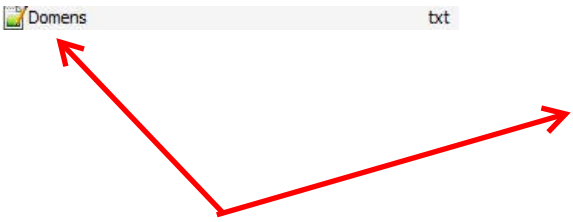
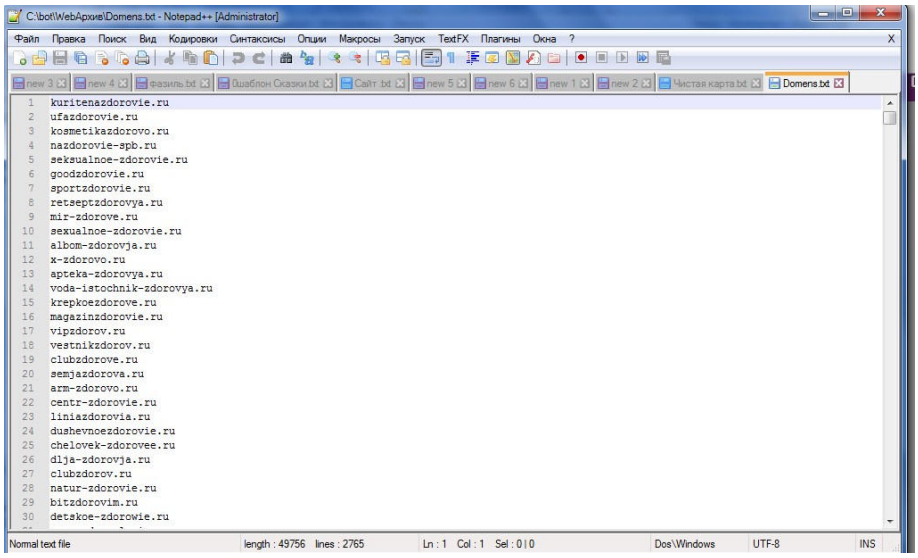
WebArchiveMaster - программа парсинга контента из ВебАрхива. Программа полностью автоматизирована и позволяет разгрузить своё время на 90%. Программа работает в связке с PHP скриптом, который можно поставить на любой хостинг или использовать **Open Server - https://ospanel.io (рекомендуется).**

Принцип работы

Принцип работы очень прост - нужно только вставить домены в текстовый файл и запустить программу - все остальное она сделает сама. Никаких настроек нет, так-как все настроено на максимальную производительность. Разбирем на примере:



