**Adapting Large Language Model for**

**JSON Extraction from Text Corpora**

|  |
| --- |
| Van Tuan-Tran1\*, Chin-Shiuh Shieh 2, Mong-Fong Horng3 |
| 1 National Kaohsiung University of Science and Technology / Student 1  2 National Kaohsiung University of Science and Technology / Professor 2,3  \*Email: {1f111169109, 2csshieh, 3mfhorng@nkust.edu.tw} |
| NSTC-112-2221-E-992-045, NSTC-112-2221-E-992-057-MY3  NSTC-112-2622-8992-009-TD1. |

# Abstract

This paper explores the adaptation of large language models (LLMs) for the task of extracting structured data in JSON format from extensive and unstructured text corpora. Specifically, we fine-tuned the Llama-2-7 billion model utilizing a combination of advanced techniques, including QLoRA (Quantized Low-Rank Adaptation), Fully Sharded Data Parallel (FSDP) training, and distributed training across multiple GPUs. Our approach addresses the challenges associated with processing large and complex datasets, which include HTML-tagged content, diverse textual paragraphs, and various formatting irregularities. By employing QLoRA, we achieve efficient low-rank adaptation, which helps in reducing the computational burden while retaining model performance. FSDP enables us to handle large-scale data by sharding model parameters and gradients, thus optimizing memory usage and speeding up the training process. Multi-GPU training further enhances scalability and accelerates the fine-tuning process, allowing us to manage and process extensive text corpora effectively. The fine-tuning process transforms raw textual information into structured JSON outputs, facilitating automated data extraction and processing. Our results demonstrate that the adapted model significantly improves the accuracy and efficiency of extracting relevant information from complex text sources, compared to traditional methods. This work not only showcases the effectiveness of combining these advanced techniques but also offers a scalable solution for applications such as web scraping, data parsing, and large-scale information retrieval.

**Keywords:** Fully Sharded Data Parallel (FSDP), Meta-Llama/Llama-2-7b, Meta-Llama/Llama-3-8b, Meta-Llama/Llama-3.1-8b, QLoRA, LoRA.

## Introduction

In the era of big data, extracting structured information from unstructured text is a critical task for various applications, including data analytics, information retrieval, and natural language understanding. JSON (JavaScript Object Notation) has become a standard format for representing structured data due to its simplicity and ease of use. However, extracting JSON data from raw text remains a challenging problem, especially when dealing with large volumes of diverse and complex textual corpora. Large Language Models (LLMs) have recently shown exceptional performance in various natural language processing (NLP) tasks, thanks to their ability to understand and generate human-like text. Models like GPT-3 and LLaMA2-7B have demonstrated the potential to comprehend context and generate structured responses. Despite these advancements, the application of LLMs for directly extracting structured data, such as JSON, from unstructured text has not been fully explored. This paper addresses this gap by investigating the adaptation of LLMs for the task of JSON extraction from text corpora. Our approach focuses on fine-tuning the LLaMA2-7B model using techniques such as Quantized Low-Rank Adaptation (QLoRA) and Fully Sharded Data Parallel. By training the model on a customized dataset, we aim to enhance its ability to identify and extract structured data in the form of JSON. The primary objective of this research is to develop a method that improves the efficiency and accuracy of JSON extraction compared to traditional methods. By leveraging the advanced capabilities of LLMs, our work aims to provide a robust solution for converting textual data into structured formats, thereby facilitating data processing and analysis in various domains.

In the following sections, we discuss the related work in the field, describe our methodology in detail, present the experimental results, and conclude with potential implications and future research directions.

