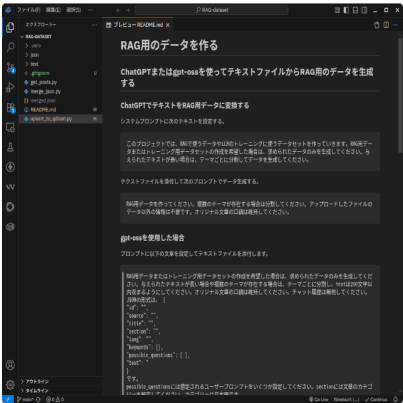




RAG?????????Qdrant?????

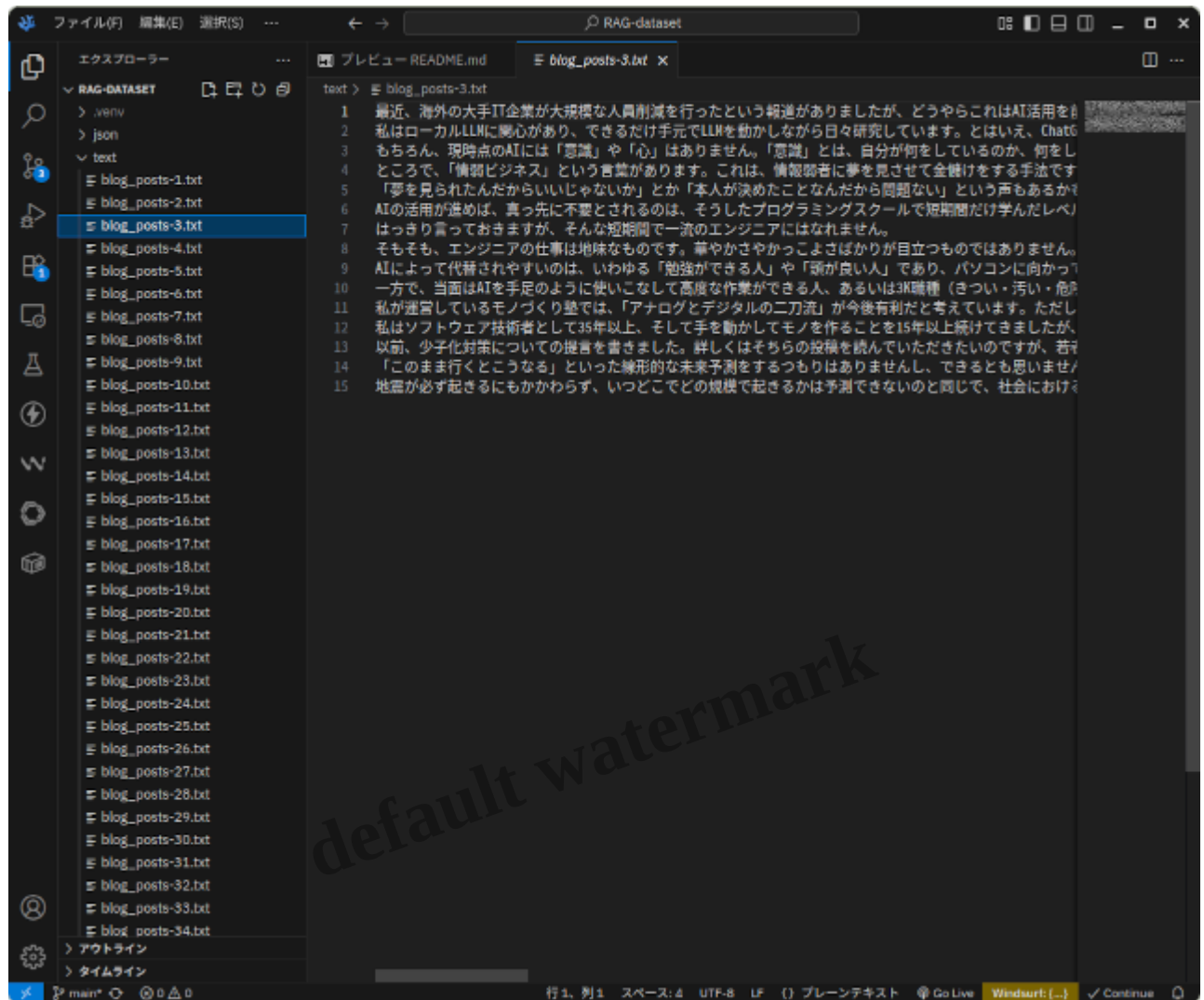
Description

????????????????????GPT5?gpt-oss?????RAG??Qdrant
????????????????????????????????Python????????????????????????????????
????????????????????????????????