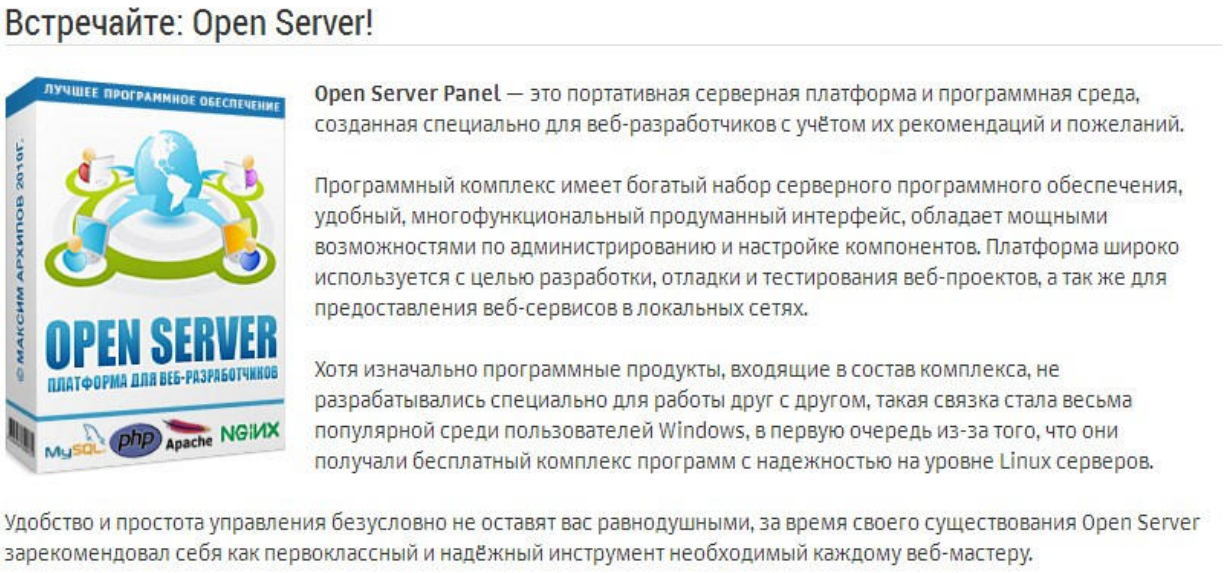


Скопируйте домены в файл: Domens.txt, запустите программу и можете отдыхать.

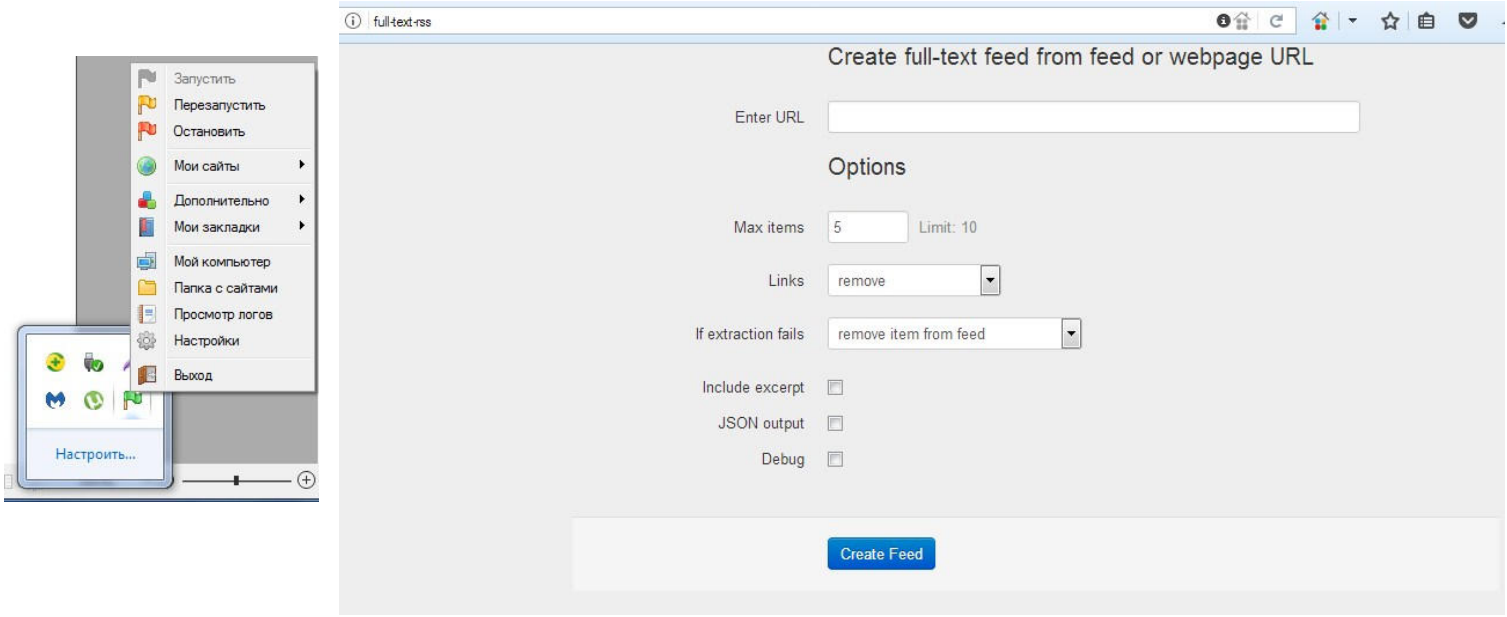
Директория должна находиться по адресу: **c:\bot\WebАрхив**

