Terakreditasi Peringkat 4
Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi, No. 225/E/KPT/2022
Masa berlaku dari Volume 8 Nomor 1 Tahun 2022 sampai dengan Volume 12 Nomor 2 Tahun 2026

Published online at: **https://ejournal.raharja.ac.id/**

JOURNAL CERITA:
Creative Education of Research in Information Technology and Artificial informatics

Vol. 12 No. 2 (2026) 207-219 e-ISSN: 2655 - 2574

Fine-tuning Transformer Phi-3 Mini dengan *QLoRA* untuk *Article Generator* Sumber Daya Terbatas

Abdul Hamid Arribathi[1], Dafa Al Farezi[*2]
[1]Magister Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Raharja, Kota Tangerang, Indonesia
[2]Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Raharja, Kota Tangerang, Indonesia
Email: abdulhamid@raharja.info[1]; **dafa.al@raharja.info[*2]**

Arribathi, A. H., & Al Farezi, D. (2026). *Fine-tuning Transformer Phi-3 Mini* dengan *QLoRA* untuk *Article Generator* Sumber Daya Terbatas. *Journal Cerita: Creative Education of Research in Information Technology and Artificial Informatics*, 12(2), 207-219

DOI: https://doi.org/10.33050/cerita.v12i2.4281

ABSTRAK

Perkembangan *Large Language Models (LLMs)* berbasis Transformer telah mendorong kemajuan signifikan dalam tugas generasi teks, termasuk generasi artikel berita. Namun, penerapan *LLM* umumnya memerlukan sumber daya komputasi yang besar sehingga kurang inklusif bagi lingkungan dengan keterbatasan infrastruktur. Penelitian ini bertujuan mengeksplorasi penerapan model *Transformer* ringan *Phi-3 Mini* yang diadaptasi menggunakan pendekatan *Quantized Low-Rank Adaptation (QLoRA)* untuk tugas generasi artikel berita pada lingkungan komputasi *low-resource*. Penelitian ini menggunakan *metode experimental research* dengan melakukan *fine-tuning* model pada dataset berita terbuka berformat terstruktur, serta mengevaluasi keluaran model. Seluruh eksperimen dilakukan menggunakan *Google colab free tier* dengan *GPU NVIDIA T4* tanpa infrastruktur berbayar. Hasil eksperimen menunjukkan bahwa model mampu menghasilkan artikel berita baru berdasarkan judul yang diberikan, namun kualitas teks masih terbatas dari sisi koherensi, kelengkapan informasi, dan ketepatan konteks. Temuan ini mengindikasikan bahwa meskipun kombinasi *Small Language Model* dan *QLoRA* memungkinkan eksperimen *LLM* pada lingkungan *low-resource*, tantangan kualitas generasi teks masih perlu ditangani melalui optimasi data, konfigurasi pelatihan, dan strategi fine-tuning lanjutan.

Kata kunci: model bahasa ringan, *Transformer Phi-3 Mini*, *QLoRA fine-tuning*, generasi artikel berita, komputasi sumber daya terbatas

Received: 20-02-2026 | Accepted: 20-06-2026 | Published Online: 08-08-2026

ABSTRACT

The rapid development of Transformer-based Large Language Models (LLMs) has led to significant progress in text generation tasks, including news article generation. However, the deployment of LLMs typically requires substantial computational resources, limiting their applicability in low-resource environments. This study investigates the use of a lightweight Transformer model, Phi-3 Mini, adapted using Quantized Low-Rank Adaptation (QLoRA) for news article generation under constrained computational settings. The research adopts an experimental research methodology by fine-tuning the model on an open, structured news dataset and evaluating the generated outputs. All experiments were conducted using the Google colab free tier with an NVIDIA T4 GPU, without relying on paid infrastructure. The experimental results show that the fine-tuned model is capable of generating new news articles based on given titles; however, the quality of the generated text remains limited in terms of coherence, contextual relevance, and informational completeness. These findings indicate that while the combination of Small Language Models and QLoRA enables feasible LLM experimentation in low-resource environments, further optimization of data, training configurations, and fine-tuning strategies is required to improve text generation quality.

Keywords: *small language models, Transformer Phi-3 Mini, QLoRA fine-tuning, news article generator, low-resource computing*

I. PENDAHULUAN

Natural Language Processing (*NLP*), khususnya *natural language generation* (*NLG*), mengalami perkembangan pesat. Dong et al. menunjukkan bahwa tugas-tugas *NLG* seperti *summarization*, dialog, dan *style transfer* kini didominasi oleh pendekatan *deep learning* berbasis *encoder–decoder* dengan mekanisme *self-attention*, menggantikan metode simbolik dan statistik konvensional (Dong et al., 2023). Sejalan dengan itu, Li et al. menegaskan bahwa *pretrained language Models* (*PLMs*) berbasis *Transformer*, seperti BERT, GPT, T5, dan BART, telah menjadi fondasi utama generasi teks modern melalui skema pre*training* berskala besar dan *fine-tuning* pada tugas spesifik, sehingga menghasilkan teks yang lebih fasih, koheren, dan relevan secara kontekstual (J. Li et al., 2021).

Perkembangan tersebut mendorong lahirnya *Large Language Models* (*LLMs*) dengan skala parameter dan data pelatihan yang sangat besar. Zhao et al. menjelaskan bahwa *LLM* modern memiliki puluhan hingga ratusan miliar parameter dan mengikuti *scaling laws* yang berkorelasi dengan peningkatan kinerja pada berbagai tugas *NLP* (Zhao et al., 2025). Minaee et al. juga menyoroti kemampuan *emergent* pada model seperti GPT, PaLM, dan LLaMA, termasuk *in-context learning* dan *instruction following*. Namun, hal ini diiringi dengan kebutuhan komputasi yang tinggi, biaya pelatihan besar, serta ketergantungan pada infrastruktur komputasi berskala klaster (Minaee et al., 2025).

Selain tantangan teknis, isu efisiensi dan keberlanjutan menjadi perhatian penting. Bender et al. mengkritisi paradigma "*bigger is better*" dengan menekankan dampak lingkungan, biaya ekonomi, keterbatasan akses, serta risiko amplifikasi bias dari data pelatihan berskala web (Bender et al., 2021). Oleh karena itu, efisiensi *LLM* tidak hanya berkaitan dengan optimasi komputasi, tetapi juga mencakup aspek keberlanjutan, keadilan akses, dan etika riset *NLP*.

Riset terkini berfokus pada *Small Language Models* (*SLMs*) dan teknik efisiensi model. Menyajikan survei komprehensif yang menekankan bahwa model berparameter lebih kecil dapat dioptimalkan melalui arsitektur ringan, strategi pelatihan efisien, serta teknik kompresi seperti *pruning, quantization, dan knowledge distillation*. Selain itu, pendekatan *parameter-efficient fine-tuning* (*PEFT*) memungkinkan *SLM* mencapai kinerja mendekati *LLM* besar dengan kebutuhan memori dan komputasi yang jauh lebih rendah.

