

Article

Not peer-reviewed version

Construction of Automated Machine Learning(AutoML) Framework Based on Large LanguageModels

[Na Sun](#)*, Zidong Yu, Nan Jiang, [Yuxiang Wang](#)

Posted Date: 2 June 2025

doi: 10.20944/preprints202506.0017.v1

Keywords: large-scale language models; automated machinelearning; model generation; hyperparameter optimization



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Construction of Automated Machine Learning (AutoML) Framework Based on Large Language Models

Na Sun ^{1,*}, Nan Jiang ², Zidong Yu ³ and Yuxiang Wang ⁴

¹ Independent Researcher, Newark CA, USA

² Carnegie Mellon University, Pittsburgh PA, USA

³ Syracuse University, College of Engineering and Computer Science, San Jose CA, USA

⁴ School of Business, Stevens Institute of Technology, New York, NY, 10018, USA

* Correspondence: sunna825825@gmail.com

Abstract: As for automatic machine learning (AutoML), which can simplify the model design and adjustment process, continues to develop with the evolution of machine learning technology, and has become one of the factors promoting intelligent applications. Nevertheless, current AutoML frameworks still have a lot of limitations in many aspects when addressing more complex problems, particularly for settings where large-scale language models (LLMs) are utilized for automated learning. In this paper, we present a large-scale language model based automated machine learning framework. First of all, the framework integrates the existing automatic feature engineering and hyperparameter optimization technologies, and further utilizes the intelligent assistance of LLM model in the process of model generation, optimization and model inference to improve the accuracy and efficiency of the whole automation process. Its innovation is the deep fusion of large-scale language models and traditional AutoML working processes, and to automatically generate and fine-tune many machine learning models in multi-modal data and complex task scenarios based on their powerful contextual understanding and generation capabilities to realize more accurate and efficient modeling. Experimental results demonstrate that proposed framework improves the model's adaptive capacity and inference efficiency.

Keywords: large-scale language models; automated machine learning; model generation; hyperparameter optimization

I. Introduction

The last few decades have unprecedented advances in the field of machine learning from algorithms, computational resources, to data availability. Innovations at the algorithm level led us to be able to handle more complex high-dimensional data, continuous enhancement of calculation resources ensures that we can process large-scale data, abundance and diversity of data lead to machine learning widely used in various fields [1]. Furthermore, the selection of machine learning model, hyperparameter tuning, data preprocessing, and entire workflow design require a very specialized knowledge and experience. So how to make machine learning more popular and lower its threshold has become an urgent problem to be solved.

With growing demand for artificial intelligence solutions on the rise in industries everywhere, efficiently, scalably and readily accessible implementations of machine learning methods are growing in importance. Traditional machine learning methods usually requires the expert to go through many times of iterative debug and adjustment in stages, the operation is not only cumbersome, labor-intensive, but also subject to subjective factors. As technology has changed, streamlining these processes in a more automated manner has become critical to boosting productivity and increasing

the range of AI use-cases. Herein, automated machine learning (AutoML) has emerged as an area of interest to researchers [2].

While AutoML has certainly improved the simplicity of the machine learning process, classical AutoML techniques tend to depend on certain fundamental ML algorithms and heuristics. Despite some success in certain narrow domains, they remain brittle against rich modalities, multimodal inputs, and exceedingly dynamic environments. The potential of AutoML was unleashed with large language models (LLMs). These LLMs have shown excellent performance in influencing natural language processing and generation tasks while also performing admirably in a wide variety of application tasks (e.g. image generation). These advancements open up a fresh chance for AutoML and the inclusion of LLMs enhances the intelligence and generality of automated process construction [3].

A key advantage of large language models is their ability to understand and generate multiple types of data. LLMs can deal with structured, semi-structured, and unstructured data, which decides their vital participation in information preprocessing and component extraction. LLMs can optimize machine learning workflows through automatic code generation, model architectures recommendations and optimization strategies. This feature lowers the technical barrier of entry in the field of machine learning and enables researchers and practitioners alike to focus better on solving practical problems instead of unwieldy technical implementations.

