

Report - LLM Exercise v13.12.25

trangnt2108

13/12/2025

Contents

Change logs	2
1. Dataset	3
1.1 Raw dataset	3
1.2 Processed dataset	4
1.3 Analysis	5
2. LLM Name Entity Recognition (NER)	5
2.1. Prompt Engineering	5
2.1.1. Prompting Strategies	5
2.1.2. Experimental Results	11
2.2 Retrieval Augmented Generation (RAG)	11
2.2.1. Chunking	11
2.2.2. Retrieval	12
2.2.3. Result	13
2.3 Finetune Model	13
2.3.1. Perpare dataset	13
2.3.2. Train Model	14
2.3.3. Result	15

Change logs

Date	Version	By	Description
09/12/2025	v09.12.25	trangnt2108	Add dataset information and analysis
11/12/2025	v11.12.25	trangnt2108	Add prompt engineer part
12/12/2025	v12.12.25	trangnt2108	Add finetuning model part
13/12/2025	v13.12.25	trangnt2108	Add RAG part

1. Dataset

1.1 Raw dataset

Folder structure:

```
data/vlps_2018_ner
  processed
  raw
    dev
      Doi Song
      Giai tri
      Giao duc
      KH - CN
      Kinh te
      Phap luat
      The gioi
      The thao
      Van hoa
      Xa hoi
    test
      Cong nghe
      Cong nghe Thiet bi - Phan cung
      Cong nghe - Vien thong
      Doi song
      Giai tri
      Giai tri - Am nhac
      Giao duc
      Khoa hoc
      Kinh te
      Kinh te Chung khoan
      Kinh te Kinh doanh
      Kinh te Tai chinh
      Nha dat
      Phap luat
      The gioi
      The thao
      The thao Bong da quoc te
      The thao Quan vot
      Van hoa
      Van hoa Du lich
      Van hoa Nghe thuat
      Xa hoi
      Xa hoi Giao thong
      Xa hoi Moi truong - Khi hau
      Xa hoi Thoi su
    train
      Doi song
      Giai tri
      Giao duc
      KH - CN
      Kinh te
      Phap luat
      The gioi
      The thao
      Van hoa
      Xa hoi
```

Example of a file “.muc”:

Người phụ nữ nguy kịch sau phẫu thuật gọt cằm: Chuyển bệnh nhân qua <ENAMEX TYPE="LOCATION">Singapore</ENAMEX>

Sau thời gian điều trị tại <ENAMEX TYPE="LOCATION">BV Nhân dân 115</ENAMEX>, <ENAMEX TYPE="LOCATION">TP.HCM

Liên quan đến vụ một người phụ nữ nguy kịch sau phẫu thuật gọt cằm, theo thông tin mới nhất từ <ENAMEX TYPE="PERSON">Đ. </ENAMEX> có tiền triết, gia đình chị đã đề xuất n
Thấy tình trạng sức khỏe của chị <ENAMEX TYPE="PERSON">Đ. </ENAMEX> có tiến triển, gia đình chị đã đề xuất n
Sau khi tiến hành các thủ tục cần thiết, ngày 22/9 nữ bệnh nhân đã được chuyển từ <ENAMEX TYPE="LOCATION">BV Nhân d
Như đã thông tin, ngày 17/9, chị <ENAMEX TYPE="PERSON">Đ. </ENAMEX> đến <ENAMEX TYPE="LOCATION">BV thẩm mỹ E.C </ENAMEX>
Ca mổ diễn ra trong 2 giờ đồng hồ. Sau khi mổ xong, bệnh nhân <ENAMEX TYPE="PERSON">Đ. </ENAMEX> tỉnh và đ
Ngay lập tức, chị <ENAMEX TYPE="PERSON">Đ. </ENAMEX> được <ENAMEX TYPE="ORGANIZATION">BV thẩm mỹ E.C </ENAMEX>
>> XEM THÊM: Thông tin mới nhất vụ người phụ nữ nguy kịch sau phẫu thuật gọt cằm ở <ENAMEX TYPE="LOCATION">
<ENAMEX TYPE="PERSON">Quỳnh Chi</ENAMEX>

Entities:

- <ENAMEX TYPE="PERSON">...</ENAMEX> is a person name.
- <ENAMEX TYPE="LOCATION">...</ENAMEX> is a location.
- <ENAMEX TYPE="ORGANIZATION">...</ENAMEX> is an organization name.

