iamarunbrahma / **vision-parse** Public

Parse PDFs into markdown using Vision LLMs

MIT license

177 stars 22 forks Branches Tags Activity

Star Notifications

Code Issues 1 Pull requests Actions Projects Security Insights

main 1 Branch 11 Tags Go to file Go to file Code

iamarunbrahma update llm.py ✓ c8b693c · 5 days ago

.github	ci: create codeql.yml	last week
benchmarks	Update requirements.txt	last week
examples	Enhanced Vision Parser and LLM functi...	last week
src/vision_parse	update llm.py	5 days ago
tests	feat: Added citation file, provided custo...	last week
.gitignore	chore: update .gitignore	last week
CITATION.cff	feat: Added citation file, provided custo...	last week
CONTRIBUTING.md	chore: add contribution guidelines, issu...	last month
Dockerfile	build: create Dockerfile and docker-co...	2 weeks ago
LICENSE	Initial commit	last month
Makefile	build: update Makefile	last week
README.md	Update README.md	5 days ago
docker-compose.yml	Update docker-compose.yml	2 weeks ago
pyproject.toml	build(deps): update pyproject.toml and ...	last week
uv.lock	build(deps): update pyproject.toml and ...	last week

README MIT license

License MIT Author Arun Brahma pypi v0.1.11

> 🚀 Parse PDF documents into beautifully formatted markdown content using state-of-the-art Vision Language Models - all with just a few lines of code!

🎯 Introduction

Vision Parse harnesses the power of Vision Language Models to revolutionize document processing:

- 📝 **Scanned Document Processing**: Intelligently identifies and extracts text, tables, and LaTeX equations from scanned documents into markdown-formatted content with high precision
- 🎨 **Advanced Content Formatting**: Preserves LaTeX equations, hyperlinks, images, and document hierarchy for markdown-formatted content
- 🤖 **Multi-LLM Support**: Seamlessly integrates with multiple Vision LLM providers such as OpenAI, Gemini, and Llama for optimal accuracy and speed
- 📁 **Local Model Hosting**: Supports local model hosting with Ollama for secure, no-cost, private, and offline document processing

🚀 Getting Started

Prerequisites

- 🐍 Python >= 3.9
- 🖥️ Ollama (if you want to use local models)
- 🤖 API Key for OpenAI or Google Gemini (if you want to use OpenAI or Google Gemini)

Installation

Install the core package using pip (Recommended):

```
pip install vision-parse
```

Install the additional dependencies for OpenAI or Gemini:

```
# For OpenAI support
pip install 'vision-parse[openai]'
```

```
# For Gemini support
pip install 'vision-parse[gemini]'
```

```
# To install all the additional dependencies
pip install 'vision-parse[all]'
```

Install the package from source:

```
pip install 'git+https://github.com/iamarunbrahma/vision-parse.git#egg=vision-parse[all]'
```

Setting up Ollama (Optional)

See examples/ollama_setup.md on how to setup Ollama locally.

⌛ Usage

Basic Example Usage

```
from vision_parse import VisionParser

# Initialize parser
parser = VisionParser(
    model_name="llama3.2-vision:11b", # For local models, you don't need to provide the api key
    temperature=0.4,
    top_p=0.5,
    image_mode="url", # Image mode can be "url", "base64" or None
    detailed_extraction=False, # Set to True for more detailed extraction
    enable_concurrency=False, # Set to True for parallel processing
)

# Convert PDF to markdown
pdf_path = "path/to/your/document.pdf" # local path to your pdf file
markdown_pages = parser.convert_pdf(pdf_path)

# Process results
for i, page_content in enumerate(markdown_pages):
    print(f"\n--- Page {i+1} ---\n{page_content}")
```

Customize Ollama configuration for better performance

```
from vision_parse import VisionParser

custom_prompt = """
Strictly preserve markdown formatting during text extraction from scanned document.
"""

# Initialize parser with Ollama configuration
parser = VisionParser(
    model_name="llama3.2-vision:11b",
    temperature=0.7,
    top_p=0.6,
    num_ctx=4096,
    image_mode="base64",
    custom_prompt=custom_prompt,
    detailed_extraction=True,
    ollama_config={
        "OLLAMA_NUM_PARALLEL": 8,
        "OLLAMA_REQUEST_TIMEOUT": 240,
    },
    enable_concurrency=True,
)

# Convert PDF to markdown
pdf_path = "path/to/your/document.pdf"
markdown_pages = parser.convert_pdf(pdf_path)
```

OpenAI or Gemini Model Usage

```
from vision_parse import VisionParser

# Initialize parser with OpenAI model
parser = VisionParser(
    model_name="gpt-4o",
    api_key="your-openai-api-key", # Get the OpenAI API key from https://platform.openai.com/api-keys
    temperature=0.7,
    top_p=0.4,
    image_mode="url",
    detailed_extraction=True, # Set to True for more detailed extraction
    enable_concurrency=True,
)

# Initialize parser with Google Gemini model
parser = VisionParser(
    model_name="gemini-1.5-flash",
    api_key="your-gemini-api-key", # Get the Gemini API key from https://aistudio.google.com/app/apikey
    temperature=0.7,
    top_p=0.4,
    image_mode="url",
    detailed_extraction=True, # Set to True for more detailed extraction
```

```
        enable_concurrency=True,
    )
```

✅ Supported Models

This package supports the following Vision LLM models:

- OpenAI: `gpt-4o` , `gpt-4o-mini`
- Google Gemini: `gemini-1.5-flash` , `gemini-2.0-flash-exp` , `gemini-1.5-pro`
- Meta Llama and LLava from Ollama: `llava:13b` , `llava:34b` , `llama3.2-vision:11b` , `llama3.2-vision:70b`

License

This project is licensed under the MIT License - see the LICENSE file for details.

Releases

11 tags

Packages

No packages published

Contributors 2

iamarunbrahma Arun Brahma

mark-beeby

Languages

● Python 92.2% ● Jinja 5.4% ● Dockerfile 1.6% ● Makefile 0.8%