Berdasarkan kondisi tersebut, terdapat kesenjangan penelitian (*research gap*) antara kemampuan *LLM* dan kebutuhan praktis akan sistem generasi teks yang efisien dan terjangkau. *LLM* menawarkan kualitas tinggi tetapi sulit diakses, sementara *SLM* (*Small Language Models*) dan teknik *fine-tuning* hemat parameter masih memerlukan kajian pada domain seperti generasi teks. Oleh karena itu, penelitian ini bertujuan mengembangkan dan mengevaluasi

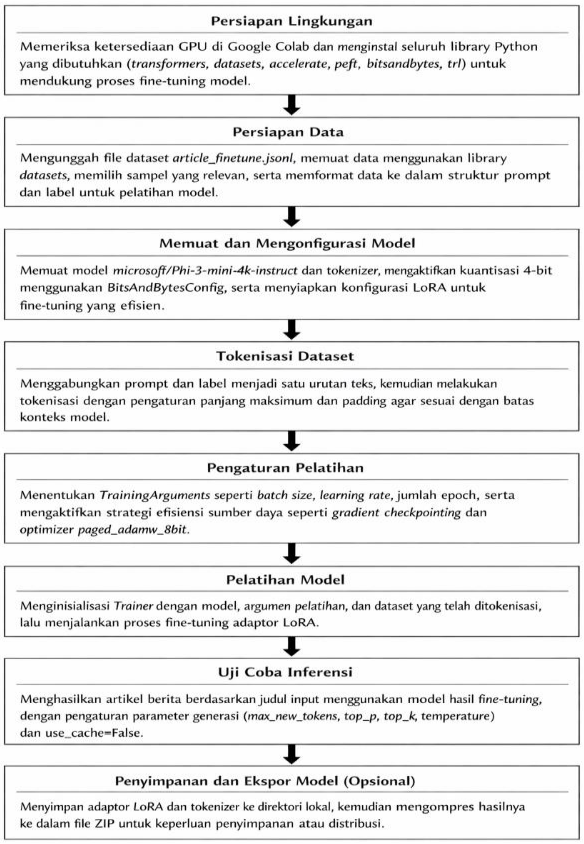
sistem generator artikel berita berbasis *Transformer Phi-3 Mini* yang di-*fine-tune* menggunakan *QLoRA* pada lingkungan komputasi rendah, guna mempelajari apakah model *Generative AI* berbasis *NLP* dapat dicapai tanpa bergantung pada *LLM* maupun infrastruktur mahal.Semua makalah ditulis dalam bahasa bahasa Indonesia. Panduan penulisan ini dilengkapi dengan deskripsi huruf, spasi, dan informasi lainnya yang berhubungan dengan penulisan makalah anda.

II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *machine learning* terapan (*applied machine learning experiments*). Dalam penelitian ini, model bahasa berbasis *Transformer* diimplementasikan, di-*fine-tune*, dan dievaluasi secara empiris pada tugas generasi artikel berita berbahasa Indonesia.

Rancangan eksperimen mengikuti praktik umum dalam penelitian generasi teks berbasis *Transformer*, yang mencakup pemilihan dan praproses dataset artikel, penyesuaian model pra-latih melalui *fine-tuning*, serta evaluasi terhadap *output* model. Pendekatan serupa telah digunakan dalam beberapa penelitian generasi headline dan teks berita, di mana model dilatih secara *end-to-end* pada pasangan artikel–judul dan dievaluasi. (Z. Li et al., 2022)

A. Alur Penelitian



Gambar 1. Alur Penelitian

B. Dataset

Penelitian ini menggunakan dataset terbuka dari *Kaggle* berjudul "The Indonesia Election *News* Dataset 2024 (Berita Pemilu 2024) in detik.com", yang disediakan dalam format CSV. Dataset ini berisi 22.094 pasangan data dan dimanfaatkan sebagai korpus utama untuk pelatihan dan evaluasi model.
Dua kolom utama yang digunakan:

1) ***title***: judul berita, digunakan sebagai konteks atau kondisi input.
2) ***article_text***: isi lengkap artikel berita, ***digunakan*** sebagai target keluaran model.

Pasangan judul–artikel ini dipetakan ke dalam tugas *sequence-to-sequence*, di mana model diinstruksikan untuk menghasilkan artikel berita lengkap berdasarkan judul yang diberikan.

Tahapan praproses meliputi pembersihan data duplikat dan entri tidak valid, normalisasi ringan teks tanpa mengubah gaya bahasa, pembagian data, validasi, dan pengujian, serta pemotongan panjang artikel agar sesuai dengan batas konteks maksimum model. Konfigurasi ini bertujuan menjaga kualitas data sekaligus menghindari kebocoran informasi antara tahap pelatihan dan evaluasi.

C. Lingkungan Eksperimen (Low-resource Setup)

Seluruh eksperimen dilakukan pada lingkungan sumber daya terbatas, dengan spesifikasi berikut:

1) *Platform*: *Google colab* (*free tier*)
2) GPU: *NVIDIA T4 (satu unit)*
3) *RAM: RAM standar Colab*
4) *Perangkat lunak utama: PyTorch, Hugging Face Transformers, PEFT, serta pustaka* pendukung pemrosesan data

Penelitian ini tidak menggunakan infrastruktur berbayar atau klaster komputasi khusus; seluruh eksperimen dijalankan dalam *Google colab* dengan satu *GPU T4*. Konfigurasi ini merepresentasikan skenario *low-resource* yang umum dijumpai dalam penelitian terapan dan pengembangan prototipe.

Studi terdahulu menunjukkan bahwa teknik *parameter-efficient fine-tuning* (*PEFT*) memungkinkan pelatihan model *Transformer* yang stabil dan efisien. Selain itu, evaluasi model generatif *Transformer* pada GPU NVIDIA T4 tunggal juga telah dilaporkan sebagai praktik yang sah dan representatif untuk tugas *NLP* skala menengah (Kim et al., 2023).

D. Arsitektur Model dan Strategi Fine-tuning

1. Pemilihan Model *Phi-3 Mini*

Model dasar yang digunakan adalah *Phi-3 Mini*, sebuah model *decoder-only Transformer* berukuran relatif kecil yang telah melalui proses *instruction tuning*. Survei terbaru mendefinisikan *Small Language Models* (*SLMs*) sebagai model *Transformer* dengan ukuran sekitar 100 juta hingga beberapa miliar parameter yang dirancang untuk *deployment* efisien pada lingkungan terbatas (Z. Lu et al., 2025). Dalam survei tersebut, model *Phi* diidentifikasi sebagai *SLM* dengan kinerja pemahaman dan penalaran yang kompetitif, sekaligus hemat sumber daya.