However, there are also challenges in integrating LLM with AutoML frameworks. First, the complexity of LLMs makes them extremely computationally demanding, requiring far more resources and time than traditional machine learning algorithms. Second, although LLMs have demonstrated great capabilities in many tasks, their "black box" nature is still a problem that needs to be solved, especially in fields that require a high degree of transparency and explainability, such as healthcare and finance.

II. Related Work

Zeineddine et al. [4] proposed an AutoML-based approach that uses behavioral and academic data from students before enrollment to automatically select the best model to improve the accuracy of students' successful predictions. Chen et al. [5] proposed iLearnPlus web-based machine learning platform, designed for the analysis, prediction, and visualization of nucleic acid and protein sequences. The platform provides comprehensive algorithmic support to automate functions such as sequence feature extraction, model construction and deployment, predictive performance evaluation, statistical analysis, and data visualization to complete complex bioinformatics tasks without programming. Ma et al. [6] used landslide data and environmental factors in the Three Gorges Reservoir area to automatically construct a landslide susceptibility prediction model. Compared to traditional machine learning methods, AutoML provides a more efficient model selection and optimization process, improving the accuracy and reliability of predictions.

Sun et al. [7] applied an AutoML workflow to mesh GRACE satellite data to estimate total water storage in the continental United States. By automatically selecting the optimal algorithm and model structure, the researchers overcame the challenges of traditional methods in large-scale data processing and improved the efficiency and accuracy of water resource monitoring. Zöller et al. [8] review of current AutoML methods and benchmarking of popular AutoML frameworks on real-world datasets. The authors evaluated several open-source frameworks, such as Auto-Sklearn, TPOT, and H2O.ai, and analyzed their effectiveness in automating machine learning processes

Tannemaat et al. [9] proposed an automated time series classification algorithm for distinguishing normal, neuropathic, and myopathic electromyography (EMG) signals. Deng et al. [10] used the AutoML method, combined with algorithms such as TPOT and H2O, to analyze the effects of biochar with different electrochemical properties in the anaerobic digestion process. Khuat et al. [11] explore the role and patterns of human-computer interaction (HCI) in AutoML systems, analyzing current practices, limited but fully automated ways of interacting, how to interact in open environments.

III. Methodologies

A. Data Preprocessing and Feature Engineering

Data preprocessing and augmentation is a crucial step in an automated machine learning framework, especially when working with unstructured data. By introducing Massive Language Models (LLMs), we are able to automatically generate preprocessing processes that are adapted to specific tasks. Suppose our input data $X = \{x_1, x_2, \dots, x_n\}$ is the raw text data, and our goal is to vectorize the text. A common approach to text vectorization is to use the TF-IDF (Word Frequency-Inverse Document Frequency) model, as shown in Equation 1:

$$TF - IDF(x_i, t) = TF(x_i, t) \cdot \log\left(\frac{n}{DF(t)}\right), \quad (1)$$

where $TF(x_i, t)$ is the frequency of the word t in text x_i , $\left(\frac{n}{DF(t)}\right)$ is the number of documents containing the word t , and n is the total number of documents.

By calculating the TF-IDF value for each word, we are able to generate a vectorized representation of the feature for each document. In this process, LLM can help optimize the selection of words and the calculation of weights, so that the vectorized features can be more suitable for the task requirements. In addition, we can augment the training set with data augmentation, and for text data, common data augmentation methods include synonym substitution, random insertion, and deletion.

For each input text x_i , we can introduce a booster function $\mathcal{A}(x_i, \alpha)$ where α controls the intensity of the enhancement. Suppose we want to enhance the text with synonym substitution, the enhanced text can be expressed as Equation 2:

$$x'_i = \mathcal{A}(x_i, \alpha) = \text{ReplaceSynonym}(x_i, \alpha), \quad (2)$$

where x'_i is the enhanced text data, and the enhanced function \mathcal{A} decides which words to replace according to the needs of the task.

The core of feature engineering lies in how to process the original data through some predefined conversion functions f , such as normalization, normalization, and cross-feature generation. For example, if we use normalization to process data x_i , the transformed feature can be expressed as Equation 3:

$$x_i = f(x_i) = \frac{x_i - \mu}{\sigma}, \quad (3)$$

where μ and σ are the mean and standard deviation of the features, respectively. This transformation makes the mean value of the features 0 and the variance 1, which improves the stability and convergence speed of subsequent model training.