### Related works

The task of extracting structured data from unstructured text has been extensively studied in the field of natural language processing (NLP). Traditional approaches have often relied on rule-based systems, regular expressions, or statistical methods to identify and extract specific data points from text. While effective in certain contexts, these methods often require significant manual effort to adapt to new domains or data formats and may struggle with the complexity and variability of natural language. Recent advancements in Large Language Models (LLMs) have opened new avenues for automating various NLP tasks. Models such as GPT-3 and LLaMA2 have demonstrated exceptional capabilities in understanding and generating human-like text. These models are trained on vast corpora and can be fine-tuned for specific applications, making them powerful tools for tasks ranging from text generation to machine translation. However, the direct application of LLMs for extracting structured data, such as JSON, from unstructured text has not been widely explored in the literature. Several studies have explored the use of LLMs for information extraction tasks. For example, fine-tuning approaches like BERT for named entity recognition (NER) and relation extraction have shown promising results. Similarly, techniques such as QLoRA (Quantized Low-Rank Adaptation) and Fully Sharded Data Parallel have been introduced to enhance the fine-tuning process, particularly when adapting large models for specific tasks. These methods help reduce memory usage and training time, making it feasible to adapt large models like LLaMA2-7B for specialized applications.

Despite these advancements, there remains a gap in the application of LLMs for direct JSON extraction from text corpora. Most existing approaches focus on either general-purpose text generation or specific information extraction tasks without addressing the conversion of unstructured text into structured JSON formats. This gap highlights the need for research that explores the potential of LLMs in automating the JSON extraction process, particularly in terms of improving accuracy and reducing manual intervention. Our work aims to bridge this gap by adapting LLMs, specifically LLaMA2-7B, for the task of JSON extraction. By employing advanced fine-tuning techniques such as QLoRA and Fully Sharded Data Parallel, we seek to enhance the model's ability to identify and extract structured information from diverse text corpora. This research contributes to the field by demonstrating the feasibility and effectiveness of using LLMs for structured data extraction, offering a robust alternative to traditional methods.

### Methodology

### 3.1. Overview

This research focuses on adapting all the Meta-Llama model for the task of extracting JSON structures from unstructured text corpora. The methodology involves fine-tuning the base model using advanced techniques to enhance its ability to generate accurate and structured JSON outputs from diverse textual inputs.

**3.2. Dataset Preparation**

A customized dataset was constructed to train and evaluate the model. The dataset comprises various text samples paired with their corresponding JSON representations. These samples were sourced from a combination of publicly available datasets and manually annotated texts to ensure a diverse and representative collection. Preprocessing steps included text normalization, tokenization, and filtering to remove noise and irrelevant information. The dataset was split into training, validation, and test sets to facilitate model training and evaluation. Here is the dataset that I have preprocessed and published on HuggingFace.

**3.3. Model Architecture**

The base model used in this study is Llama-2-7b, Llama-3-8b, Llama-3.1-8b. These large language model known for its robust natural language understanding capabilities. LLaMA was chosen due to its ability to handle complex language patterns and generate coherent outputs. No major architectural modifications were made to the model, as the primary focus was on fine-tuning its weights to adapt to the JSON extraction task.

**3.4. Fine-tuning Process**

Fine-tuning was performed using Quantized Low-Rank Adaptation (QLoRA) and Fully Sharded Data Parallel (FSDP) techniques. QLoRA was employed to reduce the computational complexity of the fine-tuning process by approximating the full model weights with low-rank matrices. This approach allows the model to retain its expressive power while being adapted to the specific task of JSON extraction. Fully Sharded Data Parallel (FSDP) was utilized to efficiently distribute the model's training across multiple GPUs. FSDP enables the model to be split into shards that are distributed across different devices, allowing for parallel processing and reducing memory requirements. This approach was particularly useful given the size of Llama2-7b, Llama3-8b, Llama3.1-8b and the need for efficient fine-tuning on large datasets.

**3.5. Training Setup**

The training was conducted on a multi-GPU setup using PyTorch and the Hugging Face Transformers library. Key hyperparameters, were optimized, including learning rate, batch size, and the number of training epochs.

