Набор утилит для обработки энциклопедических текстов

В общих чертах

Данный набор скриптов позволяет обрабатывать в преимущественно автоматическом режиме отсканированные и оцифрованные тексты советских энциклопедий стандартного формата, по типу *Энциклопедии Математической Физики* издания "*Большая Российская Энциклопедия*" 1998 года. На выходе получается отдельный xml-файл для каждой статьи энциклопедии утверждённого формата (см. далее), пригодного для загрузки в базу данных.

Подробнее

Всю обработку можно разделить на 9 отдельных частей (**автоматическая обработка**, *частично ручная обработка*):

1. *Разметка заголовков статей на отсканированных страницах энциклопедии*
2. *Исправление ошибок в размеченных заголовках*
3. **Составление общего списка размеченных заголовков и присвоение каждой статье уникального URI**
4. **Парсинг текстов статей со страниц энциклопедии согласно разметке заголовков**
5. *Проверка правописания в текстах статей*
6. **Поиск в текстах авторов, работавших над конкретной статьёй**
7. **Поиск в текстах и парсинг ссылок на литературу, использовавшуюся в статье**
8. **Поиск в текстах формул, по отдельности главных (выносных) и дополнительных (строчных)**
9. **Поиск в текстах ссылок на другие статьи (горизонтальных связей)**

В результате для каждой статьи формируется свой xml-файл, со следующей структурой (форматом):

```
URI статьи (пример uri: http://libmeta.ru/fme/article/1_Kraevaya), алфавитная
позиция
        Название статьи
        Авторы статьи
                автор 1
                ...
                автор n
        Страницы
                Начало
                Конец
        Литература
                вся изначальная строка которая парсится
                литература 1
                        автор 11
                        ...
                        автор 1n
                        название литературы 1
                        издательство
                        год
                        прочее
                литература 2
                        автор 21
                        ...
                        автор 2n
                        название литературы 2
                        издательство
                        год
                        прочее
        формулы основные
                формула основная 1 (пример uri:
http://libmeta.ru/fme/formula/main/1_1_Kraevaya)
                ...
                формула основная n
        формулы строчные
                формула строчная 1 (пример uri:
http://libmeta.ru/fme/formula/aux/1_1_Kraevaya)
                ...
                формула строчная n
        Связи типа "смотри также"
                название связанной статьи
                URI связанной статьи
        Текст статьи обработанный
        Текст статьи изначальный
```

Подготовка к работе

Установка необходимого программного обеспечения

1. Для работы скриптов необходимо установить интерпретатор языка программирования Python. Во время установки обязательно включить (поставить галочки) ***pip*** и ***py launcher***.

2. Необходимо установить дополнительные модули для Python.

3. Для этого запустить скрипт ***requirements.py***, установка будет произведена автоматически.

4. Для работы автоматической проверки орфографии необходимо установить словари русского языка. Для этого, в операционной системе Windows:

    1. Распаковать архив ***spellcheck-dicts.zip*** любым удобным способом и достать из папки ***dict-ru/*** файлы ***ru_RU.aff*** и ***ru_RU.dic***.
    2. Нажать клавиши **Win+R**, в открывшемся окне ввести `%AppData%` и нажать "Ок".
    3. В открывшемся окне проследовать по пути `Python / PythonVVV (где VVV -- версия) / site-packages / enchant / data / mingw64 (эта папка может отличаться) / share / enchant / hunspell /`.
    4. Положить файлы ***ru_RU.aff*** и ***ru_RU.dic*** здесь.

5. Для просмотра и изменения настроек вам потребуется текстовый редактор с поддержкой среды разработки Jupyter. Хорошим бесплатным вариантом является редактор Visual Studio Code. При попытке открыть файл с расширениями `.py` и `.ipynb` он должен автоматически предложить установить необходимые расширения. Если этого не произошло, найдите в левой части экрана раздел "Расширения" и установите официальные расширения от Microsoft, воспользовавшись поиском по ключевым словам "Python" и "Jupyter".

Подготовка рабочей области

Необходимо подготовить рабочие папки и некоторые файлы (здесь будут приведены названия папок и файлов, применявшиеся для работы над *Энциклопедией Математической Физики*, вы можете использовать свои):

- Рабочая папка `matphys/`:
    - "Пользовательский" словарь со специфическими именами и терминами `matphys/PWL.txt` (рекомендуется использовать предоставленный

словарь, т.к. он уже содержит много специфических терминов)
    - Папка с исходными оцифрованными текстами `matphys/rpages/`
- Папка для результатов обработки `results/`:
    - Папка для файлов разметки заголовков `results/FMEtitles/`
    - Папка для файлов обработанных статей `results/FMEarticles/`
    - Папка для результатов орфографической обработки текстов статей `results/FMEspellcheck/`

Пояснения к этапам обработки

Достаточные инструкции по работе на каждом этапе находятся в файле ***parsers.ipynb***, нет смысла полностью дублировать их здесь, поэтому будут приведены лишь некоторые дополнительные пояснения для некоторых этапов.