Let r_i be the performance of model m_i on the task (e.g., accuracy, F1 value, etc.), and we hope that by learning a strategy $\pi(a|X)$, we can choose a schema that maximizes the expected return for each data point X , as shown in Equation 4:

$$\pi(a|X) = \arg \max_a \mathbb{E}[r_a(X)], \quad (4)$$

where a represents the action taken (i.e., the chosen model architecture), $r_a(X)$ is the performance of the architecture on input X , and the expected $\mathbb{E}[r_a(X)]$ represents the return evaluation of the model architecture over the long run.

By introducing reinforcement learning, we are able to continuously adjust and optimize the model architecture based on feedback from each architecture choice, enabling more efficient learning in multi-task and complex data scenarios. We introduce an adaptive mechanism to optimize TF-IDF weights, and adjust the vocabulary selection and weight calculation. For synonym substitution, we optimize the enhancement effect by dynamically adjusting the enhancement intensity η according to the needs of the task to ensure that the generated data is more in line with the mission objectives.

B. Hyperparameter Optimization and Data Fusion

In our framework, we have introduced large-scale language models (LLMs) to assist in the generation of model architectures. Assuming that we want to select the best architecture from

multiple candidate models, we can use a generative model g_θ to generate a candidate model architecture m_i from model space \mathcal{M} , as shown in Equation 5:

$$m_i = g_\theta(X, Z), \quad (5)$$

Among them, g_θ is the generative network trained by the language model, θ is its parameter, and X and Z are the features generated after the original input data and feature engineering, respectively. By leveraging the LM's ability to understand the task, the model can be automatically formed and the model architecture can be adjusted, so that the generated model can be more suitable for complex tasks and multimodal data.

In this paper, we propose an adaptive hyperparameter optimization method based on LLM. Suppose that the set of hyperparameters of model m_i is θ_i , which contains the learning rate and regularization coefficient of the model. By introducing LLMs, we can dynamically adjust the search space of hyperparameters during the optimization process. Specifically, assuming that we have obtained an initial hyperparameter combination θ_0 , we adjust it step by step through the optimization process so that the performance of the model gradually improves, optimization goal describes as Equation 6:

$$\theta_{opt} = \arg \max_{\theta} \mathcal{L}(m_\theta, X, y), \quad (6)$$

where, $\mathcal{L}(m_\theta, X, y)$ represents the loss function of the model m_θ on the input data X and label y . By introducing the intelligent assistance of LLM, we can quickly find the optimal solution in a large-scale hyperparameter space and reduce the computational overhead common in traditional methods. **By introducing domain knowledge, LLMs can generate appropriate hyperparameter combinations based on description of input task and improve search space by step-by-step optimization.**

In order to improve the overall performance, we design an optimization strategy based on model fusion, assuming that we have trained multiple models $\{m_1, m_2, \dots, m_k\}$, we can integrate the prediction results of these models by weighted fusion. The specific convergence strategy can be expressed as Equation 7:

$$\hat{y} = \sum_{i=1}^k w_i \cdot m_i(X), \quad (7)$$

where w_i is the weight of the i -th model, $m_i(X)$ is the prediction result of the i -th model on the input data X , and \hat{y} is the final prediction output. By optimizing these weights w_i , we can improve the generalization ability and prediction accuracy of the model for different tasks.

When building machine learning models, the generalization ability of the model is crucial. In order to avoid overfitting, we introduce a regularization-based strategy to enhance the generalization ability of the model. Assuming that the loss function of the model is $\mathcal{L}(m, X, y)$, we can limit the complexity of the model by using the regularization term $\mathcal{R}(m)$ to improve its generalization ability. Common forms of regularization include L2 regularization and L1 regularization, loss function after regularization is Equation 8:

$$\mathcal{L}_{reg}(m, X, y) = \mathcal{L}(m, X, y) + \lambda \|m\|_2^2, \quad (8)$$

where λ is the regularization hyperparameter, which controls the weight of the regularization term. By adjusting λ , we were able to balance the fitting ability and complexity of the model during the training process, avoiding the model from overfitting the training data. **Traditional feature engineering generates the underlying features through predefined conversion functions, while LLMs generate task descriptions and return optimization instructions for feature engineering.**