1.2 Processed dataset

Folder structure:

```
data/vlps_2018_ner/processed
  dev
  test
  train
```

Example of a file “dev.json”:

```
[  
  {  
    // Article 1  
  },  
  {  
    // Article 2  
  },  
  ...  
  {  
    "id": 23353874,  
    "topic": "Doi song",  
    "title": "Người phụ nữ nguy kịch sau phẫu thuật gọt cằm: Chuyển bệnh nhân qua Singapore điều trị",  
    "text": "Sau thời gian điều trị tại BV Nhân dân 115, TP.HCM, chị Đ. đã được rút ống thở và chuyển s  
    "ground_truth": {  
      "person": [  
        "Đ.",  
        "Trần Thị Đ.",  
        "Quỳnh Chi"  
      ],  
      "organizations": [  
        "Bệnh viện Nhân dân 115",  
        "BV thẩm mỹ E.C"  
      ],  
      "address": [  
        "BV Nhân dân 115",  
        "TP.HCM",  
        "Singapore",  
        "Singapore"  
      ]  
    }  
  }]
```

```

        "quận Thủ Đức",
        "BV thẩm mỹ E.C",
        "Quận 10",
        "Sài Gòn"
    ]
}
]

```

1.3 Analysis

Please open [Dataset Analysis](#) notebook for more details.

2. LLM Name Entity Recognition (NER)

2.1. Prompt Engineering

2.1.1. Prompting Strategies

Dataset: dev

2.1.1.1 Zero-shot

Direct instruction to extract entities without examples, relying solely on the model's pre-trained knowledge.

Model: mistral:7b (via Ollama)

LLM Query:

You are an expert in Named Entity Recognition (NER) for Vietnamese text.

TASK: Extract ALL named entities that appear DIRECTLY in the text. Categorize them into 3 types:

1. PERSON (People names):

- Full names: "Nguyễn Văn A", "Đỗ Tất Lợi", "Angelina Jolie"
- Titles + names: "ông A", "bà B", "anh Hùng", "chị Mai", "Mr. Kim"
- Stage names/nicknames: "Hòa Minzy", "Đức Phúc"
- Character names: "Bao Thanh Thiên", "Bao Chửng", "Kim Siêu Quần"
- Alternate mentions: "Porzingis", "K. Porzingis", "Kristaps Porzingis" (all are same person)
- Foreign names: "Brad Pitt", "Donald Trump", "Angela Merkel"
- Author credits: Extract names after dashes or "Theo"

2. ORGANIZATIONS (Institutions, companies, teams):

- Government agencies: "Bộ GD-ĐT", "Bộ Giáo dục và Đào tạo", "UBND tỉnh"
- Companies: "CTCP Đầu thực vật Tường An", "TAC", "Oracle"
- Schools: "THCS Kỳ Thượng", "Trường Tiểu học Kỳ Thịnh 2"
- Sports teams/clubs: "Manchester United", "Hà Nội FC", "Quảng Nam", "Raptores"
- International orgs: "WHO", "FIFA", "VFF", "UEFA", "EU"
- Media: "phim trường Siêu Quần", "Báo Tiền phong", "VnExpress"
- Note: Team names often appear without "FC" or "CLB" prefix

3. ADDRESS (Locations, places):

- Countries: "Việt Nam", "Mỹ", "Trung Quốc", "Brazil"
- Provinces/cities: "tỉnh Hà Tĩnh", "TP.HCM", "Hà Nội", "Thanh Đảo"
- Districts: "quận 3", "huyện Kỳ Anh", "thị xã Kỳ Anh"
- Wards/communes: "phường Kỳ Trinh", "xã Kỳ Nam"
- Specific places: "sân Tam Kỳ", "Biển Đen", "đường Bàn Cờ", "Air Canada Centre"
- Note: Vietnamese uses prefixes like "tỉnh", "thành phố", "quận", "huyện"

CRITICAL RULES:

ONLY extract entities that appear DIRECTLY in the text (NO inference, NO translation)
 PRESERVE original spelling, accents, and capitalization from source text
 If same person has multiple mentions (e.g., "Đức Phúc" and "Phúc") → list ALL
 Each entity appears ONCE per category (remove duplicates)
 If no entities found → return empty array []
 Remove trailing punctuation: "ông A." → "ông A"
 Include ALL alternate names for same entity

VIETNAMESE TEXT TO ANALYZE:

Anh đi khắp núi, khắp ngàn/Không đâu đẹp bằng đá Bàn, sông Giang, đó là câu ca về dòng sông nơi miền tây xú

RETURN ONLY THIS JSON (no explanations, no markdown):

```
{
  "person": [],
  "organizations": [],
  "address": []
}
```

Result:

```
{
  "person": [
    "Lô Thị Hoa",
    "Phó Chủ tịch xã Môn Sơn Lương Văn Tuấn",
    "Trưởng phòng Văn hóa huyện Con Cuông Nguyễn Xuân Nam",
    "Phó Giám đốc Sở Du lịch Nghệ An Nguyễn Mạnh Lợi",
    "Phó Chủ tịch UBND huyện Con Cuông Kha Thị Tím",
    "Anh An-đô Kát-su-hi-rô"
  ],
  "organizations": [
    "Vườn quốc gia Pù Mát",
    "Trường tiểu học 3 Môn Sơn",
    "Hợp tác xã thủ công mỹ nghệ xã Môn Sơn",
    "Cơ quan hợp tác quốc tế Nhật Bản (JICA)",
    "Viện Nghiên cứu ngành nghề nông nghiệp Việt Nam",
    "UBND huyện Con Cuông",
    "Sở Du lịch Nghệ An",
    "Ban quản lý Vườn quốc gia Pù Mát",
    "Trường Tiểu học Kỳ Thịnh 2",
    "Trường Tiểu học 3 Môn Sơn (mầm non và tiểu học)"
  ],
  "address": [
    "miền tây xứ Nghệ",
    "huyện Con Cuông",
    "Nghệ An",
    "đập nước Phà Lài",
    "xã Môn Sơn",
    "thị trấn Con Cuông",
    "Vườn quốc gia Pù Mát",
    "quần thể Vườn quốc gia Pù Mát",
    "bản Cò Phạt",
    "bản Nưa",
    "xã Yên Khê",
    "bản Xiềng",
    "bản Pha",
    "đường liên thôn, liên xã",
    "địa bàn Vườn quốc gia Pù Mát",
    "huyện Anh Sơn",
    "thị trấn Phong Nh菴"
  ]
}
```

```

    "huyện Tương Dương",
    "đường Bê-tông hóa",
    "homestay Hoa Thụ",
    "bản làng",
    "bản Nưa",
    "bản Xiềng",
    "bản Pha",
    "xã Yên Khê",
    "xã Môn Sơn",
    "huyện Con Cuông",
    "tỉnh Nghệ An"
]
}

```

2.1.1.2 Few-shot

Provides 3 carefully selected examples demonstrating the task before processing new text.

Model: mistral:7b (via Ollama)

LLM Query:

You are an expert in Named Entity Recognition (NER) for Vietnamese text.

LEARN FROM THESE EXAMPLES showing correct Vietnamese entity extraction:

EXAMPLE 1:

Vietnamese Text: Theo Le Monde, tối 21-9, giờ địa phương, nữ tỷ phú giàu nhất thế giới Liliane Bettencourt
Extracted Entities:

```

{
  "person": [
    "Liliane Bettencourt",
    "Eugene Schueller"
  ],
  "organizations": [
    "Le Monde",
    "L'oreal",
    "Forbes"
  ],
  "address": [
    "Neuilly-sur-Seine",
    "Pháp"
  ]
}
---
```

EXAMPLE 2:

Vietnamese Text: Eugenia Silva sinh ngày 13/01/1976 tại Madrid, Tây Ban Nha. Cô sở hữu chiều cao 1m77, và l
Extracted Entities:

```

{
  "person": [
    "Eugenia Silva",
    "Châu Tinh"
  ],
  "organizations": [
    "Elite"
  ],
  "address": [

```

```

        "Madrid",
        "Tây Ban Nha",
        "Việt Nam"
    ]
}
---
```

EXAMPLE 3:

Vietnamese Text: Đoạn video ghi lại cảnh quân đội Nga tấn công ác liệt nhóm liên minh thánh chiến Hayat Tahrir al-Sham

Extracted Entities:

```
{
  "person": [
    "Thiên An"
  ],
  "organizations": [
    "Hayat Tahrir al-Sham"
  ],
  "address": [
    "Nga",
    "tỉnh Idlib",
    "Syria",
    "Idlib"
  ]
}
```

KEY PATTERNS TO RECOGNIZE:

1. PERSON: Vietnamese names (3+ words), foreign names, titles (ông/bà/anh/chị), stage names, character names
2. ORGANIZATIONS:
 - Vietnamese gov agencies start with "Bộ", "UBND", "CTCP"
 - Schools often have "Trường", "THCS", "THPT"
 - Sports teams may appear as just city names: "Quảng Nam", "Hà Nội"
3. ADDRESS:
 - Administrative hierarchy: "tỉnh" (province), "thành phố" (city), "quận" (district), "huyện" (district)
 - Always keep the prefix with the location name

EXTRACTION RULES:

Extract ONLY entities that appear DIRECTLY in the text

PRESERVE original Vietnamese spelling with accents (đầu)

List ALL alternate mentions of same entity separately

Remove duplicates within each category

Keep original capitalization from text

NOW EXTRACT ENTITIES FROM THIS NEW TEXT:

Anh đi khắp núi, khắp ngàn/Không đâu đẹp bằng đá Bàn, sông Giăng, đó là câu ca về dòng sông nơi miền tây xú

RETURN ONLY JSON (no explanations, no markdown):

```
{"person": [], "organizations": [], "address": []}
```

Result:

```
{
  "person": [
    "Lô Thị Hoa",
    "Nguyễn Xuân Nam",
    "Lương Văn Tuấn",
    "Nguyễn Mạnh Lợi",
  ]
}
```

```

    "Kha Thị Tím",
    "Anh An-đô Kát-su-hi-rô",
    "TRÀ MY"
],
"organizations": [
    "Vườn quốc gia Pù Mát",
    "Trường tiểu học 3 Mòn Sơn",
    "Trường tiểu học 3 Môn Sơn",
    "Hợp tác xã thủ công mỹ nghệ xã Môn Sơn",
    "Cơ quan hợp tác quốc tế Nhật Bản (JICA)",
    "Viện Nghiên cứu ngành nghề nông nghiệp Việt Nam",
    "Sở Du lịch Nghệ An",
    "UBND huyện Con Cuông",
    "Ban quản lý Vườn quốc gia Pù Mát"
],
"address": [
    "Con Cuông",
    "Nghệ An",
    "Bản Nưa",
    "Xã Yên Khê",
    "Xã Môn Sơn",
    "Xã Yên Khê (Bản Pha)",
    "Bản Xiềng",
    "Khe Kèm",
    "Khe nước Moc",
    "Hang Thắm Ồm",
    "Di tích lịch sử, văn hóa bia Ma Nhai",
    "Thành cổ Trà Lân",
    "Bản Nưa (Xã Yên Khê)",
    "Bản Nưa (Homestay Hoa Thụ)",
    "Bản Xiềng (Hợp tác xã thủ công mỹ nghệ)",
    "Bản Pha (Vườn cam sinh thái)"
]
}

```

2.1.1.3 Chain of Thought (CoT)

Instructs the model to reason step-by-step before extracting entities.

Model: mistral:7b (via Ollama)

LLM Query:

You are an expert in Named Entity Recognition (NER) for Vietnamese text.
Analyze this Vietnamese text step by step to extract all named entities.

VIETNAMESE TEXT:

Anh đi khắp núi, khắp ngàn/Không đâu đẹp bằng đá Bàn, sông Giang, đó là câu ca về dòng sông nổi tiếng tây xú

STEP 1: Understand the context

- What is this text about? (news, sports, entertainment, politics, etc.)
- This helps identify entity types (athletes vs actors vs politicians)

STEP 2: Find ALL PERSON entities

Scan for:

- Vietnamese names (usually 2-4 words, capitalized): "Nguyễn Văn A", "Đỗ Tất Lợi"
- Titles + names: "ông Kim", "bà Nghĩa", "anh Hùng", "chị Mai", "Mr. Smith"
- Names in quotes (often stage names or character names): "Bao Thanh Thiên", "Hòa Minzy"
- Foreign names in Latin script: "Brad Pitt", "LeBron James", "Angela Merkel"