Не забывайте проверять **настройки** скриптов перед запуском (хотя бы перед первым).

Скрипты и инструкции в ***parsers.ipynb*** расположены в предполагаемом порядке использования.

1. Разметка заголовков статей на отсканированных страницах энциклопедии

После запуска вам будут предлагаться для подтверждения предполагаемые заголовки на страницах из выбранного диапазона. Чтобы можно было понять, что именно считается заголовком, он подчёркивается. Например:

```
...какой-то текст.$$ПРИМЕР з а г о л о в к а - текст статьи...
                   ^^^^^^^^^^^^^^^^^^^^^^^^
```

Ваша задача -- удостовериться, что границы заголовка были определены правильно, и поправить их предложенными средствами при необходимости. Границы статей будут напрямую определены по границам заголовков, поэтому важно знать их точно.

Как в приведённом примере, большая часть загловков состоит из основной части, написанной заглавными буквами, и "дополнительной", которая чаще всего

написана в посимвольном стиле, с пробелом после каждой буквы, и которую тоже стоит включить в разметку. Учтите это.

Рекомендуется держать открытым оригинальный pdf-скан энциклопедии и сверять очередной предлагаемый загловок с ожидаемым. Это поможет избежать добавления лишних "заголовков" и (особенно!) случаев пропуска заголовков по различным причинам.

На случай форс-мажорных ситуаций (заголовок был распознан как изображение и от него осталась лишь ссылка, слишком мало заглавных букв и он не был распознан или же вдруг не получается достаточно точно, полюс-минус до знаков пунктуации, указать границы и т.д.) предусмотрена возможность добавлять заголовки по одиночке, с помощью скрипта 1.1.. Добавлять их в общие файлы, либо в отдельный (см. **настройки**) -- на Ваше усмотрение.

2. Исправление ошибок в размеченных заголовках

Помимо того, что заголовок -- это первое, что увилит пользователь, корректная работа поисковых систем в базах данных также будет затруднена при наличии в них ошибок.

После выполнения скрипта 2.1. для каждого заголовка можно во второй строке просмотреть результаты попытки заменить латиницу на схожую по начертанию кириллицу, попытки "срастить" слова, написанные в посимвольном стиле и т.д.. Также в третьей строке приведена попытка дополнительно применить коррекцию орфографии (виду ошибок в расставлении пробелов она выдаёт много лишних исправлений, поэтому вынесена отдельно, на усмотрение). Пример:

```
<title_old>АБЕJЕВА ФУНКНИА</title_old>
<title_new>АБЕЛЕВА ФУНКНИА</title_new>
<title__sc>_______ ФУНКЦИЯ</title__sc>
```

(Примечание: если проверка орфогафии пытается исправить очевидно верное слово, то можно добавить это слово в "персональный" словарь, на будущее.)

Здесь нужно исправить все ошибки в заголовках, расставить проблелы как необходимо, проверить орфографию. Финальный вариант заголовка должен

лежать в <title_new>.

После окончания проверки и выполнения скрипта 2.2. рекомендуется запустить скрипт 2.1. в режиме "только проверка орфографии" на получившемся результате. Это может помочь обнаружить случайные пропуски проверки, т.к. скорее всего орфография будет на них ругаться, и это будет видно в третьей строке. После данной "повторной" проверки нужно снова запустить скрипт 2.2., разумеется.

5. Проверка правописания в текстах статей

В результате работы сканера на выбранном диапазоне статей можно будет просмотреть все места в текстах, которые проверка орфографии посчитала подозрительными. Чаще всего, они относятся к одному из следующих видов:

1. Слова с ошибками
2. Новые термины
3. Новые имена собственные
4. Слова, внутри которых оказался вставлен пробел, слова на переносе между страницами и т.п..

Первые нужно просто исправлять, вторые и третьи -- добавлять в словарь (оригинальный или исправленный вариант по необходимости), а четвёртые чаще всего можно даже просто удалять (жалко, что они останутся неисправленными, но в конечном счёте они не так сильно портят читаемость, а придумать алгоритм, определяющий, что и с чем нужно срастить -- весьма нетривиальная задача).