An initial learning rate of 2e-5 was selected, with gradual decay throughout the training process, gradient accumulation is 4, mixed precision training is were also employed to optimize the training efficiency and reduce memory consumption.

|  |  |
| --- | --- |
| Hyper-Parameters | Values |
| Model name | meta-llama/Llama-2-7b-hf  meta-llama/Llama-3-8b-hf  meta-llama/Llama-3.1-8b-hf |
| New model name | llama-2-7b-json\_extract-lora\_adapter  llama-3-8b-json\_extract-lora\_adapter  llama-31-8b-json\_extract-lora\_adapter |
| Dataset | chwenjun225/Instruction\_top\_5\_insurance  \_brands\_june\_news\_and\_twitter\_only |
| Batch size | 2 |
| Context length | 4096 |
| Precision | torch.float16 |
| Gradient accumulation steps | 4 |
| Epochs | 5 |
| Training type | QLoRA |
| Inference type | LoRA adapter |

**Table 1:** Hyper-Parameter for training

**3.6. Evaluation Metrics**

The model's performance was evaluated using EleutherAI/Language Model Evaluation Harness metrics such as loss, and accuracy. The generated JSON outputs were also assessed for structural correctness and completeness, ensuring that the extracted information accurately matched the target formats. Here is the following task, that I have used for evaluating all of my fine-tuned models.

* ***Arc\_challenge***: consists of multiple-choice science questions that are challenging and require reasoning and domain-specific knowledge to solve. These questions often require more complex inference and a deeper understanding of the subject matter, including common sense reasoning, understanding of scientific concepts, and the ability to apply these concepts in novel ways.
* ***Arc\_easy***: The questions in this set are more straightforward and can often be answered with simpler reasoning or retrieval of factual knowledge. They still cover a range of science topics but generally require less complex reasoning than the challenge set.
* ***Hellaswag***: Tests a model's ability to understand and predict the likely progression of events or situations, which requires a strong grasp of common sense and everyday knowledge.
* ***Openbookqa***: This task evaluates a model's ability to use background knowledge in a flexible way to reason about questions. It requires the model not only to recall facts but also to apply them in new contexts.
* ***Piqa***: This task tests a model's understanding of the physical world and its ability to apply that understanding to infer the most likely outcomes of physical events or interactions.

1. **Experiment**

We performed the entire work on two graphics processors, the RTX3090 GPU (24GB VRAM) and RTX4070 (12 VRAM), the training time was more than 72 hours. We performed on three models, namely llama-2-7 billion parameters abbreviated as llama-2-7b, llama-3-8 billion parameters abbreviated as llama-3-8b and llama-3.1-8 billion parameters abbreviated as llama-31-8b, with the proposed methodology above, the following are the results we experimented on the self-collected dataset and evaluated based on the excellent repository of EleutherAI/llm-harness-evaluation.

**4.1. Llama-2-7 bilion parameters**

We aimed to evaluate the accuracy of barcode localization using YOLOv8. We measured the precision, recall, and F1-score of the barcode detection process on our diverse dataset. The metrics were calculated by comparing the ground truth barcode regions with the regions identified by YOLOv8.

**Fig. 2. Barcode localization accuracy**

The experiment assessed the model's capability to accurately locate barcode regions within complex and varied images.

**4.2. Restoration quality and decoding accuracy**

Focus on evaluating the effectiveness of REAL-ESRGAN in improving the quality and readability of barcode images. The results show that the REAL-ESRGAN model has removed noise and blur and returned the restored image to its original state. We then measured the image super-resolution ability of REAL-ESRGAN by comparing the quality of the recovered barcode images with the original high-resolution images from our dataset. Next aims to evaluate the overall accuracy and reliability of our barcode decoding system. We used Pyzbar to decode barcode images and measured decoding accuracy on images subjected to a variety of challenges, including blur, noise, and environmental brightness changes. This experiment determined the extent to which our integrated approach mitigates the limitations commonly associated with barcode decoding under adverse conditions.

|  |  |  |
| --- | --- | --- |
|  |  | A close-up of a qr code  Description automatically generated |

**Fig. 3. The region of QR-code was restored using REAL-ESRGAN**

|  |  |  |
| --- | --- | --- |
| A bar code on a white surface  Description automatically generated |  | A bar code with numbers and a green line  Description automatically generated |

**Fig. 4. The region of barcode UPC was restored using REAL-ESRGAN**

|  |  |  |
| --- | --- | --- |
|  |  | A close up of a vent  Description automatically generated |

**Fig. 5. The region of barcode CODE-128 was restored using REAL-ESRGAN**

Furthermore, the tests confirm the synergistic effect of combining YOLOv8, REAL-ESRGAN, and Pyzbar in solving the challenges associated with barcode image processing, ensuring speed and accuracy in these applications changing environmental conditions as we can see in Fig. 6.