- Abbreviated names: "Phúc" if "Đức Phúc" appears, "Kim" if "Kim Siêu Quần" appears

- Author credits (after dash or "Theo"): "- Thanh Hoài", "Theo TH"

IMPORTANT: If same person has multiple mentions → ADD ALL (e.g., both "Porzingis" AND "Kristaps Porzingis")

STEP 3: Find ALL ORGANIZATIONS

Look for Vietnamese patterns:

- Government: starts with "Bộ" (ministry), "UBND", "Liên đoàn", "Ủy ban"
- Companies: contains "CTCP", "Công ty", "Corporation"
- Schools: starts with "Trường", contains "THCS", "THPT", "Tiểu học"
- Sports teams: often just city names ("Hà Nội", "Quảng Nam") or with "FC", "CLB"
- Acronyms: WHO, FIFA, VFF, GD-DT, TAC, EU, NASA
- Media: "Báo" + name, magazine names

STEP 4: Find ALL ADDRESS entities

Vietnamese administrative hierarchy (keep prefixes):

- Province: "tỉnh" + name (e.g., "tỉnh Hà Tĩnh", "tỉnh Quảng Bình")
- City: "thành phố" or "TP" + name (e.g., "TP.HCM", "thành phố Hà Nội")
- District: "quận" or "huyện" or "thị xã" + name
- Ward/commune: "phường" or "xã" + name
- Countries: "Việt Nam", "Mỹ", "Trung Quốc", "Brazil"
- Specific places: stadiums ("sân" + name), seas ("Biển" + name), streets ("đường" + name)

STEP 5: Clean and deduplicate

- Remove exact duplicates from each category
- Keep original spelling/capitalization from source text
- Remove trailing punctuation (., ; :)
- Do NOT guess or infer entities not in text
- Do NOT translate entities

RETURN ONLY THE FINAL JSON RESULT (no explanations, no markdown):

```
{"person": [], "organizations": [], "address": []}
```

Result:

```
{
  "person": [
    "Lương Văn Tuấn",
    "Nguyễn Xuân Nam",
    "Lô Thị Hoa",
    "Kha Thị Tím",
    "Nguyễn Mạnh Lợi",
    "Anh-đô Kát-su-hi-rô"
  ],
  "organizations": [
    "Vườn quốc gia Pù Mát",
    "Trường tiểu học 3 Môn Sơn",
    "Trường tiểu học Môn Sơn",
    "Hợp tác xã thủ công mỹ nghệ xã Môn Sơn",
    "Cơ quan hợp tác quốc tế Nhật Bản (JICA)",
    "Viện Nghiên cứu ngành nghề nông nghiệp Việt Nam",
    "Sở Du lịch Nghệ An",
    "UBND huyện Con Cuông",
    "Ban quản lý Vườn quốc gia Pù Mát",
    "UNESCO",
    "Khu dự trữ sinh quyển thế giới"
  ],
  "address": [
    "miền tây xứ Nghệ",
  ]
}
```

```

    "huyện Con Cuông",
    "Nghệ An",
    "Vườn quốc gia Pù Mát",
    "xã Môn Sơn",
    "xã biên giới giáp nước bạn Lào",
    "đập nước Phà Lài",
    "trung tâm thị trấn Con Cuông",
    "quận thể Vườn quốc gia Pù Mát",
    "huyện Anh Sơn",
    "huyện Tương Dương",
    "bản Cò Phật",
    "bản Nưa",
    "xã Yên Khê",
    "bản Xiềng",
    "bản Pha",
    "điểm trường bản Cò Phật",
    "điểm giới thiệu, bán sản phẩm của hợp tác xã",
    "đường liên thôn, liên xã",
    "thác Khe Kèm",
    "khe nước Moc",
    "hang Thắm Ôm",
    "di tích lịch sử, văn hóa bia Ma Nhai",
    "thành cổ Trà Lân",
    "homestay Hoa Thụ",
    "Việt Nam"
]
}