Pemilihan *Phi-3 Mini* didasarkan pada beberapa pertimbangan:

1) Ukuran model yang relatif kecil memungkinkan pelatihan dan inferensi pada satu GPU T4. (Z. Lu et al., 2025)
2) Sifat instruction-tuned memudahkan adaptasi ke tugas generasi artikel berita berbasis *prompt*.
3) Keselarasan dengan tren riset terkini yang menekankan efisiensi model tanpa ketergantungan pada *LLM* berskala besar. (Z. Lu et al., 2025)

Selain itu, studi tentang kompresi dan deployment *LLM* dalam sumber daya terbatas menekankan bahwa model berukuran kecil dan terbuka cocok untuk *fine-tuning* efisien pada infrastruktur terbatas. (Girija et al., 2025)

2. Strategi *Fine-tuning* dengan *QloRA*

Untuk mengadaptasi *Phi-3 Mini* pada tugas generasi artikel berita, penelitian ini menerapkan *Quantized Low-Rank Adaptation* (*QLoRA*) sebagai *parameter-efficient fine-tuning*. *QLoRA* mengombinasikan kuantisasi bobot model pra-latih ke presisi rendah dengan penyisipan adaptor *LoRA* ber-rank rendah, sehingga hanya sebagian kecil parameter yang diperbarui selama pelatihan.

Pendekatan ini dikatakan efektif dalam menurunkan kebutuhan memori dan komputasi tanpa penurunan kinerja yang signifikan, khususnya dengan data dan sumber daya terbatas (Girija et al., 2025; Nwaiwu, 2025). Dengan konfigurasi *QLoRA*, proses *fine-tuning* dalam penelitian ini diharapkan dapat berjalan pada satu GPU T4 *Colab free tier* tanpa kehabisan memori, sekaligus mempertahankan stabilitas dan kualitas model.

III. HASIL DAN PEMBAHASAN

A. Set up environment

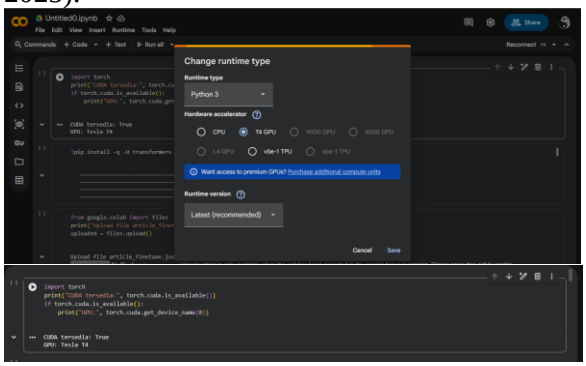
1. Pengecekan Ketersediaan *GPU NVIDIA T4* pada *Google colab*

Pada awal sesi *Google colab* dilakukan pengecekan tipe GPU untuk memastikan bahwa lingkungan eksekusi menggunakan NVIDIA T4, sehingga kapasitas memori dan kemampuan akselerasi komputasi sesuai dengan kebutuhan *fine-tuning* model *Phi-3 Mini* menggunakan *QLoRA*.

GPU NVIDIA T4 dipilih karena diklaim relevan untuk beban kerja *deep learning* modern, sehingga sesuai untuk eksperimen *Transformer*

berukuran kecil hingga menengah pada lingkungan sumber daya terbatas (Ali et al., 2023; Desislavov et al., 2023).

Beberapa studi juga menunjukkan bahwa NVIDIA T4 banyak digunakan pada skenario GPU-as-a-Service karena mampu memberikan performa yang memadai dengan konsumsi daya relatif rendah (Wang et al., 2021). Dalam konteks penelitian ini, satu GPU T4 sudah mencukupi untuk memuat model *Phi-3 Mini* yang dikompresi dengan *QLoRA* serta menjalankan proses *fine-tuning* dan inferensi, tanpa memerlukan infrastruktur *multi-GPU* atau layanan berbayar (Girija et al., 2025; Nwaiwu, 2025).



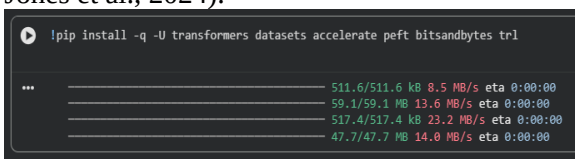
Gambar 1. GPU Google colab

2. Instalasi Library Python

Lingkungan eksperimen dikonfigurasi menggunakan ekosistem Python berbasis open-source yang umum digunakan dalam penelitian *NLP*, yaitu ***Transformers, datasets, accelerate, PEFT, bitsandbytes***, dan **trl**. Library ***Transformers*** dan ***datasets*** digunakan untuk memuat model *Transformer* pra-latih serta mengelola dataset secara efisien dalam eksperimen generasi teks (Auriemma et al., 2023; Lahiri et al., 2025).

1) ***Accelerate*** dimanfaatkan untuk mengelola pemetaan perangkat dan mixed-precision *training* secara transparan, sehingga mempermudah pelaksanaan *fine-tuning* pada GPU tunggal dengan memori terbatas serta menjaga reprodusibilitas (Lahiri et al., 2025; Tucudean et al., 2024).
2) ***PEFT*** digunakan untuk menerapkan metode *parameter-efficient fine-tuning* seperti *LoRA* dan *QLoRA*, yang memungkinkan adaptasi model dengan pembaruan parameter minimal dan jejak memori yang lebih rendah (Girija et al., 2025; Nwaiwu, 2025).
3) ***Bitsandbytes*** (untuk mendukung efisiensi memori), digunakan dalam kuantisasi 8-bit dan 4-bit, sehingga model *Phi-3 Mini* dapat dilatih pada GPU T4 tanpa melampaui kapasitas memori. (Girija et al., 2025)
4) ***Trl*** dimanfaatkan sebagai kerangka pelatihan tingkat tinggi yang terintegrasi dengan *Transformers* dan *PEFT*, sehingga menyederhanakan pengelolaan proses supervised *fine-tuning* pada tugas generasi teks (Belanec et al., 2025).

Penggunaan kombinasi library dalam ekosistem *Hugging Face* ini sejalan dengan praktik standar dalam literatur, serta memperkuat aspek efisiensi, reprodusibilitas, dan komparabilitas eksperimen pada penelitian *Transformer* dan *LLM* (Auriemma et al., 2023; Jones et al., 2024).



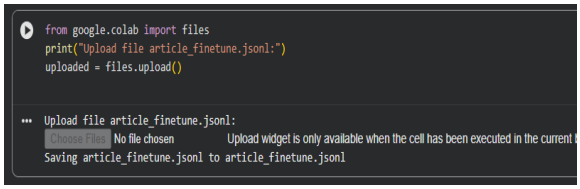
Gambar 2. Kode instalasi library

B. Pra-Processing Data

1. Upload dan Konversi Dataset dari *CSV* ke *JSON*

Dataset yang tersedia dalam format CSV dikonversi ke format JSON sebelum digunakan dalam *fine-tuning*. Dokumentasi resmi *Hugging Face Datasets* menekankan bahwa format JSON dengan beberapa objek JSON per baris (JSONL) merupakan bentuk yang paling efisien untuk merepresentasikan baris-baris data individual, karena memudahkan pemetaan langsung ke features bertipe string, label, atau struktur bertingkat pada saat pemanggilan load_dataset("json", ...).