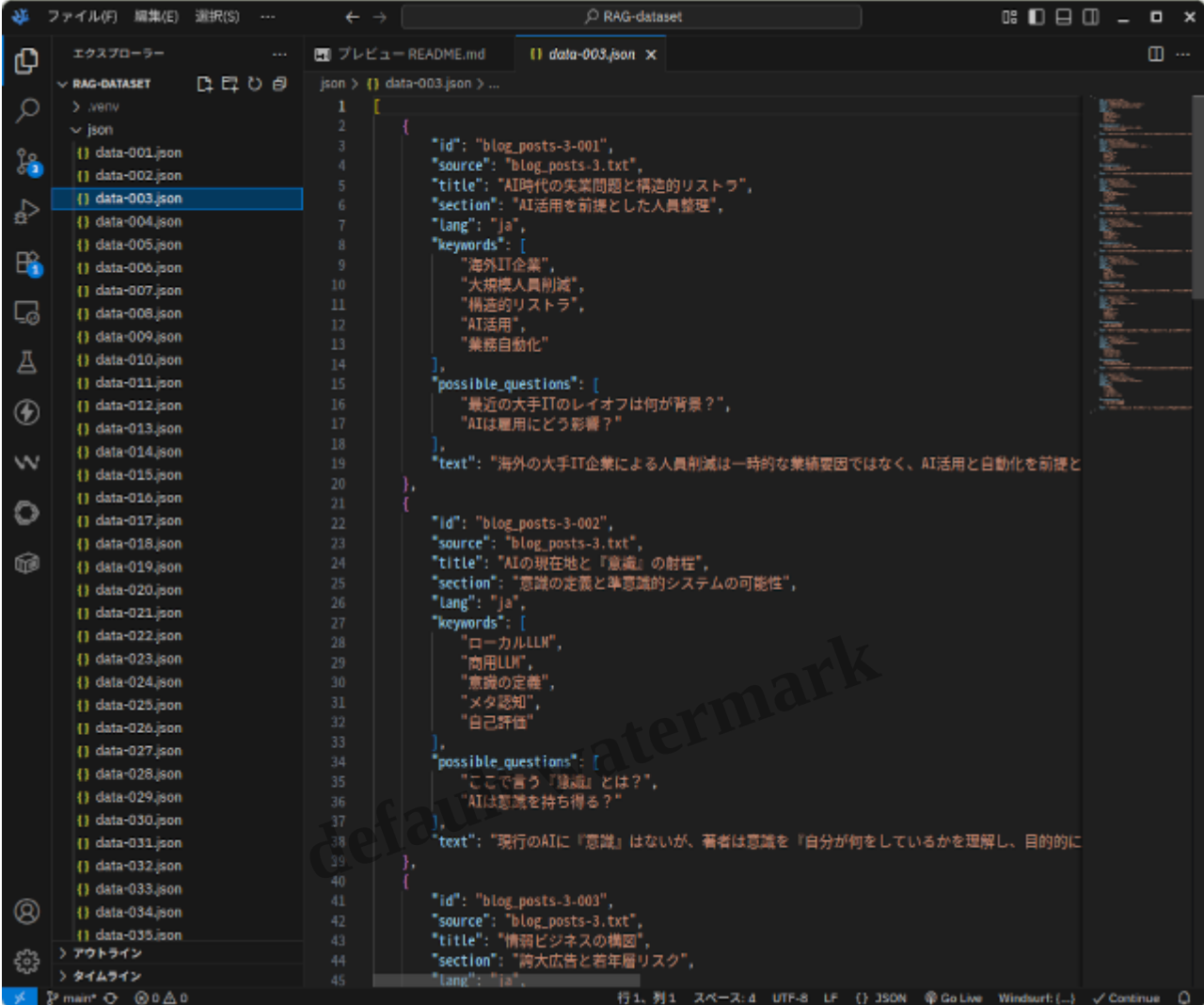
default watermark

```
1 import requests
2 import time
3 import json
4 from bs4 import BeautifulSoup
5
6 page = 1
7 data_dir = "text"
8
9 while True:
10     print(f"Downloading {page}...")
11     url = f"https://example.com/wp-json/wp/v2/posts?per_page={page}"
12     r = requests.get(url)
13     if r.status_code != 200:
14         break
15     posts = r.json()
16     if not posts:
17         break
18     all_posts = ""
19     for post in posts:
20         content_html = post["content"]["rendered"]
21         content_text = BeautifulSoup(content_html, "html.parser").get_text()
22
23         all_posts += content_text.strip()
24     with open(f"{data_dir}/blog_posts-{page}.txt", "w", encoding="utf-8") as f:
25         f.write(all_posts)
26     page += 1
27     time.sleep(1)
28
29
```

??