Пример найденного слова с ошибкой:

```
<word pos="2004" result="1" add_to_pwl="0">
  <source>группы</source>
  <context>, g\right\}-$ базис группы гомологий поверхнос</context>
  <suggestion>группы</suggestion>
</word>
```

ВАЖНО! Вносить изменения следует только в <suggestion>. Внесение изменений в <source> может повлечь неправильную подстановку исправлений в текст.

8. Поиск в текстах формул, по отдельности главных (выносных) и дополнительных (строчных)

Может возникнуть два вопроса:

1. Зачем нужно выносить формулы в отдельный файл (скрипт 8.1.), разве это не дублирование результатов?
2. Что означает "проверка формул" (скрипт 8.2.)?

Общий сборник всех формул может пригодиться для отдельной области исследований, анализа формул, поэтому на всякий случай они отделяются от текстов (сохраняется только uri для установления взаимосвязей) и складываются в один "удобный" файл.

Чтобы удостовериться в правильности распознавания формул после сканирования бумажного оригинала, был сделан скрипт, который случайным образом выбирает указанное число главных формул и помещает их в математическое окружение внутри Markdown. Дополнительно указывается приблизительное место расположения формулы, чтобы правильность можно было сверить визуально со сканом. Для открытия данного md-файла нужен какой-либо просмотрщик Markdown с поддержкой рендеринга katex. Visual Studio Code, который был рекоментдован к установке так умеет (если не по умолчанию, то по крайней мере при наличии пары дополнений, которые не должно составить проблем найти и установить).

9. Поиск в текстах ссылок на другие статьи (горизонтальных связей)

Несмотря на то, что скрипт в файле ***parsers.ipynb*** и файл ***relations.py*** в сущности представляют собой одно и то же, есть некоторые нюансы по их применению.

Окружение Jupyter Notebook не позволяет запускать параллельные (многоядерные) вычисления (а попытка это сделать может, вероятно, привести к необходимости перезагружать компьютер и/или порче готовых данных!), поэтому скрипт в ***parsers.ipynb*** по умолчанию перенастроен на упрощённый

непараллельный поиск по ключевым словам, в то время как ***relations.py*** по умолчанию запускает несколько процессов сложного детального поиска (и даёт в разы больший результат).

Настройки

Во всех скриптах есть блок настроек, обозначенный как `----- VARS -----`.

Вы можете менять эти настройки по своему усмотрению. Однако, если вы не понимаете, что означает та или иная настройка, не у кого уточнить, и нет возможности разобраться в этом самостоятельно, оставьте как есть -- последний раз, когда кто-то проверял, всё работало.

Рассмотрим настройки скриптов, отдельно для каждого файла и утилиты (здесь будут приведены названия папок и файлов, применявшиеся для работы над *Энциклопедией Математической Физики*, вы можете использовать свои)

Файл *lib.py*

Содержит функции, общие для многих утилит, и содержит общие (не локальные) настройки:

- `COMBINATIONS_CORR_ALPHABET` -- словарь, использующийся для корректировки неверно распознанных кириллических символов, изначально распознанных как латиница. Скорее всего, изменять его не придётся.
- `COMBINATIONS_CORR_UNICODE` -- словарь, похожий на предыдущий, но использующийся для корректировки кириллических символов, распознанных как специфические символы из таблицы Unicode. Вам вполне могут таковые встретиться, и вы можете их сюда добавить.
- `COMBINATIONS_CORR_OTHER` -- словарь, похожий на предыдущие, но использующийся для корректировки в специфических ситуациях. Скорее всего, менять его не придётся.
- `XML_EXCLUDES` -- служебный словарь, используется для чтения xml-файлов. Его изменять не нужно.
- `PERSONAL_WORD_LIST` -- "Пользовательский" словарь с со специфическими именами и терминами. Пример: `"./matphys/PWL.txt"`.

- `URI_PREFIX` -- префикс, использующийся для формирования uri и url в ходе обработки. Пример: `"http://libmeta.ru/fme/"`.

Файл *base_titles_parser.py*

Является утилитой для разметки заголовков. Локальные настройки:

- `PAGES_DIR` -- указание на папку с исходными оцифрованными текстами. Пример: `"./matphys/rpages/"`.
- `EXIT_DIR` -- указание на папку, в которую будет помещён предварительный результат разметки заголовков. Пример: `"./matphys/"`.
- `EXIT_FILE` -- имя файла для записи предварительных результатов разметки заголовков. Пример: `"FMEv2.xml"`.
- `START_PAGE`, `END_PAGE` -- первая и последняя страница для обработки. Используйте, чтобы не обрабатывать всё за один раз.
- `LEAD_WORDS`, `AFT_WORDS` -- для облегчения разметки, с обеих сторон от обнаруженного заголовка выводятся несколько слов для обозначения контекста. Можно изменить их количество.
- `CAPS_QUOT` -- число от 0 до 1, определяющее долю заглавных букв в слове, чтобы оно считалось частью заголовка.
- `EXCEPTIONS` -- список исключений к предыдущему правилу. Например, римские числа и обозначения физических величин.