Implikasi praktis bagi eksperimen ini adalah pipeline *fine-tuning* dapat memanfaatkan pemetaan kolom (***title, article_text***) ke *field* JSON secara eksplisit, sehingga proses pemuatan, *sharding*, dan streaming data oleh *Hugging Face Datasets* menjadi lebih stabil dan hemat memori dibanding mempertahankan format CSV yang lebih kaku untuk skenario data *semi-terstruktur*.



Gambar 3. Kode upload dataset

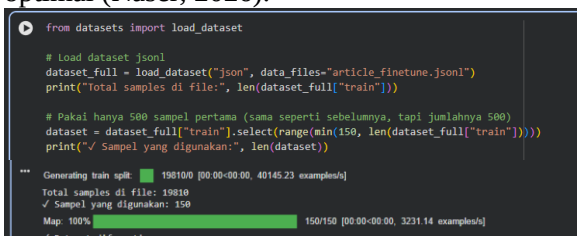
2. Pengambilan Sampel Data

Sebanyak 150 sampel dipilih dari total artikel berita sebagai subset untuk tahap

eksperimen awal (*pilot experiment*). Pemilihan subset ini bertujuan menyeimbangkan keterbatasan sumber daya komputasi (GPU NVIDIA T4 tunggal) dengan kebutuhan eksplorasi desain eksperimen, seperti skema *prompt*, konfigurasi *QLoRA*, dan stabilitas proses pelatihan.

Penggunaan dataset *pilot* berukuran kecil sejalan dengan temuan literatur yang menunjukkan bahwa perilaku dan tren performa model pada skala besar dapat diperkirakan melalui eksperimen pada subset terbatas, sehingga perancangan menjadi lebih efisien tanpa kehilangan representativitas (Naser, 2026).

Dalam penelitian ini, *fine-tuning* menggunakan 150 sampel sebagai tahap eksploratif untuk mengevaluasi stabilitas pelatihan, indikasi konvergensi, serta potensi permasalahan seperti *overfitting* atau ketidakstabilan gradien, sebelum melanjutkan ke skala data yang lebih besar, sehingga sumber daya komputasi dapat digunakan secara lebih optimal (Naser, 2026).

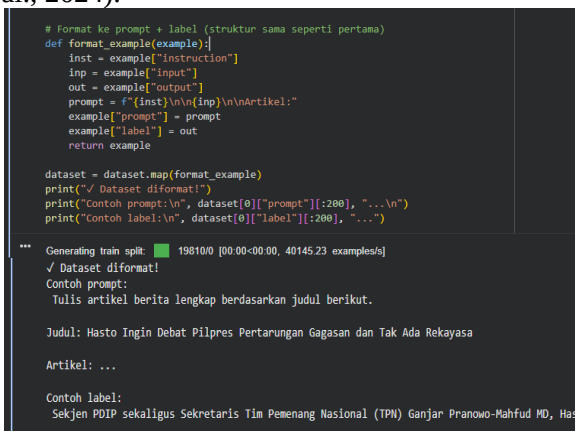


Gambar 4. Kode pengambilan sampel data

3. Melakukan Format Data

Sampel terpilih diformat ke pasangan *prompt-label*, di mana *prompt* berisi instruksi generasi yang digabungkan dengan judul artikel, sedangkan label berisi teks artikel sebagai target keluaran. Pemisahan ini memungkinkan model mempelajari pemetaan dari instruksi bahasa alami ke teks hasil. Format tersebut sejalan dengan praktik *instruction-finetuning* pada model generatif modern, yang merepresentasikan setiap contoh sebagai pasangan instruksi dan respons untuk meningkatkan kemampuan generalisasi pada skenario *zero-shot* dan *few-shot* (Won Chung et al., 2024). Dokumentasi resmi *Transformers* juga menekankan pentingnya pemisahan peran user (*prompt*) dan assistant (label) dalam chat template agar struktur data konsisten dengan pola pelatihan model *instruction-tuned*.

Dalam penelitian ini, struktur *prompt–label* memastikan bahwa *Phi-3 Mini* dapat mempelajari hubungan eksplisit antara instruksi generasi berbasis judul, sehingga pendekatan *fine-tuning* yang digunakan selaras dengan paradigma pengembangan model generatif kontemporer**Error! Bookmark not defined.** (Won Chung et al., 2024).



Gambar 5. Kode formating dataset

C. Set up Library Model Phi-3 Mini

Tahap awal pengembangan melibatkan pemanggilan model dasar ***microsoft/Phi-3-mini-4k-instruct*** beserta *tokenizer* resminya melalui framework *Transformers*. Model ini merupakan *Transformer* dekoder dengan 3,8B parameter yang secara eksplisit dirancang sebagai model *instruction-tuned* dengan panjang konteks maksimum 4.096 token, sehingga konfigurasi awal perlu diselaraskan. *Tokenizer* dikonfigurasi untuk:

1) Menggunakan skema ***padding*** dan ***truncation*** yang konsisten agar seluruh pasangan *prompt*–label dapat diproyeksikan ke tensor berukuran tetap, sesuai rekomendasi dokumentasi *Transformers* mengenai pengaturan ***padding, truncation,*** dan ***max_length***.
2) Memastikan bahwa panjang sekuens tidak melampaui kapasitas *4K* model *Phi-3*, sehingga mengurangi risiko pemotongan informasi penting secara tak terkontrol.

Gambar 6. Kode import *model* dan tokenizer

Konfigurasi presisi dan kuantisasi disesuaikan dengan skema *QLoRA*, yaitu memuat bobot dasar dalam presisi rendah (misalnya 4-bit) dan mengaktifkan *mixed-precision* (FP16/BF16) pada aktivasi. Pengaturan ini dimaksudkan untuk:

1) Menurunkan jejak memori GPU sehingga model *Phi-3 Mini* tetap dapat dilatih pada GPU T4,
2) Menjaga stabilitas numerik selama *fine-tuning* dengan tetap mempertahankan presisi lebih tinggi pada jalur komputasi yang sensitif (misalnya gradien dan adaptor *LoRA*).