A diagram of a barcode

Description automatically generated

**Fig. 6. System work flow on practical data**

**4.4. Compare with the previous works**

**Table. 1.** **Comparison REAL-ESRGAN with previous works, on data objects specified as image barcodes and QR codes**

|  |  |  |
| --- | --- | --- |
| **Works** | **Inputs** | **Outputs** |
| **SwinIR** [20] | **A bar code on a white surface  Description automatically generated** | **A bar code on a white surface  Description automatically generated**  **A barcode on a white surface  Description automatically generated**  **A close-up of a qr code  Description automatically generated** |
| **ESRGAN** [21] | **A bar code on a white surface  Description automatically generated** | **A bar code with numbers  Description automatically generated**  **A bar code with writing on it  Description automatically generated**  **A qr code on a wall  Description automatically generated** |

|  |  |  |
| --- | --- | --- |
| **RealSR** [22] |  |  |
| **BSRGAN**  [23] |  |  |
| **Real-**  **ESRGAN**  [19] |  |  |

Explanation of metrics:

**Table. 2. Compare the calculated metrics with the ground truth image**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input** | **SwinIR** [20] | **ESRGAN** [21] | **RealSR** [22] | **BSRGAN** [23] | **Real-ESRGAN** [19] | **Ground-truth** |
| **Barcode UPC** | **A bar code on a white surface  Description automatically generated** | **A close up of a bar code  Description automatically generated** | **A bar code on a white surface  Description automatically generated** | **A bar code on a white surface  Description automatically generated** | **A bar code on a white surface  Description automatically generated** |  | **A bar code on a yellow background  Description automatically generated** |
| **Width and Height** | 137x66 | 548x264 | 548x264 | 548x264 | 548x264 | 548x264 |  |
| **PSNR (dB)** |  | 6.99 | 6.70 | 6.55 | 6.37 | 6.28 |
| **SSIM [0,…1]** |  | 0.0734 | 0.0733 | 0.0781 | 0.0584 | 0.0783 |
| **Time reference (ms)** |  | 200 | 188 | 235 | 218 | 244 |
| **Barcode EAN** |  | **A barcode on a white surface  Description automatically generated** | **A bar code with writing on it  Description automatically generated** | **A bar code on a white surface  Description automatically generated** | **A bar code on a white surface  Description automatically generated** | **A close up of a window  Description automatically generated** | **A bar code with numbers  Description automatically generated** |
| **Width and Height** | 103x40 | 412x156 | 412x156 | 412x156 | 412x156 | 412x156 |  |
| **PSNR (dB)** |  | 6.94 | 6.79 | 6.69 | 6.55 | 6.56 |
| **SSIM [0,…1]** |  | 0.0956 | 0.0867 | 0.0887 | 0.0664 | 0.0758 |
| **Time reference (ms)** |  | 119 | 87 | 25 | 22 | 206 |
| **QR-Code** |  | **A qr code on a white surface  Description automatically generated** | **A qr code on a wall  Description automatically generated** | **A qr code on a white surface  Description automatically generated** | **A qr code on a wall  Description automatically generated** |  | **A qr code on a yellow background  Description automatically generated** |
| **Width and Height** | 37x46 | 144x180 | 144x180 | 144x180 | 144x180 | 144x180 |  |
| **PSNR (dB)** |  | 6.99 | 6.89 | 6.65 | 6.77 | 6.86 |
| **SSIM [0.…1]** |  | 0.1055 | 0.1174 | 0.1202 | 0.1129 | 0.1198 |
| **Time reference (ms)** |  | 101 | 139 | 26 | 24 | 188 |

* Width and Height: Width and height of image.
* PSNR (dB): Computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image.

