У проекта tve есть определенная файловая структура:

Базовая директория содержит 2 подпапки, файлы со скриптами, а также jupyter ноутбуки для быстрого прототипирования или наглядной оценки качества, а также служебные файлы.  
Скрипты:

* get\_keywords.py – скрипт для получения ключевых слов при помощи YAKE, запускаемых локально LLM по методу KeyBERT. Для выбора того, какая система будет доставать ключевые слова нужно раскомментировать соответствующие элементы или добавить свои инстансы базового класса KeyWordExtractorBase (например YAKExtractor или KeyBERTExtractor)

в спискок *extractors* на 59 строке

Аргументы для запуска:

-i/--input <папка> - название папки содержащей документы или папки с одним документом. ВНИМАНИЕ: папка расположена по пути “[BASE\_PATH/DOCS\_PATH](#_Переменные_проекта)/<папка> ”. Обязательный параметр

-o/--output <папка> - название папки куда сохранять результаты получения ключевых слов. Необязательный параметр, если не указан то папка будет называться также как и *input.* ВНИМАНИЕ: папка расположена по пути “[BASE\_PATH/DATA\_PATH](#_Переменные_проекта)/<папка> ”.

-n/--number <число> - ограничение числа документов на данный запуск скрипта

-w/--num-of-workers <число> - количество асинхронно работающих задач (полезно когда используется доступ к API) стандартно: 1

--no-rewrite – необязательный параметр, который указывает, что нужно очищать старые результаты экспериментов (расположены в папке output)

--no-skip – необязательный параметр, который указывает, что НЕ нужно пропускать обработку патентного документа, если его результаты для данной системы уже имеются в папке *output*. (Полезно, когда обработка датасета производится в несколько последовательных запусков скриптов, сначала для одной системы затем для других)

* get\_keywords\_llm.py – скрипт который позволяет получать ключевые слова при помощи API больших языковых моделей, а именно YandexGPT, GigaChat, ChatGPT. Позволяет получать базовый термвектор или сразу расширенный тервектор (за один промпт)

Аргументы для запуска:

-i/--input <папка> - см. Описание для get\_keywords.py

-o/--output <папка> - см. Описание для get\_keywords.py

-n/--number <число> - см. Описание для get\_keywords.py

-w/--num-of-workers <число> - см. Описание для get\_keywords.py

--no-rewrite – см. Описание для get\_keywords.py

--no-skip - см. Описание для get\_keywords.py

-m/--model <буква обозначающая модель> - выбор модели (y – YandexGPT, g – GigaChat, c - ChatGPT)

-p/--prompt <имя> - выбор используемого промпта, хранящегося в папке “[BASE\_PATH/PROMPTS\_PATH](#_Переменные_проекта)” в формате json.

* get\_translations.py – скрипт который позволяет использовать проприетарные LLM (GigaChat, YandexGPT, ChatGPT) для расширения/перевода базовых термвекторов, а также для получения из документа базового/расширенного термвектора.

Аргументы для запуска:

-i/--input <папка> - название папки содержащей результаты работы остальных скриптов в JSON. ВНИМАНИЕ: папка расположена по пути “[BASE\_PATH/DATA\_PATH](#_Переменные_проекта)/<папка> ”. Обязательный параметр

* get\_relevant.py – скрипт который позволяет получать результаты поиска по термвекторам полученным в результате исполнения остальных скриптов. Выбор поисковой платформы производится в коде (стандартно выбран ElasticSearch). ВНИМАНИЕ, за один запрос в систему не посылается более 200 терминов, стандартно выдаётся топ-50 самых релевантных документов.

Аргументы для запуска:

-o/--output <папка> - см. Описание для get\_keywords.py. (стандартно: input + ”\_rel”

-n/--number <число> - см. Описание для get\_keywords.py

-w/--num-of-workers <число> - см. Описание для get\_keywords.py

--no-rewrite – см. Описание для get\_keywords.py

--no-skip - см. Описание для get\_keywords.py

-i/--input <папка> - название папки содержащей результаты работы остальных скриптов в JSON. ВНИМАНИЕ: папка расположена по пути “BASE\_PATH/DATA\_PATH/<папка> ”. Обязательный параметр

-t/--timeout <число> - время таймаута для одного запроса в секундах

--wo-kws – параметр, указывающий, требуется ли в файл результатов поиска записывать термвектора, по которым производился поиск (Полезно для уменьшения размера файлов)

--scores – параметр, указывающий, использовать ли веса терминов в поисковой платформе (если таковы имеются в файлах)

# Переменные проекта

В проекте используются различные переменные, которые должны быть прописаны в локальном *.env* файле.

Переменные стандартизирующие пайплайн обработки данных:

* BASE\_PATH – Базовый путь данных, где находятся нижеперечисленные папки (стандартно “data”)
* DOCS\_PATH – Папка где находятся папки с JSON/XML документами (стандартно “data”)
* DATA\_PATH – Папка где находятся папки с результатами экспериментов (в формате json)
* PROMPTS\_PATH – Путь где находятся промты для Больших языковых моделей.
* LOGS\_PATH – Путь где находятся логи выполнения скриптов.

А также переменные содержащие информацию для некоторых классов:

* FIPS\_API\_KEY – API ключ для сервиса «Поисковая платформа»
* ES\_URL – URL сервера ElasticSearch
* CACHE\_DIR – Базовая папка для кэширования
* PROMT\_IP – IP сервера Промпт