Установка скрипта

Разберем установку бесшаблонного парсинга - скачиваем **Open Server.**



Запускаем локальный сервер



Запускаем сервер, после запуска вставляем в браузер название скрипта (база данных не требуется) и все, программа готова к работе. Точно также можно установить на хостинг - просто копируете скрипт на домен или поддомен и все готово к работе.

Во входящих настройках установить путь к скрипту. Если вы установили на поддомене **http://feed.cheerfulness.ru**, то так и пишите, если на локальном сервере, то пишите: **http://full-text-rss/**. Скрипт **full-text-rss**, находящийся в папке, нужно перенести на локальный сервер или хостинг. Программа будет к нему обращатся для парсинга.

Принцип работы

Как работает программа - берет выборочно домен и проверяет его на ответ 200 (сайт работает). Если сайт работает, домен удаляется и берется следующий. После получения нужного домена, программа подключается к Вебархиву и запрашивает количество файлов за все годы (не по снепшотам). Если файлов нет, возвращается к выбору другого домена. Если файлы есть, программа забирает ссылки и включает фильтрацию (css, png, jpg, reply и т.д.).

После этого чистит ссылки и включает скрипт скрапинга, забирает текст и начинает его чистить от всего мусора, тегов и т.д.

Программа пишет все статьи в один файл без заголовков, и этому есть причины - все это отработана неделями тестирования и выбран лучший вариант среди тысяч - разберем некоторые:

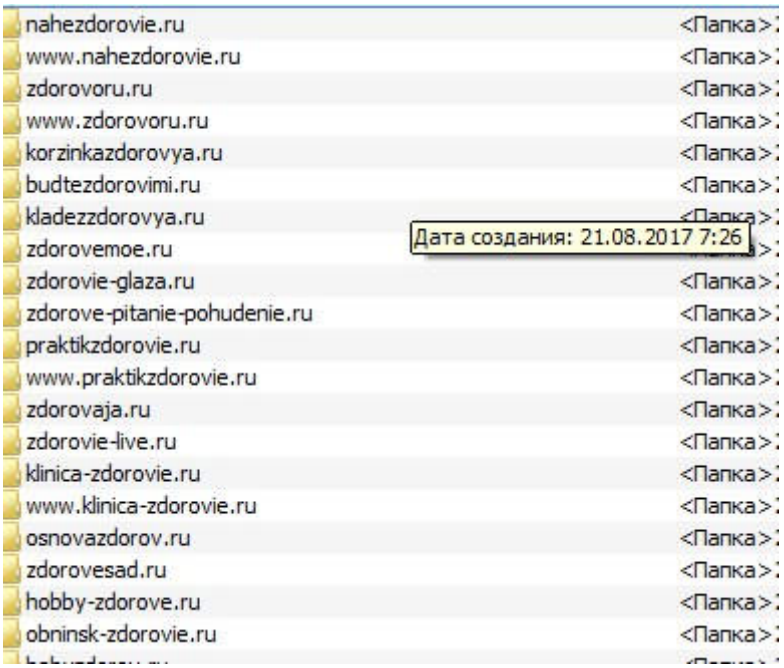
Парсить приходится самые различные системы управления контентом, как cms, так и обычные сайты и фреймворки и самоделки, в которых просто нет зацепок для программы, типа Title или H1 - их просто может не быть. Поэтому программа работает так - берет текст, фильтрует его, пишет в один файл, затем удаляет дубли (здесь ещё один из тысяч подводных камней - сайы имеют неявные дубли, и одна страница может открываться как по адресам: /page&?p233, так и по /ozdorovlenie/ и по /ozdorovlenie.html, к тому же стоят редиректы и другие всевозможные перенаправления.

