

Quick Start with EazyML GenAI APIs

1 | Overview

Generative AI (GenAI) is an involved science, its models often complex, not easy to understand. The sequel describes how to use EazyML GenAI API based package to explore information extraction with EazyML. The python package is called “eazyml_upload_extract_information.py”. This package is offered as a quick start to dive into the comprehensive set of GenAI APIs described on eazyml.com.

2 | Authenticate

EazyML authenticates you with username and API token. You can obtain your API token by logging into EazyML service portal and then navigating to “My Accounts” → “API Key”.

Usage:

```
python eazyml_upload_extract_information.py --username <username> --api_key <api_key>
```

Example:

```
python eazyml_upload_extract_information.py --username vikr.nunia@gmail.com --api_key <api_key>
```

Authentication successful.

Authentication information is stored in authentication.json

Please note that your authentication information gets stored in a local file and then gets used for all subsequent calls described in the sequel below. Authentication is the mandatory first step in any experiment.

3 | Upload and Preprocess a Document

Post authentication, you can then upload a document to EazyML for pre-processing. The pre-processing steps include sanity checks. If you are uploading a document then the index name and document_path is mandatory. There is one **overwrite** parameter which is optional with default value as ‘no’ and can be used to overwrite the document embeddings stored under specified **index_name** in search engine. You may optionally provide a **prefix name** that will be prefixed to the auto generated output file names – so that you can keep your various output files organized for your documents.

Usage:

```
python eazym1_upload_extract_information.py --prefix_name  
<file_prefix> --document_path <document path> --index_name  
<index_name>
```

Example:

```
python eazym1_upload_extract_information.py --prefix_name NLP --  
document_path NLP.pdf --index_name nlp --overwrite "no"
```

Authentication successful.

Indexing the document...

The document is indexed successfully to EazyML: NLP.pdf

The response is stored in NLP_upload_document.json

Indexing document time: 2.81 secs

A Boolean flag indicating whether document is stored (indexed) is : True

Likely next steps:

```
Python eazym1_upload_extract_information.py -query None -index_name nlp  
--extract_information
```

Please provide an existing index name that you mentioned while uploading a document.

After a successful upload of your document, a Boolean flag will indicate that your document is stored in search engine for subsequent operations on this document– such as extracting information relevant to the pdf by providing user query.

4 | Extract Information from the document

Once the document is successfully uploaded and pre-processed, it's ready for extracting information for any queries user may have regarding the document. The sequel shows how to extract information for user queries.

Usage:

```
python eazym1_upload_extract_information.py --prefix_name  
<file_prefix> --query <query> --index_name <index_name>
```

Example:

```
python eazym1_upload_extract_information.py --prefix_name NLP --query  
What is bottom-up attention model? --index_name nlp
```

Authentication successful.

Extracting Information ...

Information extracted successfully.

Information Extracting time: 0.35 secs

The answer retrieved for provided query from the document is : The bottom-up attention model is a mechanism that proposes image regions based on features extracted from an image using Faster R-CNN. Each proposed region is associated with a feature vector, and this model helps in determining feature weightings for attention calculation at the level of objects and other salient image regions.

The response is stored in NLP_extract_information.json

After an information is retrieved successfully, answer response is provided back to user for reference. There is one output file generated containing information retrieved from the document relevant to the query.

1 | Custom Configuration

Here's how you can customize machine learning and transparency experiments with your own configuration parameters. Your configuration parameter choices could be specified in a configuration file in [.ini format](#). List of all available configuration parameters can be found in the [sample configuration file](#). The configuration parameters specifically for transparent machine learning are explained in a [user guide here](#). The sequel describes how you upload your custom configuration for an experiment.

Usage:

```
python eazymml_upload_extract_information.py --config_file  
<configurationfile>
```

Example:

```
python eazymml_upload_extract_information.py --config_file  
sample_config.ini
```

Authentication successful.

Uploading config file ...

Config file uploaded and set successfully.

You are welcome to enhance this package with various other configurability options that EazyML provides and build powerful applications.