Файл *relations.py*

Является утилитой для поиска ссылок на статьи (горизонтальных связей). Локальные настройки:

- `ARTICLES_DIR` -- указание на папку для файлов обработанных статей. Пример: `"./results/FMEarticles/"`.
- `STRICT_SEQUENCING` -- отвечает за распознавание слов в последовательностях в произвольном порядке. При значении по умолчанию (`False`) последовательность слов, к примеру, "*интеграл Лебега*" и "*Лебега интеграл*", будет считаться одной и той же. Скорее всего, менять его не придётся.
- `BRUTE_FORCE_MODE` -- отвечает за детальное сканирование текста. Значение по умолчанию: `True`, в противном случае будут распознаны только ссылки по

ключевым словам типа "*См. также ...*". Скорее всего, менять его не придётся.
- `USE_MULTIPROCESSING` -- отвечает за использование многоядерного ускорения вычислений. Значение по умолчанию: `True`. Скорее всего, менять его не придётся.
- `KEEP_FREE` -- если вы не хотите, чтобы при вычислениях задействовалась вся доступная вычислительная мощность, можете указать, чтобы определённое число логических процессоров в вашей системе оставалось свободным. Значение по умолчанию: `0`.

Основной файл *parsers.ipynb*

Является сборником большинства утилит для обработки, представленные в формате **Jupyter Notebook** в порядке, предполагаемом для использования:

1. Базовый парсер заголовков

См. файл ***base_titles_parser.py***, а также:

1.1. Добавление заголовков по одному:

- `PAGES_DIR` -- указание на папку с исходными оцифрованными текстами. Пример: `"./matphys/rpages/"`.
- `EXIT_DIR` -- указание на папку, в которую будет помещён предварительный результат разметки заголовков. Пример: `"./matphys/"`, `"./results/FMEtitles/"`.
- `EXIT_FILE` -- имя файла для записи предварительных результата разметки заголовка. Пример: `"FMEv2.xml"`, `"FMEtitles-added-manually.xml"`.
- `PAGE` -- страница, на которой будет производиться поиск указанного отдельного заголовка.
- `TITLE` -- точная формулировка заголовка, который должен быть извлечён из текста.

2. Исправление ошибок в заголовках:

2.1. Составитель пар "оригинальный - исправленный" для заголовков:

- `WORK_DIR` -- указание на папку, в которую был помещён предварительный результат разметки заголовков. Пример: `"./matphys/"`.

- `INPUT_FILE` -- указание на файл с результатами предварительной разметки заголовков. Пример: `"FMEv2.xml"`.
- `CORRECTION_FILE` -- имя файла, в который будут записаны результаты предварительной обработки и исправления ошибок в заголовках. Пример: `"FMEcorr.xml"`.
- `COMBINATIONS_CORR` -- словарь, добавляющий обработку некоторых часто встречающихся в заголовках сочетаний символов.
- `SPELLCHECK_ONLY` -- позволяет включить режим, задействующий при обработке заголовков исключительно проверку орфографии.

2.2. Подстановщик исправленных заголовков:

- `WORK_DIR` -- указание на папку, в которую был помещён предварительный результат разметки заголовков и результат их обработки. Пример: `"./matphys/"`.
- `INPUT_FILE` -- указание на файл с результатами предварительной разметки заголовков. Пример: `"FMEv2.xml"`.
- `CORRECTION_FILE` -- указание на файл с результатами обработки и исправления ошибок в заголовках. Пример: `"FMEcorr.xml"`.
- `EXIT_FILE` -- имя файла для записи финального варианта размеченных и обработанных заголовков. Пример: `"FMEtitles.xml"`.

3. Сортировщик / сливщик файлов заголовками:

- `WORK_DIR` -- указание на папку, в которую будет записан результат, а также содержащую папку с файлами размеченных и обработанных заголовков. Пример: `"./results/"`.
- `TITLES_DIR` -- указание на папку внутри `WORK_DIR`, содержащую папку с файлами размеченных и обработанных заголовков. Пример: `"FMEtitles/"`.
- `INPUT_FILES` -- список всех файлов в `TITLES_DIR` с размеченными и обработанными заголовками, из которых будет формироваться результат. Пример: `["FMEtitles-p5-100.xml", ...]`.
- `MANUALLY_ADDED_FILE` -- указание на файл в `TITLES_DIR` с дополнительными (по задумке добавленных индивидуально с помощью скрипта 1.1.) размеченными и обработанными заголовками. Пример: `"FMEtitles-added-manually.xml"`.
- `URI_CACHE` -- имя файла в `TITLES_DIR` для хранения кеша назначенных URI. Пример: `"FMEtitles-uri-cache.xml"`.