Это одна и та же страница, и это создает очень большие проблемы не только для поисковых систем.

Поэтому все пишется в один файл и после того, как все страницы скачены, программа удаляет все дубли и затем каждую страницу сохраняет в отдельный файл. Это нужно для массовой проверки через антиплагиат - я использую **eTXT Антиплагиат**, она позволяет использовать пакетную проверку хоть тысячи файлов. Для капчи я использую **XEvil.**
Вот как это выглядит (готовый сайт):

Готовая статья19	txt	5 750	21.08.2017 07:37
Готовая статья18	txt	11 401	21.08.2017 07:37
Готовая статья17	txt	12 018	21.08.2017 07:37
Готовая статья16	txt	10 086	21.08.2017 07:37
Готовая статья15	txt	2 916	21.08.2017 07:37
Готовая статья14	txt	19 867	21.08.2017 07:37
Готовая статья13	txt	2 992	21.08.2017 07:37
Готовая статья12	txt	2 188	21.08.2017 07:37
Готовая статья11	txt	4 512	21.08.2017 07:37
Готовая статья10	txt	3 804	21.08.2017 07:37
Готовая статья9	txt	10 309	21.08.2017 07:37
Готовая статья8	txt	12 723	21.08.2017 07:37
Готовая статья7	txt	4 243	21.08.2017 07:37
Готовая статья6	txt	4 195	21.08.2017 07:37
Готовая статья5	txt	19 689	21.08.2017 07:37
Готовая статья4	txt	11 733	21.08.2017 07:37
Готовая статья3	txt	3 837	21.08.2017 07:37
Готовая статья2	txt	19 647	21.08.2017 07:37
Готовая статья1	txt	2 570	21.08.2017 07:37
Статьи в одном файле	txt	266 643	21.08.2017 07:37
Все текстовые данные	txt	267 542	21.08.2017 07:37

Все статьи сохраняются в папку с названием домена, с которого они были скачены. Это сделано для того, чтобы, если статьи понравятся, можно попытаться восстановить дроп.



Что делать со статьями

Расскажу про свою методику. После того, как скачено около тысячи текстов, я выбираю 500-600 статей по размеру (3 - 15 тысяч символов одна статья, остальные сбрасываю в резервную папку для саттелитов или дорвеев), пакетно загружаю в **eTXT Антиплагиат** и запускаю проверку на уникальность. Я ставлю настройки 80% уникальности и антиплагиат сам раскидывает их в разные папки - прошедшие уникальность и не прошедшие.

Затем я за копейки покупаю на Телдери старый трастовый сайт 2-3 лет, который давно не обновлялся и работает в убыток и публикую на нем статьи. Статьи очень хорошо заходят и сидят в выдаче, многие мои сайты были приняты во все биржи, некоторые в РСЯ. На молодом сайте так делать опасно, так как статьи инициированы, и скорее всего яндекс про них знает- уникальность позволяет только определить, что этих статей нет на других сайтах.

Продавал статьи на бирже и не имел ни одного отрицательного отзыва, но жадность сгубила и на объемах биржа спалила, из-за того, что кто-то еще продавал эти же статьи. Но деньги успел вывести, неплохую сумму. Так что поакуратнее, даже если статья показывает 100% уникальности на всех сервисах антиплагиата, не факт, что вас не забанят при загрузке статьи на биржу, т.к. у них своя база и каждую загруженную статью они сравнивают, не было ли такой ранее.

Где взять брошенные домены

Брошенные домены можно взять на expireddomains.net. Регистрируетесь, вводите ключевое слово в поиск, например, если нужны домены о здоровье, то пишите: zdo или zdr и скачиваете домены списком.