R is the maximum fluctuation in the input image data type. If the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

The MSE represents the cumulative squared error between the compressed and the original image

* SSIM: Computed for the image with respect to the reference image. The reference image is usually needs to be of perfect quality. This quantitative measure considers three parameters namely luminance, contrast and structural information between the two images to computed the SSIM value.
* Time reference (ms): Model time for restoring photos is measured in milliseconds.

**A graph of different colored lines

Description automatically generated**

**Fig.7. The chart of PSNR metrics**

A graph of different colored lines

Description automatically generated

**Fig.8. The chart of SSIM metrics**

A graph of a number of lines

Description automatically generated with medium confidence

**Fig.9. The chart of Time Reference**

When looking at the metrics evaluation, we see that the PSNR index of the models when compared to the ground-truth image is all below 20. The SSIM index is all below 0.2, although these indices are really low compared to the ground-truth image, so these methods still need to improve in terms of algorithms as well as training data in the future. But with the naked eye, we can see that the Real-ESRGAN [19] method has the best image recovery ability among all the remaining methods.

1. **Conclusion**

As we conclude this study, we reflect on the significance of our findings and the broader implications of this integrated system. Unlocking barcode localization accuracy: The first step in our workflow involves resizing images using the OpenCV2 library, ensuring compatibility with the YOLOv8 model. This preparation not only helps optimize processing speed but also lays the foundation for accurate barcode positioning. Even in challenging environments due to varying lighting conditions and image distortions, YOLOv8 demonstrated outstanding accuracy, emphasizing its effectiveness in locating barcodes. The test shows REAL-ESRGAN's excellent ability to improve the quality and readability of barcode images, REAL-ESRGAN's image super-resolution capabilities, such as restoring barcode images to their original state. pristine, dramatically improving the clarity and fidelity of barcode presentation. This step is important in maximizing the accuracy of subsequent barcode decoding. The final step is to decode the barcode using Pyzbar, which will create an integrated system capable of decoding barcode images with outstanding accuracy. While Pyzbar may falter in such situations, our integrated approach exploits the strengths of both YOLOv8 and REAL-ESRGAN to overcome these challenges and achieve reliability. Implications and future directions: The implications of this research extend across industries where barcode technology is indispensable. As we conclude this study, we acknowledge that this integrated approach is only a stepping stone toward a more complex and flexible barcode decoding system. Future research directions could explore further improvements to the YOLOv8 model, fine-tune REAL-ESRGAN for specific barcode types, and develop Pyzbar enhancements to handle challenging imaging conditions awake. In a border context, the combination of computer vision, super-resolution imaging and barcode decoding illustrates the profound potential of interdisciplinary approaches in solving the world's complex challenges. gender. Finally, we compare the REAL-ESRGAN method with old super-resolution methods and the visual evaluation results show that REAL-ESRGAN recovers barcode images in the best state.

**Acknowledgement**

This research was partly supported by the National Science and Technology Council, Taiwan with grant numbers NSTC 112-2221-E-992-045.