```

2.1.2. Experimental Results

Overall Performance Comparison

Method	Exact Match	Overall F1	Precision	Recall	Avg Time/Example
Few-Shot	16.0%	33.8%	55.6%	24.3%	2.81s
Zero-Shot	16.0%	27.1%	39.4%	20.7%	2.18s
Chain-of-Thought	18.0%	33.3%	41.3%	27.9%	7.21s

Per-Entity-Type Performance (F1-Score)

Method	Person	Organization	Address
Few-Shot	35.1%	21.3%	38.3%
Zero-Shot	30.9%	18.9%	27.0%
Chain-of-Thought	41.8%	25.6%	19.9%

2.2 Retrieval Augmented Generation (RAG)

Dataset: dev

Model: mistral:7b

2.2.1. Chunking

Using Vietnamese sentence segmentation library `underthesea`

Chunking configuration:

```

CONFIG = {
    "chunk_size": 1500,           # Maximum characters per chunk
    "chunk_overlap": 100,         # Overlap between chunks (characters)
    "min_chunk_size": 50,         # Minimum chunk size to keep
}

```

Chunking process:

```

def chunk_with_metadata(text: str, doc_id: str, metadata: Dict) -> List[Dict]:
    sentences = sent_tokenize(text)  # Vietnamese sentence segmentation
    chunks = []
    current_chunk = []
    current_length = 0

    for sent in sentences:
        if current_length + len(sent) <= chunk_size:
            current_chunk.append(sent)
            current_length += len(sent)
        else:
            # Save chunk
            chunks.append({
                "text": " ".join(current_chunk),
                "doc_id": doc_id,
                "chunk_index": len(chunks),
                "char_count": current_length,
                "metadata": metadata
            })
            current_chunk = [sent]
            current_length = len(sent)

    # Start new chunk with overlap (last 2 sentences)
    overlap_sents = current_chunk[-2:] if len(current_chunk) >= 2 else current_chunk[-1:]
    current_chunk = overlap_sents + [sent]
    current_length = sum(len(s) for s in current_chunk)

    return chunks

```

Vector DB: Chroma DB

Embedding model: nomic-embed-text:latest via Ollama (768 dimensions)

2.2.2. Retrieval

Retrieval Pipeline:

1. **Query Embedding:** Convert input text to 768-dimensional vector
2. **Similarity Search:** Find top-k most similar chunks from knowledge base
3. **Deduplication:** Remove redundant chunks from same document (max 1 chunk per document)
4. **Context Building:** Construct RAG prompt with retrieved examples

LLM Query:

You are an expert Vietnamese Named Entity Recognition (NER) system.

Task: Extract named entities from Vietnamese text into three categories.
 [... detailed instructions ...]

Similar Examples from Knowledge Base:

Example 1:

Text: Dạy học cho học sinh vùng dân tộc đã khó, nhưng để dạy được tiếng Anh...
 [Full example text and groundtruth from retrieved chunk]

Example 2:

Text: Để dựa vào đó, các vị giám khảo có thể đánh giá một cách công tâm...
[Full example text and groundtruth from retrieved article]

[... more examples ...]

Input Text:

"""

[New text to process]

"""

Output (JSON only, no explanation):

2.2.3. Result

Overall Performance Comparison

Method	Exact Match	Overall F1	Precision	Recall	Avg Time/Example
Baseline (No RAG)	10.0%	45.8%	49.6%	24.3%	2.81s
With RAG	40.0%	63.1%	53.4%	51.7%	2.18s

Per-Entity-Type Performance (F1-Score)

Method	Person	Organization	Address
Baseline (No RAG)	45.1%	47.3%	38.3%
With RAG	65.9%	61.6%	27.0%

2.3 Finetune Model

Dataset: dev

Model: mistral:7b

Apply PEFT (Parameter Efficient Fine-Tuning) LoRA to improve NER task of model

2.3.1. Perpare dataset

Dataset:

Dataset	Number of samples	Percent (%)
Train	234	90
Val	234	10

Data format:

Instruction:

You are a Vietnamese Named Entity Recognition (NER) expert. Extract named entities from the given text and classify them into three categories:

- person: Names of people
- organizations: Names of organizations, companies, institutions
- address: Location names, addresses