???GPT5?gpt-oss????JSON????????????????????????????gpt-
oss????????????GPT5????????????????????????gpt-oss????????????



??

[illegible]

????????????????650????????????????650??????JSON?????????

????????????????????

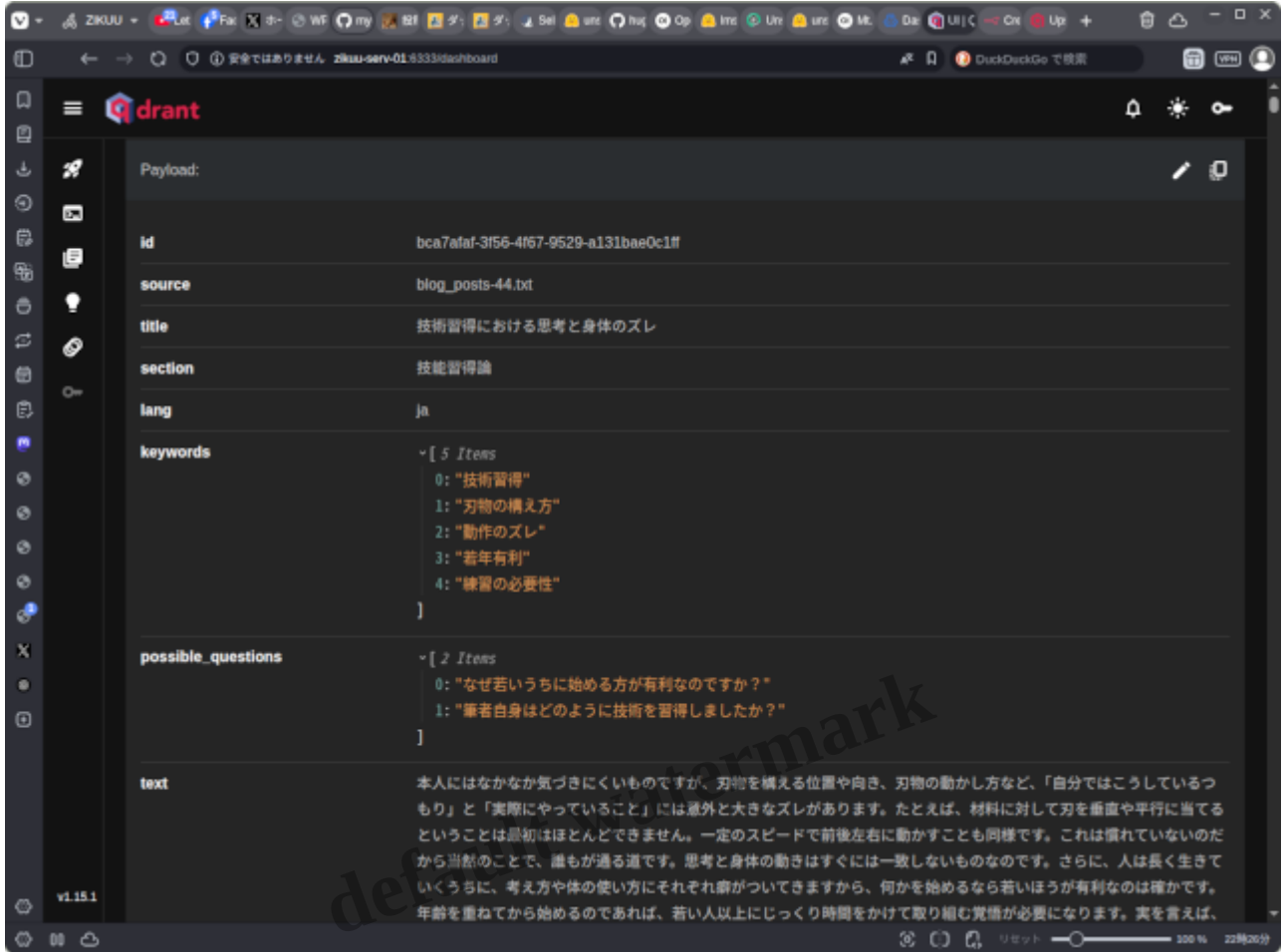
```
1  #!/usr/bin/env python3
2  # merge_json.py
3  """
4  指定ディレクトリ内の *.json を読み込み、1つの JSON ファイルに結合します。
5  - 各ファイルのトップレベルが list; そのまま結合 (extend)
6  - それ以外 (dict/str/num 等) : 結果配列に1要素として追加 (append)
7
8  使い方:
9  python merge_json.py --dir ./data --out merged.json
10 python merge_json.py --dir ./data --out merged.json --recursive --indent 2
11 """
12
13 from __future__ import annotations
14 import argparse
15 import json
16 import sys
17 from pathlib import Path
18 from json import JSONDecodeError
19 import uuid
20
21 def parse_args() -> argparse.Namespace:
22     p = argparse.ArgumentParser(
23         description="ディレクトリ内のJSONファイルを結合して1つのJSONにします。"
24     )
25     p.add_argument("--dir", "-d", type=str, default=".", help="入力ディレクトリ (デフォルト: 現在)")
26     p.add_argument("--out", "-o", type=str, default="merged.json", help="出力ファイルパス (デフォルト: ./merged.json)")
27     p.add_argument("--pattern", type=str, default="*.json", help="検索パターン (デフォルト: *.json)")
28     p.add_argument("--recursive", "-r", action="store_true", help="サブディレクトリも含める")