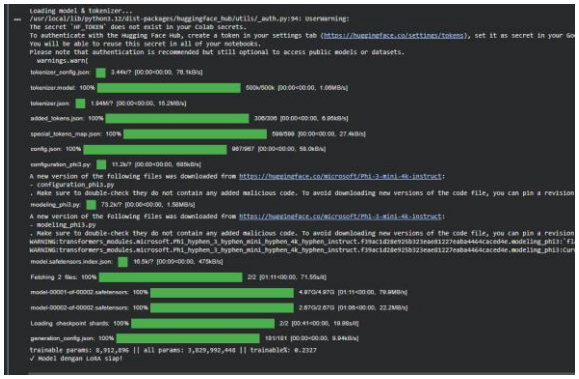
Mode eksekusi model dibedakan secara eksplisit antara fase pelatihan (mengaktifkan *gradient computation*, menghubungkan adaptor *LoRA*/*QLoRA*, dan mengatur *dropout*) dan fase inferensi (menonaktifkan gradien, mengunci adaptor, dan mengoptimalkan *generation*). Pemisahan ini memastikan bahwa konfigurasi yang digunakan selama *fine-tuning* tidak secara tidak sengaja terbawa ke tahap evaluasi, yang dapat menyebabkan ketidakkonsistenan hasil.

```
model = prepare_model_for_kbit_training(model)

lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM",
    target_modules=["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"],
)

model = get_peft_model(model, lora_config)
model.print_trainable_parameters()
print("✓ Model dengan LoRA siap!")
```

Gambar 7. Kode konfirugrasi Model untuk QLoRA

Konfigurasi juga mencakup penyesuaian agar kompatibel dengan *fine-tuning* berbasis adaptor, seperti:

1) Pemetaan lapisan-lapisan kunci sebagai target injeksi *LoRA*,
2) Penetapan parameter model utama sebagai *frozen* dan hanya mengizinkan pembaruan pada parameter adaptor. Hal ini selaras dengan praktik *parameter-efficient fine-tuning* yang menekankan pembatasan untuk meningkatkan stabilitas pelatihan dan efisiensi komputasi.
3) Secara keseluruhan, konfigurasi awal model *Phi-3 Mini* diperlukan untuk memastikan kesesuaian antara arsitektur dengan skema *fine-tuning QLoRA* yang digunakan, sehingga proses pelatihan dapat berlangsung stabil, efisien secara memori, dan tetap mempertahankan kemampuan mengikuti instruksi yang telah dibentuk pada tahap *instruction-tuning*.



Gambar 8. Output konfigurasi Model berhasil

D. Tokenisasi Dataset

Dataset hasil *pra-processing* kemudian ditokenisasi menggunakan *tokenizer* bawaan *Transformer Phi-3 Mini*. Setiap pasangan *prompt*-label diubah dari teks mentah menjadi token numerik (*input IDs*) sesuai dengan kosakata dan skema sub-kata yang digunakan oleh model.

Tokenisasi berfungsi sebagai penghubung antara teks manusia dan representasi diskret yang dapat diproses oleh mekanisme *self-attention* pada *Transformer*, dan sebagai *indeks embedding* selama proses pelatihan dan inferensi (Tucudean et al., 2024). Literatur menunjukkan bahwa tokenisasi yang konsisten penting untuk menjaga keseragaman representasi teks, mengendalikan panjang sekuens, serta meningkatkan efisiensi komputasi dan stabilitas pelatihan model berbasis perhatian (Edo Dotan et al., 2024; Tucudean et al., 2024).

Dalam *fine-tuning Phi-3 Mini* dengan *QLoRA*, penggunaan *tokenizer* pra-latih memastikan kesesuaian distribusi token dengan parameter model yang telah dipelajari sebelumnya, sehingga kualitas generasi teks dan pemanfaatan konteks dapat tetap optimal.

```
max_length = 1024

def tokenize_function(examples):
    # Sama persis: gabung prompt + label
    full_texts = [p + " " + l for p, l in zip(examples["prompt"], examples["label"])]
    tokenized = tokenizer(
        full_texts,
        max_length=max_length,
        truncation=True,
        padding="max_length",
        return_tensors=None,
    )
    tokenized["labels"] = tokenized["input_ids"].copy()
    return tokenized

print("Tokenisasi dataset...")
tokenized_dataset = dataset.map(
    tokenize_function,
    batched=True,
    remove_columns=dataset.column_names,
    desc="Tokenizing"
)

print("✓ Tokenisasi selesai!")
print("Samples:", len(tokenized_dataset))
print("Kolom:", tokenized_dataset.column_names)
```

```
Tokenisasi dataset...
Tokenizing: 100% 150/150 [00:00<00:00, 440.87 examples/s]
✓ Tokenisasi selesai!
Samples: 150
Kolom: ['input_ids', 'attention_mask', 'labels']
```

Gambar 9. Tokenisasi dataset

E. Fine-tuning dengan QLoRA

1. Konfigurasi *TrainingArguments*

Sebelum proses pelatihan, objek *TrainingArguments* dikonfigurasi untuk mengendalikan perilaku *training loop*, termasuk pengaturan ukuran ***batch, learning rate,* jumlah *epoch***, strategi penyimpanan *checkpoint*, serta opsi *mixed-precision* dan akumulasi gradien. Dokumentasi resmi *Transformers* menyatakan bahwa *TrainingArguments* berperan sebagai pusat pengaturan hiperparameter dan strategi pelatihan, termasuk manajemen memori dan pelaporan metrik.

Pengaturan *TrainingArguments* penting untuk menjaga stabilitas pelatihan dan efisiensi komputasi, khususnya pada *GPU* kelas menengah seperti *NVIDIA T4*. Studi hiperparameter menunjukkan bahwa konfigurasi *learning rate*, ukuran *batch*, dan jumlah *epoch* berpengaruh langsung terhadap konvergensi dan kinerja model, sementara pengaturan yang tidak tepat dapat menyebabkan degradasi performa meskipun arsitektur model tidak berubah (Kumar et al., 2024; Saputra et al., 2025).

Dalam penelitian ini, *TrainingArguments* disesuaikan untuk menyeimbangkan stabilitas pembaruan adaptor *QLoRA* dan keterbatasan memori GPU, melalui pemilihan ukuran *batch* moderat, strategi *checkpointing* yang hemat ruang, serta *logging* terjadwal. Konfigurasi ini mendukung reprodusibilitas eksperimen dan mempermudah analisis hasil pada sesi komputasi yang berbeda.**Error! Bookmark not defined.**

```
from transformers import TrainingArguments

training_args = TrainingArguments(
    output_dir="./phi3-article-qlora-200",  # ubah nama folder
    per_device_train_batch_size=1,          # tetap 1
    gradient_accumulation_steps=2,          # dari 4 jadi 2 (effective batch=2)
    learning_rate=5e-5,                     # lebih kecil lagi (dari 1e-4)
    num_train_epochs=3,                     # dari 2 jadi 3 epoch
    logging_steps=5,                        # log lebih sering (dari 10)
    save_steps=50,                          # checkpoint tiap 50 step
    bf16=True,
    optim="paged_adamw_8bit",
    lr_scheduler_type="cosine",
    warmup_ratio=0.1,                       # warmup lebih panjang (dari 0.05)
    weight_decay=0.0,
    report_to=[],
    save_total_limit=1,
    remove_unused_columns=False,
    gradient_checkpointing=True,
    max_grad_norm=1.0,
)

est_steps = len(tokenized_dataset) // (
    training_args.per_device_train_batch_size * training_args.gradient_accumulation_steps
) * training_args.num_train_epochs

print("✓ Training arguments siap!")
print("Samples:", len(tokenized_dataset))
print("Estimasi total steps:", est_steps)

✓ Training arguments siap!
Samples: 150
Estimasi total steps: 225
```