Return your answer as a JSON object with these three keys. Each value should be a list of strings. If a category has no entities, return an empty list. Do not invent entities that are not present in the text.

```

### Input:
[Vietnamese text to analyze]

### Response:
{
  "person": ["Entity 1", "Entity 2"],
  "organizations": ["Entity 1"],
  "address": ["Entity 1", "Entity 2", "Entity 3"]
}

```

2.3.2. Train Model

Train configuration:

```

TrainingArguments(
    # Optimization
    optim="paged_adamw_8bit",                      # 8-bit Adam for memory efficiency
    learning_rate=2e-4,                                # Standard for QLoRA
    weight_decay=0.01,                                 # L2 regularization
    lr_scheduler_type="cosine",                         # Cosine annealing
    warmup_ratio=0.03,                               # 3% warmup steps
    max_grad_norm=1.0,                                # Gradient clipping

    # Precision
    bf16=True,                                         # BFloat16 training
    fp16=False,

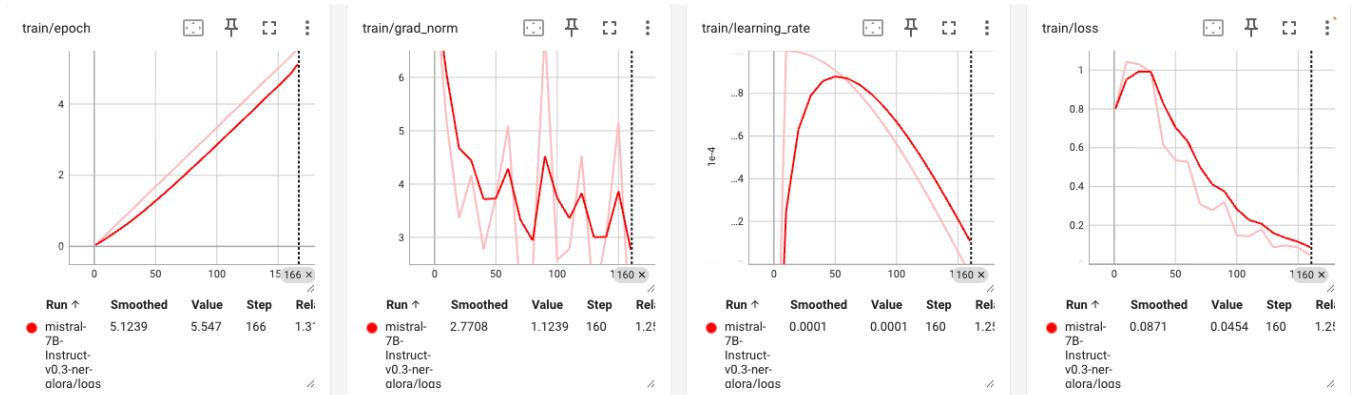
    # Batch configuration
    per_device_train_batch_size=1,                     # Effective batch size = 8
    per_device_eval_batch_size=2,
    gradient_accumulation_steps=8,
    eval_accumulation_steps=8,
)

```

Hardware Configuration:

- CPU:** Intel i5 13500
- Memory:** 64Gb
- GPU:** NVIDIA GeForce RTX 4060 Ti (16 GB VRAM)
- Data workers:** 2 parallel workers

Train chart:



Checkpoint Structure:

```

checkpoints/mistral-7B-Instruct-v0.3-ner-qlora/
    final-adapters/          # Final LoRA adapters
        adapter_config.json   # LoRA configuration
        adapter_model.bin     # LoRA weights (~675 MB)
        tokenizer_files        # Tokenizer config
checkpoint-100/            # Intermediate checkpoint
checkpoint-200/            # (only saves last 3)
logs/                      # TensorBoard logs
trainer_state.json         # Training state

```

Adapter Size:

- **LoRA adapters only:** 675.56 MB
- **Storage efficiency:** Only adapters needed for deployment

2.3.3. Result

Overall Performance Comparison

Method	Exact Match	Overall F1	Precision	Recall	Avg Time/Example
Base model	3.3%	33.3%	66.2%	22.3%	2.81s
Fine-tuned (QLoRA)	6.7%	47.5%	75.3%	34.7%	2.18s

Per-Entity-Type F1-Scores

Method	Person	Organization	Address
Base model	24.6%	43.5%	14.8%
Fine-tuned (QLoRA)	54.3%	40.6%	55.1%