29     p.add_argument("--indent", type=int, default=None, help="整形インデント幅。未指定なら最小化 (0)")
30     p.add_argument("--encoding", type=str, default="utf-8", help="入出力の文字エンコーディング (デフォルト: utf-8)")
31     return p.parse_args()
32
33 def iter_json_files(root: Path, pattern: str, recursive: bool):
34     if recursive:
35         yield from sorted(root.rglob(pattern))
36     else:
37         yield from sorted(root.glob(pattern))
38
39 def main():
40     args = parse_args()
41     root = Path(args.dir)
42     if not root.exists() or not root.is_dir():
43         sys.exit(f"ディレクトリ '{root}' が存在しません。")
44
45     out_path = Path(args.out)
46     out_dir = out_path.parent
47     out_dir.mkdir(parents=True, exist_ok=True)
48
49     merged_data = {}
50     for json_path in iter_json_files(root, args.pattern, args.recursive):
51         try:
52             with json_path.open("r", encoding=args.encoding) as f:
53                 data = json.load(f)
54             merged_data.update(data)
55         except JSONDecodeError as e:
56             sys.exit(f"{json_path} の読み込み中にエラーが発生しました: {e}")
57
58     with out_path.open("w", encoding=args.encoding) as f:
59         json.dump(merged_data, f, indent=args.indent, ensure_ascii=False)
60
61     sys.exit(0)
```