Gambar 10. Konfigurasi TrainingArguments

2. *Training* Model dengan *Trainer*

Proses *fine-tuning* dilaksanakan menggunakan *methode Trainer* dari *Hugging Face Transformers*, yang menyediakan abstraksi *training loop* lengkap untuk model *Transformer*, termasuk perhitungan loss, propagasi balik gradien, pembaruan bobot, evaluasi berkala, dan penyimpanan *checkpoint*. Dalam skema ini, *Trainer* diinisialisasi dengan:

1) model *Phi-3 Mini* yang telah dikonfigurasi dan diperkaya adaptor *QLoRA*,
2) objek *TrainingArguments* yang memuat seluruh pengaturan pelatihan,
3) serta dataset yang telah ditokenisasi beserta data collator yang sesuai.

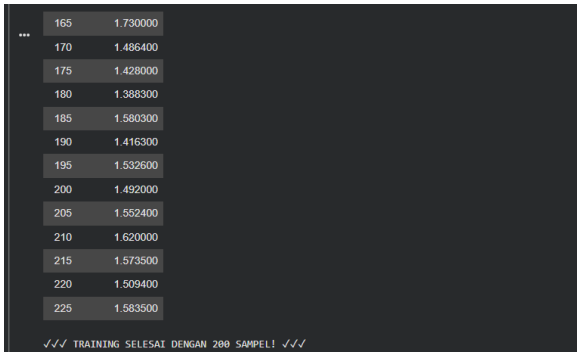
```
from transformers import Trainer

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
)

print("✓ Trainer siap, mulai training...\n")
trainer.train()
print("\n✓✓✓ TRAINING SELESAI DENGAN 200 SAMPEL! ✓✓✓")

✓ Trainer siap, mulai training...

WARNING:transformers_modules.microsoft.Phi_hyphen_3_hyphen_mini_hyphen_4k_hyphen_instruct.f3
WARNING:transformers_modules.microsoft.Phi_hyphen_3_hyphen_mini_hyphen_4k_hyphen_instruct.f3
[225/225 1:37:26, Epoch 3/3]

Step  Training Loss
5     4.926800
10    4.005400
15    2.662400
20    4.113200
```

Gambar 11. Fine-tuning model dengan Training methode

Pendekatan ini menghasilkan integrasi yang terstruktur antara model, data, dan konfigurasi pelatihan tanpa perlu menulis ulang *training loop* manual, sebagaimana direkomendasikan dalam panduan resmi *Transformers*.

Penggunaan *Trainer* dipilih karena memberikan konsistensi dan keandalan proses pelatihan, termasuk dukungan untuk fitur penting seperti ***gradient accumulation, mixed-precision training***, evaluasi terjadwal, dan *logging* terstandardisasi.

Implikasinya bagi penelitian ini adalah bahwa proses *fine-tuning Phi-3 Mini* dengan *QLoRA* dapat dijalankan secara lebih terukur dan mudah direplikasi: konfigurasi eksperimen terdokumentasi secara eksplisit melalui *TrainingArguments*, sedangkan *Trainer* memastikan bahwa prosedur pelatihan yang dijalankan konsisten, sehingga meminimalkan kesalahan implementasi manual dan memfokuskan perhatian pada desain tugas generasi berita dan analisis hasil.



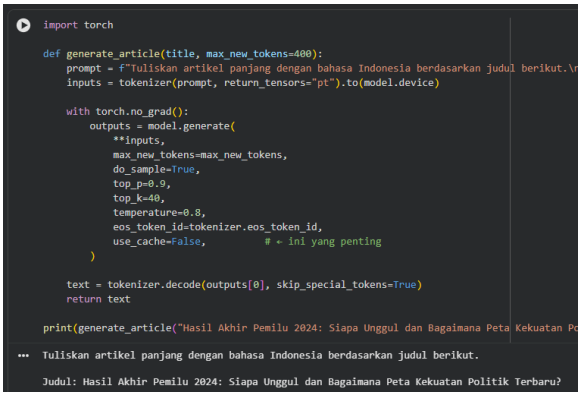
Gambar 12. Fine-tuning berhasil

F. Hasil Uji Coba dan Evaluasi Model

1. Inferensi dan Pengaturan *Parameter*

Setelah proses *fine-tuning*, model *Phi-3 Mini* diuji melalui fungsi generasi teks yang menerima judul artikel sebagai *prompt* dan menghasilkan artikel sebagai keluaran. Pada tahap inferensi, digunakan beberapa parameter utama untuk mengontrol karakteristik teks yang dihasilkan.

1) Parameter ***max_new_tokens*** digunakan untuk membatasi panjang teks agar sesuai dengan rentang artikel berita pada data pelatihan.
2) ***do_sample=True*** dikombinasi dengan top-p dan top-k sampling untuk mengaktifkan dekoding stokastik, sehingga tercapai keseimbangan antara koherensi dan keragaman teks (Franceschelli & Musolesi, 2026; Gian Wiher et al., 2022).
3) ***temperature*** digunakan untuk mengatur tingkat determinisme keluaran, nilai rendah berarti stabil, sedangkan nilai tinggi meningkatkan variasi namun berisiko menurunkan koherensi.
4) ***use_cache=False*** diterapkan untuk menjaga kompatibilitas antara adaptor *low-rank* dan skema kuantisasi, serta mencegah potensi kegagalan eksekusi akibat keterbatasan memori GPU.



Gambar 13. Konfigurasi parameter untuk menggunakan model

Pengaturan ini penting untuk menghindari fenomena *text degeneration* seperti repetisi berlebihan atau deviasi topik, yang diketahui sensitif terhadap panjang sekuens dan strategi sampling (Franceschelli & Musolesi, 2026; Gian Wiher et al., 2022; Nan et al., 2022).