**References**

1. O. Gallo and R. Manduchi, “*Reading 1d barcodes with mobile phones using deformable templates”*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 9, pp. 1834-1843, 2011.
2. G. Sörös and C. Flörkemeier, *"Blur-resistant joint 1d and 2d barcode localization for smartphones",* Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia, pp. 1-8, 12 2013.
3. A. Zamberletti, I. Gallo, M. Carullo and E. Binaghi, *"Neural image restoration for decoding 1-d barcodes using common camera phones",* VISAPP, vol. 1, pp. 5-11, 01 2010.
4. H. Zhang, G. Shi, L. Liu, M. Zhao and Z. Liang, *"Detection and identification method of medical label barcode based on deep learning",* 2018 Eighth International Conference on Image Processing Theory Tools and Applications (IPTA), pp. 1-6, 11 2018.
5. D. K. Hansen, K. Nasrollahi, C. B. Rasmusen and T. B. Moeslund, *"Real-time barcode detection and classification using deep learning"*, IJCCI, pp. 321-327, 2017.
6. J. Redmon and A. Farhadi, *"Yolov3: An incremental improvement"*, CoRR, vol. abs/1804.02767, 2018, [online] Available: http: //arxiv.org/abs/1804.02767.
7. D. A. Fish, A. M. Brinicombe, E. R. Pike and J. G. Walker, *"Blind deconvolution by means of the richardson–lucy algorithm",* J. Opt. Soc. Am. A, vol. 12, no. 1, pp. 58-65, 01 1995.
8. A. D. Hillery and R. T. Chin, *"Iterative wiener filters for image restoration"*, IEEE Transactions on Signal Processing, vol. 39, no. 8, pp. 1892-1899, 08 1991.
9. D. Trong, C. Phuong, T. Tuyen and D. Thanh, *"Tikhonovs regularization to the deconvolution problem"*, Communication in StatisticsTheory and Methods, vol. 43, pp. 4384-4400, 10 2014.
10. S. Esedoglu, *"Blind deconvolution of bar code signals"*, Inverse Problems, vol. 20, pp. 121-135, 07 2004.
11. S. Yahyanejad and J. Strm, *"Removing motion blur from barcode images"*, CVPRW, pp. 41-46, 7 2010.
12. Y. Lou, E. Esser, H. Zhao and J. Xin, *"Partially blind deblurring of barcode from out-of-focus blur"*, SIAM Journal on Imaging Sciences [electronic only], vol. 7, pp. 740-760, 4 2014.
13. S. Nah, T. H. Kim and K. M. Lee*, "Deep multi-scale convolutional neural network for dynamic scene deblurring"*, IEEE CVPR, pp. 257-265, 7 2017.
14. M. Noroozi, C. Paramanand and P. Favaro, *"Motion deblurring in the wild"*, Lecture Notes in Computer Science, pp. 65-77, 01 2017.
15. X. Tao, H. Gao, X. Shen, J. Wang and J. Jia, *"Scale-recurrent network for deep image deblurring",* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8174-8182, 6 2018.
16. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, et al., *"Generative adversarial networks",* ArXiv, vol. Abs/1406.2661, 2014.
17. O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin and J. Matas, *"Deblurgan: Blind motion deblurring using conditional adversarial networks",* IEEE/CVF CVPR, pp. 8183-8192, 8 2018.
18. O. Kupyn, T. Martyniuk, J. Wu and Z. Wang, *"Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better",* IEEE ICCV, pp. 8877-8886, 2019.
19. Jocher, G., Chaurasia, A., & Qiu, J. (2023). **YOLO by Ultralytics (Version 8.0.0)** [Computer software]. <https://github.com/ultralytics/ultralytics>
20. X. Wang, L. Xie, C. Dong and Y. Shan, *"Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data,"* 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 1905-1914, doi: 10.1109/ICCVW54120.2021.00217.
21. J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool and R. Timofte, *"SwinIR: Image Restoration Using Swin Transformer,"* 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 1833-1844, doi: 10.1109/ICCVW54120.2021.00210.
22. Wang, X. et al. (2019). ESRGAN: *“Enhanced Super-Resolution Generative Adversarial Networks”*. In: Leal-Taixé, L., Roth, S. (eds) Computer Vision – ECCV 2018 Workshops. ECCV 2018. Lecture Notes in Computer Science(), vol 11133. Springer, Cham. <https://doi.org/10.1007/978-3-030-11021-5_5>.
23. X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li and F. Huang, *"Real-World Super-Resolution via Kernel Estimation and Noise Injection,"* 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1914-1923, doi: 10.1109/CVPRW50498.2020.00241.
24. Zhang, K. et al. *“Designing a Practical Degradation Model for Deep Blind Image Super-Resolution.”* 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021): 4771-4780.