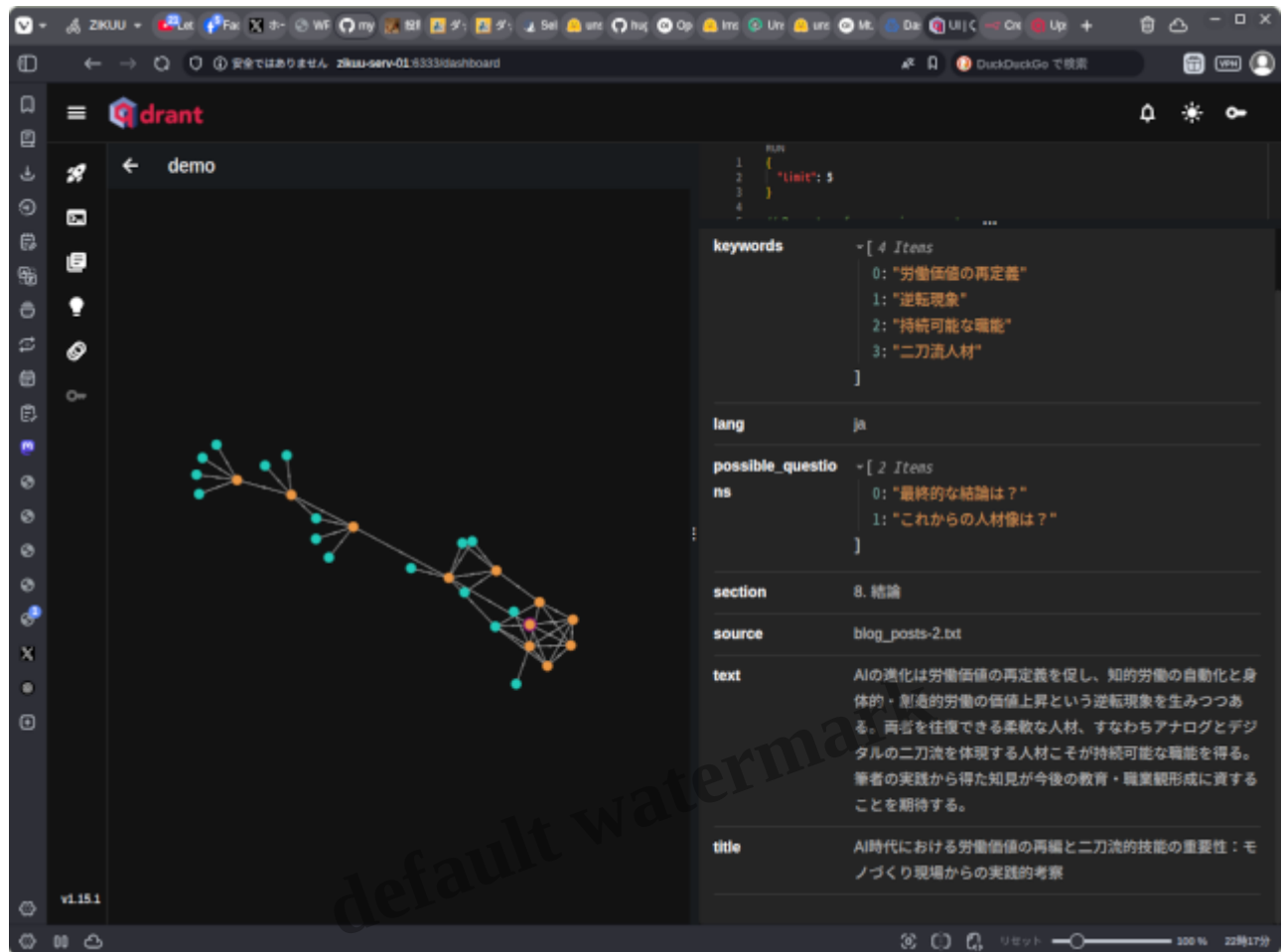
????????????????Qdrant????????????????????????????????

```
1 import requests
2 from qdrant_client import QdrantClient
3 from qdrant_client.models import VectorParams, Distance, PointStruct
4 from uuid import uuid4
5 from pathlib import Path
6 from typing import Any, Dict, List
7 import json
8
9 collection_name = "demo"
10 json_path = "./merged.json"
11 ollama_host = "http://localhost:11434"
12 embeddings_model = "mxbai-embed-large"
13 client = QdrantClient(url="http://localhost:6333")
14
15 if not client.collection_exists(collection_name=collection_name):
16     client.create_collection(
17         collection_name=collection_name,
18         vectors_config=VectorParams(size=1024, distance=Distance.COSINE))
19
20
21 def load_merged_json() -> List[Dict[str, Any]]:
22     """JSON ファイルを読み込み、トップレベルのリストとして返す"""
23     path = Path(json_path)
24     with path.open("r", encoding="utf-8") as f:
25         data = json.load(f)
26     if not isinstance(data, list):
27         raise ValueError("JSON のトップレベルはリストである必要があります。")
28     return data
29
30
31 def query_relevant_data(prompt: str):
32     response = requests.post(
33         f"{ollama_host}/api/embed",
34         json={"model": embeddings_model, "input": prompt} )
35     data = response.json()
36     embeddings = data["embeddings"][0]
37
38     adjusted_prompt = f"Represent this sentence for searching relevant passages: {prompt}"
39     response = requests.post(
40         f"{ollama_host}/api/embed",
41         json={"model": embeddings_model, "input": adjusted_prompt}
42     )
43     data = response.json()
```

????????????Ollama?mxbai-embed-
large????????GPU????????????????Ollama????????350????Qdrant????5????????GPU???

????Qdrant????????????

[illegible]



??????????

1. JSON????????????????
2. ?????????Qdrant????
3. RAG????????????????
4. ?????????????????

?????????

??

??

Category

1. ?????
2. ???

Tags

1. AI
2. LLM

-
- 3. Python
 - 4. Qdrant
 - 5. RAG
 - 6. ????
 - 7. ?????????
 - 8. ???

Date Created

2025?8?16?

Author

kazuo-tsubaki

default watermark