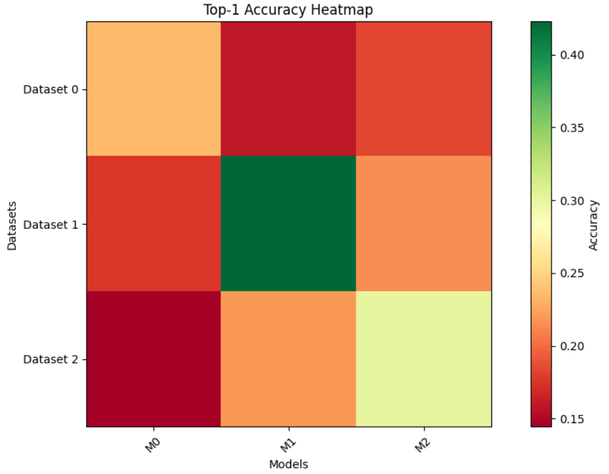


Figure 3:

Optimization methods and Top N scores play a crucial role in enhancing the performance of machine learning models. Bayesian optimization provides a structured approach to hyperparameter tuning, while Top N scores offer a nuanced evaluation of model predictions. Together, they contribute to developing robust and accurate models capable of effective knowledge transfer between diverse corpora.

This detailed exploration underscores the importance of systematic optimization and evaluation techniques in advancing machine learning models' capabilities and their practical applications.

4 Conclusions

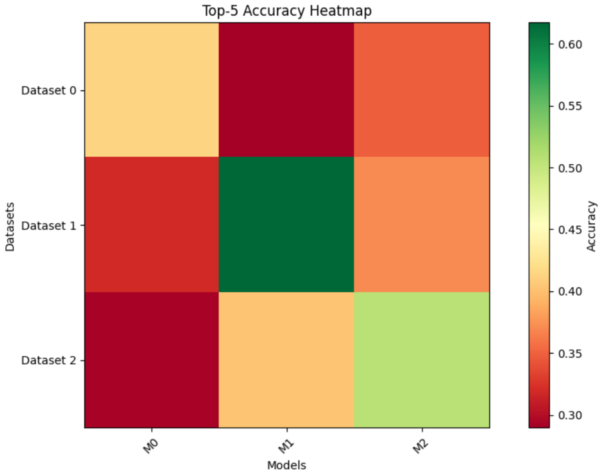


Figure 4:

Bayesian Optimization for U Variable In this project, Bayesian optimization is employed to optimize the variable U in the model weight merging formula: weight-of-merged-model = U*weight-of-M1 + (1-U)*weight-of-M2

The goal is to find the optimal value of U that results in the most successful merged model based on performance metrics.[1] **Optimization Process**

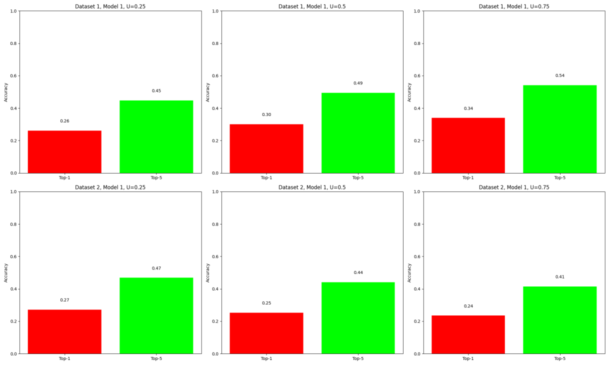


Figure 5:

- **Initial Sampling:**

  The process begins by sampling initial values of U within the range [0, 1]. These initial

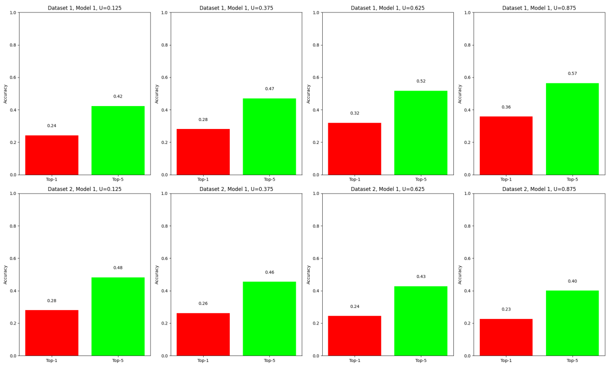


Figure 6:

samples help in creating a baseline for further optimization.[4]

- **Model Evaluation:**

  For each sampled value of U the weights of models M1 and M2 are combined according to the formula. The merged model's performance is then evaluated on the validation set using predefined metrics.

- **Iterative Optimization:**
  The optimization process iteratively updates the surrogate model and acquisition function to converge towards the optimal value of U. Each iteration involves:
  Selecting the next U based on the acquisition function.
  Combining the weights of M1 and M2 using the selected U
  Evaluating the merged model's performance.
  Updating the surrogate model with the new performance data.

- **Convergence Criteria:**
  The optimization continues until a stopping criterion is met, such as a maximum number of iterations or convergence of the acquisition function. The value of U that yields the highest performance is selected as the optimal U.

Purpose and Implementation
Purpose: The primary purpose of optimizing U is to find the best linear combination of the weights of M1 and M2 that maximizes the performance of the merged model. This approach leverages the strengths of both models to achieve superior performance.
Implementation: The implementation involves the following steps: Define the range of U as [0, 1]. Use Bayesian optimization to iteratively sample and evaluate values of U. Combine the weights of M1 and M2 using the sampled U. Evaluate the merged model's performance on a validation set. Update the surrogate model and acquisition function based on the performance data.
Evaluation Metrics
Performance Metrics: The performance of the merged model is evaluated using metrics such as Top-N accuracy. These metrics provide a comprehensive understanding of the model's predictive capabilities[2].
Top-N Scores: Top-N scores are used to measure how often the correct prediction appears within the top N predictions made by the model. This metric is particularly useful for assessing the model's ability to make accurate predictions.
Conclusion
By focusing on optimizing the variable U, this project aims to find the most effective combination of weights [2]from models M1 and M2. Bayesian optimization provides a structured and efficient approach to explore the possible values of U and identify the optimal value that maximizes the merged model's performance. This optimization process is crucial for enhancing the generalization and predictive capabilities of the model, ultimately contributing to better performance in real-world applications.

5 References

References

[1] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[2] Yuchi Ma, Shuo Chen, Stefano Ermon, and David B Lobell. Transfer learning in environmen-

tal remote sensing. *Remote Sensing of Environment*, 301:113924, 2024.

[3] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[4] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[5] Gokul Yenduri, M Ramalingam, G Chemmalar Selvi, Y Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G Deepti Raj, Rutvij H Jhaveri, B Prabadevi, Weizheng Wang, et al. Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*, 2024.