IV. Experiments

A. Experimental Setup

The experiment uses a classic AutoML benchmark dataset from the UCI machine learning library, which covers a variety of machine learning tasks such as classification, regression, and recommender systems, including structured data, image data, and text data. The dataset is characterized by its task diversity and annotation accuracy, which is very suitable for building and

testing large language model (LLM)-based AutoML frameworks, especially in model selection automation, feature engineering, and cross-domain applications.

B. Experimental Analysis

In order to fully evaluate its performance and benefits, the experiments were compared with existing mainstream, four common baseline methods:

- Auto-sklearn is an automated machine learning tool based on scikit-learn that uses Bayesian optimization, meta-learning, and ensemble building to automatically select algorithms and adjust hyperparameters.
- TPOT is a genetically programming-based AutoML tool that is able to automatically search machine learning pipelines through evolutionary algorithms, suitable for exploratory data analysis and model discovery.
- H2O AutoML provides an open-source, automated machine learning platform that supports a variety of algorithms and models to automate data preprocessing, feature engineering, model training, and evaluation.
- MLJAR is an automated machine learning platform that provides an easy-to-use interface that supports automated data preprocessing, feature selection, and model training for rapid building and deployment of machine learning models.

Accuracy is the most common classification task evaluation metric, and the proportion of samples that are correctly predicted by the calculation model is the proportion of all samples. Figure 1 shows that with the increase of training epochs, the accuracy of various AutoML methods generally shows an upward trend and eventually converges. Ours-AutoML showed the highest accuracy and fast convergence of all methods, indicating that our method is better able to optimize the model and achieve higher performance during training.

In contrast, Auto-sklearn and TPOT also have significant improvements in accuracy, but there are some fluctuations in some epoch segments, suggesting that they may be affected by more random factors at certain parameter settings. H2O AutoML and MLJAR showed a steady improvement trend during training, but they still failed to reach the accuracy of our method at the final convergence.

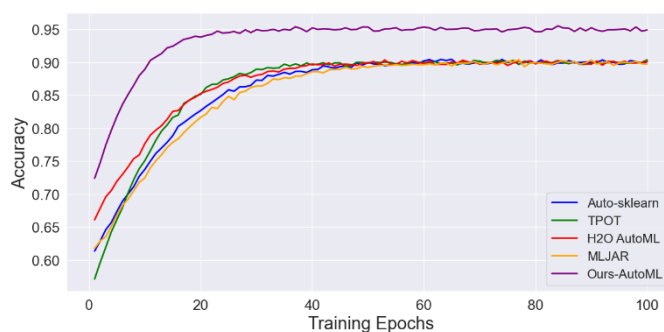


Figure 1. Accuracy Comparison of Different AutoML Methods.

As can be seen in Figure 2, the mean square error gradually decreases and tends to level off as the model parameters increase. As you can see from this graph, Ours-AutoML exhibits the smallest mean square error across the entire range, indicating that our method has stronger performance in parameter tuning and model optimization. In contrast, Auto-sklearn, TPOT, and H2O AutoML, while performing well in some areas, generally have a large mean square error, especially when the model complexity is high.

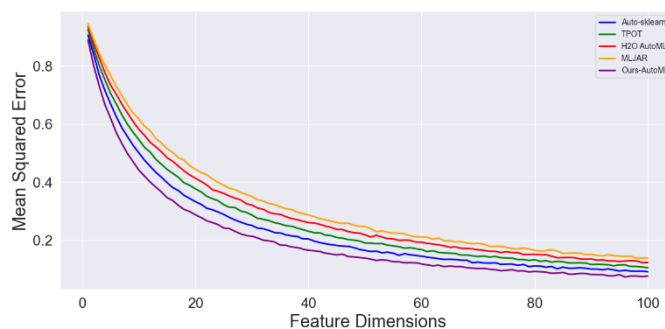


Figure 2. MSE Comparison of Different AutoML Methods.