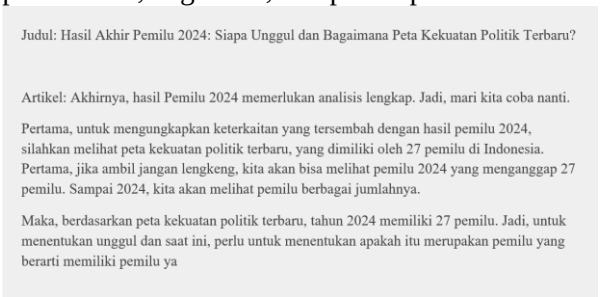
Dalam konteks *low-resource*, kombinasi parameter yang konservatif namun tetap stokastik membantu memitigasi keterbatasan model terkuantisasi, sehingga model tetap mampu menghasilkan variasi narasi berita yang masuk akal tanpa mengorbankan stabilitas keluaran (Z. Lu et al., 2025).

2. Analisis Kualitas Hasil Generasi Teks

Secara fungsional, model *Phi-3 Mini* yang telah di-fine-tune mampu menghasilkan artikel baru yang secara permukaan relevan dengan judul *prompt* yang diberikan. Contohnya, untuk judul:

> "Hasil Akhir Pemilu 2024: Siapa Unggul dan Bagaimana Peta Kekuatan Politik Terbaru?"

Model menghasilkan teks yang membahas hasil pemilu, dinamika koalisi, dan peta kekuatan politik pasca-pemilu. Hal ini menunjukkan bahwa model dapat menangkap topik umum dan menghasilkan struktur artikel dengan pembukaan, bagian isi, dan penutup.



Gambar 14. Contoh teks artikel yang dihasilkan model

Analisis terhadap keluaran model menunjukkan beberapa keterbatasan utama:

1) Narasi yang dihasilkan cenderung umum dan repetitif, dengan pengulangan frasa generik tanpa pengayaan sudut pandang atau detail kontekstual yang memadai.
2) Ditemukan ketidakoherenan semantik, seperti pergeseran topik yang tiba-tiba dan penggabungan informasi yang tidak konsisten secara kronologis.

Keterbatasan tersebut dapat dijelaskan oleh beberapa faktor teknis, seperti:

1) Penggunaan data yang sangat terbatas (150 sampel) membatasi kemampuan model dalam mempelajari variasi gaya dan kedalaman struktur penulisan, sehingga model cenderung mengandalkan pola narasi umum dari tahap pra-latih (Naser, 2026).
2) kapasitas *Phi-3 Mini* sebagai *Small Language* Model membuat pemodelan dependensi jangka panjang dan konsistensi narasi pada teks panjang lebih rentan dibandingkan *LLM* (W. Lu et al., 2025; Van Nguyen et al., 2025).
3) Faktor lain berasal dari kompromi kualitas akibat kuantisasi dan penerapan *PEFT*, yang meskipun efisien secara komputasi, dapat menurunkan stabilitas distribusi probabilitas token pada proses dekoding panjang, sehingga memengaruhi kualitas generasi secara keseluruhan.

Dalam penelitian ini, fine-tuning dilakukan dengan berbagai variasi konfigurasi *TrainingArguments*, seperti peningkatan jumlah *epoch*, penyesuaian *batch* size, serta penggunaan jumlah data sampel yang lebih besar. Namun, proses pelatihan sering terhenti sebelum fine-tuning selesai (runtime disconnected) karena batasan durasi sesi dan memori GPU. Berikut beberapa konfigurasi yang dilakukan dalam beberapa percobaan dan dampaknya pada pemrosesan model:

Tabel 1. Tabel Konfigurasi

No.	Sampel	*Batch size*	Akumulasi *gradient*	*Epochs*	Estimasi waktu *Fine-tuning*	Hasil
1	20000	2	4	2	124:35:54 Jam	Runtime disconnect (3:50:00 jam)
2	1000	2	4	2	32:12:52 Jam	Runtime disconnect (1:30:48 jam)
3	1000	2	4	1	30:02:36 Jam	Runtime disconnect (3:42:07 jam)
4	200	2	4	2	30 - 60 Menit	Runtime disconnect (1:47:00 jam)
5	200	1	2	3	30 - 60 Menit	Berhasil (1:37:26 jam)

Meskipun model secara teknis mampu menghasilkan teks yang relevan dengan judul dan menyerupai struktur artikel berita, kualitas narasi yang dihasilkan masih terbatas dari sisi koherensi, variasi leksikal, dan kedalaman faktual. Temuan ini konsisten dengan literatur mengenai tantangan generasi teks pada *SLM*, yang menekankan bahwa optimasi untuk lingkungan sumber daya terbatas sering kali disertai penurunan kualitas pada tugas generasi bebas yang panjang dan kompleks (W. Lu et al., 2025; Van Nguyen et al., 2025).

IV. KESIMPULAN

Hasil penelitian menunjukkan bahwa *fine-tuning* model *Transformer Phi-3 Mini* dengan *QLoRA* dapat dilakukan pada lingkungan komputasi *low-resource* (tanpa infrastruktur berbayar). Eksperimen berhasil dijalankan menggunakan *Google colab* free tier dengan GPU NVIDIA T4, menegaskan bahwa eksperimen berbasis *LLM* tetap feasible meskipun dengan keterbatasan sumber daya.

Model *Phi-3 Mini* yang telah di-fine-tune mampu menghasilkan artikel berita berdasarkan judul yang diberikan, menunjukkan kelayakan pendekatan *Small Language* Model untuk tugas generasi teks. Namun, kualitas keluaran masih terbatas, ditandai oleh repetisi, koherensi yang belum optimal, serta kesesuaian konteks yang belum sepenuhnya tercapai, yang dipengaruhi oleh beberapa faktor seperti: Ukuran dataset yang relatif kecil; Dataset hanya berisi artikel dengan topik pemilu 2024; Durasi pelatihan yang terbatas akibat batasan waktu *Google colab*; Spesifikasi

sumber daya yang rendah; Serta penggunaan konfigurasi pelatihan yang paling rendah agar *fine-tuning* berhasil.

Dengan demikian, penelitian ini menegaskan adanya *trade-off* antara efisiensi komputasi dan kualitas hasil generasi teks pada penerapan *LLM* di lingkungan *low-resource*.

Penelitian selanjutnya dapat dikembangkan ke beberapa arah: 1) Penggunaan dataset yang lebih besar dan beragam untuk meningkatkan pemahaman struktur dan konteks artikel berita; 2) Eksplorasi optimalisasi konfigurasi pelatihan, seperti *learning rate*, jumlah *training* steps, dan strategi sampling, guna meningkatkan kualitas *output*; 3) Evaluasi kinerja model menggunakan metrik kuantitatif seperti ROUGE atau BLEU agar penilaian performa lebih objektif; 4) Pemantauan dan analisis penggunaan RAM dan disk selama proses pelatihan; 5) Perbandingan Phi 3 Mini dengan model *Transformer* ringan lain untuk memperoleh gambaran performa *SLM* yang lebih komprehensif; 6) Pengombinasian *QLoRA* dengan teknik lain, seperti *knowledge distillation* atau *prompt tuning*, berpotensi meningkatkan kualitas generasi teks pada lingkungan sumber daya terbatas.