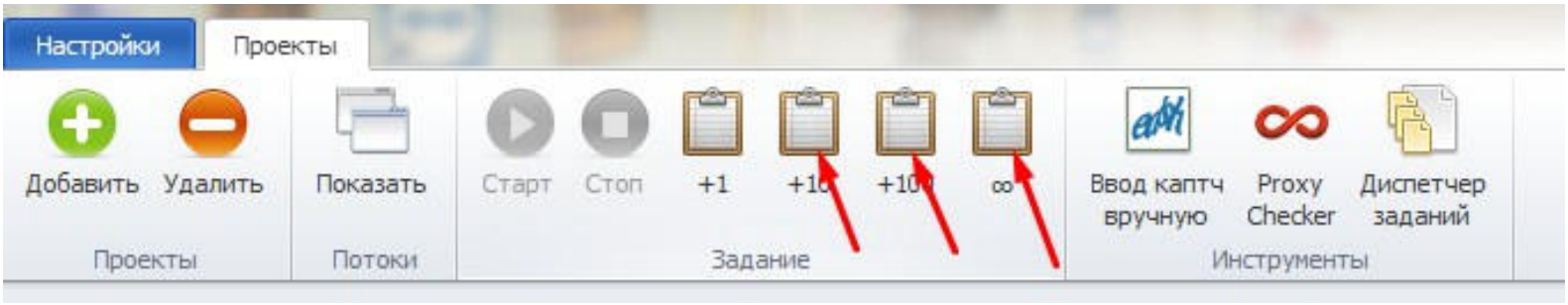
Вот пример работы программы за два часа - скачено около 300 текстов, выборочная проверка текста показала, что уникального контента очень много.

3. Теперь все данные сохраняются в одну папку, без "www"
4. Отрегулирован PHP скрипт, но мусор все равно будет цеплять - если текст небольшой, а данных на странице много (комментарии, рекламные слоганы, которые бывают больше текста), то неизбежно бесшаблонный парсер захватит их. Если текст чистый более-менее, то всё ненужное отсечётся.

- Названия файлов для подготовки проверки на уникальность теперь имеют название домена
- Уменьшена жесткость фильтрации
- Изменен алгоритм, теперь идет дополнительная проверка по снепшотам, если Архив не дает файл
- Теперь есть докачка текстов при сбое
- Пофиксены недочеты и мелкие ошибки

В файл **temp_domen.txt** помещается домен на случай сбоя и парсинг начинается с него. Если домен не нужен, удалите его и шаблон будет брать домены из файла **Domens.** Сам файл **удалять нельзя,** т.к. шаблон использует его. Файл **"Чистая карта.txt"** отвечает за парсинг и восстановления после сбоя, при запуске Зеннопостера - он начинает работать там, где был прерван. Если хотите использовать другой домен, **удалите файл** Чистая карта.txt.

Также нужно добавлять задания к выполнению. Это сделано для того, чтобы избежать избыточной цикличности и сбросить все данные при завершении работы.

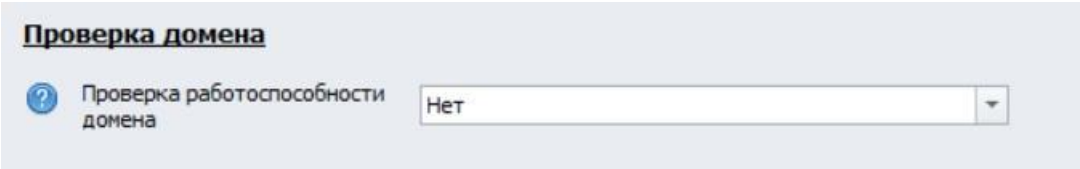


Внимание!

Если вы запустили домен на проверку, но потом передумали его скачивать, вам нужно удалить служебный файл **Чистая карта.txt** (он отвечает за докачку данных при сбое или ошибке) и файл **temp_urls.txt** (он отвечает за сбор ссылок) - если вы остановили на поздней стадии, то этого файла не будет, так-как он промежуточный. Не перепутайте с **temp_domen.txt,** это служебный файл для резервирования текущего домена.