Figure 3 shows the distribution of training time for each method over multiple experiments. Boxplots allow you to visually compare the median, quartile, and outliers of each method's training time. As you can see, the training time of Ours-AutoML is generally shorter, while the training time of TPOT and H2O AutoML is longer.

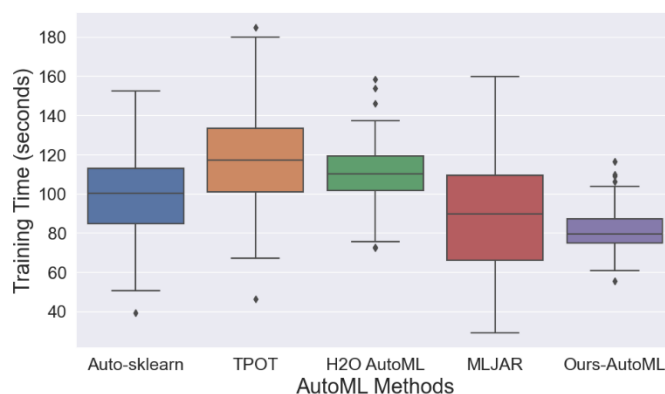


Figure 3. Model Training Time Comparison.

V. Conclusion

In conclusion, we propose an automated machine learning framework based on large-scale language models, which combines the existing feature engineering and hyperparameter optimization techniques, and uses the intelligent assistance of LLMs in model generation, optimization and inference to improve the accuracy and efficiency of the automation process. In the future, with the further development of LLM technology, this framework is expected to show greater capabilities in more complex tasks, especially when dealing with diverse data.

References

1. Wang, C., Wu, Q., Liu, X., & Quintanilla, L. (2022, August). Automated machine learning & tuning with flaml. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 4828-4829).
2. Fadzail, N. F., Zali, S. M., Mid, E. C., & Jailani, R. (2022, August). Application of Automated Machine Learning (AutoML) method in wind turbine fault detection. In Journal of Physics: Conference Series (Vol. 2312, No. 1, p. 012074). IOP Publishing.
3. Escalante, H. J., Yao, Q., Tu, W. W., Pillay, N., Qu, R., Yu, Y., & Houlby, N. (2021). Guest editorial: Automated machine learning. IEEE Transactions on Pattern Analysis & Machine Intelligence, 43(09), 2887-2890.
4. Zeineddine, H., Braendle, U., & Farah, A. (2021). Enhancing prediction of student success: Automated machine learning approach. Computers & Electrical Engineering, 89, 106903.

5. Chen, Z., Zhao, P., Li, C., Li, F., Xiang, D., Chen, Y. Z., ... & Song, J. (2021). iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization. *Nucleic acids research*, 49(10), e60-e60.
6. Ma, J., Lei, D., Ren, Z., Tan, C., Xia, D., & Guo, H. (2024). Automated machine learning-based landslide susceptibility mapping for the three gorges reservoir area, China. *Mathematical Geosciences*, 56(5), 975-1010.
7. Sun, A. Y., Scanlon, B. R., Save, H., & Rateb, A. (2021). Reconstruction of GRACE total water storage through automated machine learning. *Water Resources Research*, 57(2), e2020WR028666.
8. Zöllner, M. A., & Huber, M. F. (2021). Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research*, 70, 409-472.
9. Tannemaat, M. R., Kefalas, M., Geraedts, V. J., Remijn-Nelissen, L., Verschuuren, A. J. M., Koch, M., ... & Bäck, T. H. W. (2023). Distinguishing normal, neuropathic and myopathic EMG with an automated machine learning approach. *Clinical Neurophysiology*, 146, 49-54.
10. Deng, Y., Zhang, Y., & Zhao, Z. (2024). A data-driven approach for revealing the linkages between differences in electrochemical properties of biochar during anaerobic digestion using automated machine learning. *Science of the Total Environment*, 927, 172291.
11. Khuat, T. T., Kedziora, D. J., & Gabrys, B. (2022). The roles and modes of human interactions with automated machine learning systems. *arXiv preprint arXiv:2205.04139*.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.