DAFTAR PUSTAKA

Ali, G., Side, M., Bhalachandra, S., Wright, N. J., & Chen, Y. (2023). Performance-Aware Energy-Efficient GPU Frequency Selection using DNN-based Models. ACM International Conference Proceeding Series, 433–442. https://doi.org/10.1145/3605573.3605600

Auriemma, S., Madeddu, M., Miliani, M., Bondielli, A., Lenci, A., & Passaro, L. (2023). Challenging specialized transformers on zero-shot classification. http://ceur-ws.org

Belanec, R., Srba, I., & Bielikova, M. (2025). PEFT-Factory: Unified Parameter-Efficient Fine-Tuning of Autoregressive Large Language Models. http://arxiv.org/abs/2512.02764

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. https://doi.org/10.1145/3442188.3445922

Desislavov, R., Martínez-Plumed, F., & Hernández-Orallo, J. (2023). Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. Sustainable Computing: Informatics and Systems, 38. https://doi.org/10.1016/j.suscom.2023.100857

Dong, C., Li, Y., Gong, H., Chen, M., Li, J., Shen, Y., & Yang, M. (2023). A Survey of Natural Language Generation. ACM Computing Surveys, 55(8). https://doi.org/10.1145/3554727

Edo Dotan, Gal Jaschek, Tal Pupko, & Yonatan Belinkov. (2024). Effect of tokenization on transformers for biological sequences. Oxford Academic, 40(4). https://doi.org/https://doi.org/10.1093/bioinformatics/btae196

Franceschelli, G., & Musolesi, M. (2026). DiffSampling: Enhancing Diversity and Accuracy in Neural Text Generation. http://arxiv.org/abs/2502.14037

Gian Wiher, Clara Meister, & Ryan Cotterell. (2022). On Decoding Strategies for Neural Text Generators. https://doi.org/10.1162/tacl

Girija, S. S., Arora, L., Kapoor, S., Pradhan, D., Raj, A., & Shetgaonkar, A. (2025). Optimizing LLMs for Resource-Constrained Environments: A Survey of Model Compression Techniques. Proceedings - 2025 IEEE 49th Annual Computers, Software, and Applications Conference, COMPSAC 2025, 1657–1664. https://doi.org/10.1109/COMPSAC65507.2025.00224

Jones, J., Jiang, W., Synovic, N., Thiruvathukal, G., & Davis, J. (2024). What do we know about Hugging Face? A systematic literature review and quantitative validation of qualitative claims. International Symposium on Empirical Software Engineering and Measurement, 13–24. https://doi.org/10.1145/3674805.3686665

Kim, S., Mangalam, K., Moon, S., Malik, J., Mahoney, M. W., Gholami, A., & Keutzer, K. (2023). Speculative Decoding with Big Little Decoder. http://arxiv.org/abs/2302.07863

Kumar, J., Dalal, N., & Sethi, M. (2024).

International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Hyperparameters in Deep Learning: A Comprehensive Review. In Original Research Paper International Journal of Intelligent Systems and Applications in Engineering IJISAE (Vol. 2024, Number 4). www.ijisae.org

Lahiri, A., Shukla, S., Stear, B., Ahooyi, T. M., Beigel, K., Margolskee, E., & Taylor, D. (2025). Benchmarking transformer embedding models for biomedical terminology standardization. Machine Learning with Applications, 21, 100683. https://doi.org/10.1016/j.mlwa.2025.100683

Li, J., Tang, T., Xin Zhao, W., & Wen, J.-R. (2021). Pretrained Language Models for Text Generation: A Survey. https://www.ijcai.org/proceedings/2021/0612.pdf

Li, Z., Wu, J., Miao, J., & Yu, X. (2022). News headline generation based on improved decoder from transformer. Scientific Reports, 12(1). https://doi.org/10.1038/s41598-022-15817-z

Lu, W., Luu, R. K., & Buehler, M. J. (2025). Fine-tuning large language models for domain adaptation: exploration of training strategies, scaling, model merging and synergistic capabilities. Npj Computational Materials, 11(1). https://doi.org/10.1038/s41524-025-01564-y

Lu, Z., Li, X., Cai, D., Yi, R., Liu, F., Zhang, X., Lane, N. D., & Xu, M. (2025). Small Language Models: Survey, Measurements, and Insights. http://arxiv.org/abs/2409.15790

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2025). Large Language Models: A Survey. http://arxiv.org/abs/2402.06196

Nan, L., Jaime, L., Flores, Y., Zhao, Y., Liu, Y., Benson, L., Zou, W., & Radev, D. (2022). R2D2: Robust Data-to-Text with Replacement Detection. https://huggingface.co/Jean-Baptiste/

Naser, M. Z. (2026). A review of machine learning with small and limited data. Journal of Big Data. https://doi.org/10.1186/s40537-025-01346-9

Nwaiwu, S. (2025). Parameter-efficient fine-tuning for low-resource text classification: a comparative study of LoRA, IA3, and ReFT. Frontiers in Big Data, 8. https://doi.org/10.3389/fdata.2025.1677331

Saputra, H., Muchtar, K., Chitraningrum, N., Andria, A., & Febriana, A. (2025). Performance evaluation of hyper-parameter tuning automation in YOLOV8 and YOLO-NAS for corn leaf disease detection. Sinergi (Indonesia), 29(1), 197–206. https://doi.org/10.22441/sinergi.2025.1.018

Tucudean, G., Bucos, M., Dragulescu, B., & Caleanu, C. D. (2024). Natural language processing with transformers: a review. In PeerJ Computer Science (Vol. 10). PeerJ Inc. https://doi.org/10.7717/PEERJ-CS.2222

Van Nguyen, C., Shen, X., Aponte, R., Xia, Y., Basu, S., Hu, Z., Chen, J., Parmar, M., Kunapuli, S., Barrow, J., Wu, J., Singh, A., Wang, Y., Gu, J., K. Ahmed, N., Lipka, N., Zhang, R., Chen, X., Yu, T., … Huu Nguyen, T. (2025). A Survey on Small Language Models. 807–821. https://doi.org/10.26615/978-954-452-098-4-093

Wang, M., Yang, T., Flechas, M. A., Harris, P., Hawks, B., Holzman, B., Knoepfel, K., Krupa, J., Pedro, K., & Tran, N. (2021). GPU-Accelerated Machine Learning Inference as a Service for Computing in Neutrino Experiments. Frontiers in Big Data, 3. https://doi.org/10.3389/fdata.2020.604083

Won Chung, H., Hou, L., Longpre, S., Zoph, B., Tai, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Shane Gu, S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., … Salakhutdinov, R. (2024). Scaling Instruction-Finetuned Language Models. In Journal of Machine Learning Research (Vol. 25). http://jmlr.org/papers/v25/23-0870.html.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J.-R. (2025). A Survey of Large

Language Models.
http://arxiv.org/abs/2303.18223