- `EXIT_FILE` -- имя файла в `WORK_DIR` для записи результата. Пример: `"FMEtitles-merged.xml"`.
- `URI_SAFER` -- включает защиту от изменения URI, значение по умолчанию: `True`. Скорее всего, менять его не придётся.

4. Парсер текстов статей:

- `TITLES_FILE` -- указание на файл с размеченными и обработанными заголовками с присвоенными URI. Пример: `"./results/FMEtitles-merged.xml"`.
- `PAGES_DIR` -- указание на папку с исходными оцифрованными текстами. Пример: `"./matphys/rpages/"`.
- `EXIT_DIR` -- указание на папку, в которую будут сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.
- `COMBINATIONS_CORR` -- словарь, добавляющий обработку некоторых часто встречающихся в текстах сочетаний символов.

5. Проверка правописания в текстах:

5.1. Сканер:

- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.
- `EXIT_DIR` -- указание на папку, в которую будет сохранён предварительный результат орфографической обработки текстов статей. Пример: `"./matphys/"`.
- `CONTEXT_SIZE` -- для облегчения обработки пользователю предоставляются не только изначальное слово и предложение его исправления, но и строка для контекста, с отступом в обозначенное количество символов влево и вправо.
- `DEFAULT_RESULT_FLAG`, `DEFAULT_ADD_TO_PWL_FLAG` -- значения по умолчанию для флагов исправления слова:
  - Результат: изначальный -- 0 или исправленный -- 1
  - Добавление в словарь: не добавлять -- 0, добавить как есть -- 1, добавить в словарь в нижнем регистре (в тексте не изменяется) -- 2 или сделать первую букву заглавной и добавить (затрагивает также и текст0 -- 3
- `OVERRIDE_FORCE_CYRILLIC` -- словарь слов **написанных кириллицей** для приоритетной обработки. Эти слова будут изменены строго как указано. В

слове-кандидате все, какие возможно, латинские символы будут распознаны как кириллические (т.е. лат. *Cccp* и кир. *Cccp* оба будут считаться **одинаковыми**).
- `OVERRIDE_AS_IS` -- то же самое, но и предыдущий, но попытка перевести латиницу в кириллицу произведена **не будет** (т.е. лат. *Cccp* и кир. *Cccp* оба будут считаться **разными**).

5.2. Пополнение словаря:

- `SPELLCHECK_DIR` -- указание на папку, в которой находятся окончательные результаты орфографической обработки текстов статей. Пример: `"./results/FMEshellcheck/"`.

5.3. Подстановка исправленной орфографии:

- `SPELLCHECK_DIR` -- указание на папку, в которой находятся окончательные результаты орфографической обработки текстов статей. Пример: `"./results/FMEshellcheck/"`.
- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.

6. Парсер авторов статьи:

- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.
- `COMBINATIONS_CORR` -- словарь для исправления возможных ошибок в именах авторов, преимущественно ошибки unicode.

7. Парсер литературы:

- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.
- `COMBINATIONS_CORR_LOCAL` -- словарь исправлений, необходимый, чтобы улучшить распознавание границ сегмента с литературой в текстах.

8. Парсер формул:

- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.

- `MIN_INLINE_LEN` -- минимальная длина дополнительных (строчных) формул. Значение по умолчанию: `0`. Используйте, чтобы, например, исключить из списка формул одиночные символы в математическом окружении LaTeX.

8.1. Вынос формул:

- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.
- `EXIT_FILE` -- имя файла, в который будет записан общий список формул со всех статей. Пример: `"./results/FMEformulas.xml"`.

8.2. Проверка формул:

- `ARTICLES_DIR` -- указание на папку, в которую сохранены индивидуальные xml-файлы статей. Пример: `"./results/FMEarticles/"`.
- `EXIT_FILE` -- имя файла, в который будут записаны формулы для проверки и оценки. Пример: `"./matphys/FMEformulas_check.md"`.
- `NUMBER` -- количество записываемых для проверки и оценки формул.

9. Парсер ссылок типа "смотри также"

См. файл ***relations.py***. Многоядерные вычисления не работают в оболочке Jupyter Notebook, поэтому настройки, отвечающие за его включение и за детальный поиск здесь переопределяются.