Настройка скрипта - путь к скрипту на сервере или локальном сервере.

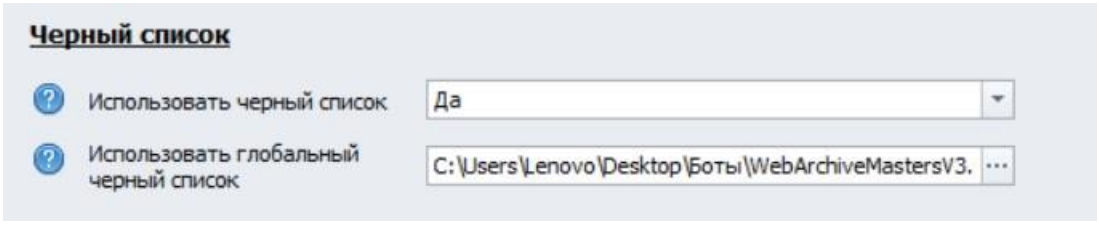
Если вы используете Опен Сервер, должна быть такая ссылка, ведущая на скрипт-обработчик: http://full-text-rss. Устанавливать скрипт нужно по примерно такому пути (у вас он может отличаться) - C:\server\OSPanel\domains\full-text-rss. Если вы хотите использовать скрипт на домене или поддомене, установите путь к скрипту, например - http://script.site.com или http://site.com.



Проверка домена - запрос домена на ответ 200.

Проверка работоспособности домена - если вы используете ручную проверку доменов и знаете, что они неработоспособны, можете поставить "нет", если используете список и не знаете, работает домен или нет, ставьте "да" - это позволит не захватить работающий домен.

По умолчанию проверка отключена, так-как я использую ручную проверку.



Черный список - путь к глобальному файлу "blacklisting"(должен быть один для всех).

В черном списке находятся уже проверенные домены. Он нужен, чтобы не проверять домены, с которых уже взят текст по второму кругу.

Использовать глобальный черный список – здесь указываете путь к файлу «blacklisting.txt» - именно туда записываются отработанные домены. Если список не указан, шаблон остановит работу, пока вы не укажете правильный путь.

Фильтрация - если не хватит памяти для обработки, домен перезапишется и возьмется другой. Для слабых компьютеров поставьте секунду задержки.

При фильтрации удаляются ссылки, которые имеют признаки: js, css, ico и т.д. Это внутренняя фильтрация, и самостоятельно делать ничего не нужно, иначе можно отсечь нужные файлы.

Перезагрузка инстанса – установка своего значения.

Во время работы шаблон использует ресурсы браузера. Данные копятся и скорость понижается, а запросы к ресурсам компьютера растут. Но при перезагрузке инстанса, чтобы сбросить все данные, он может зависнуть или встать в бесконечную загрузку. Вы можете не использовать перезагрузку инстанса, для этого установите значение 999999. Либо вы можете увеличить количество циклов до перезагрузки (по умолчанию – 75), если памяти компьютера более 4 Гб.

Первичная проверка - проверка присутствия текста на странице.

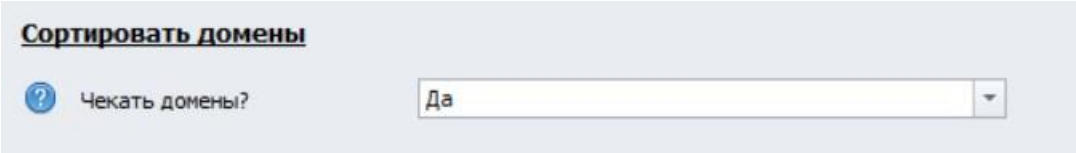
Если количество текста на странице меньше значения (по умолчанию 200 символов), то пойдет перепроверка через снепшот Вебархива и будет выведена информация «Стандартно текст не найден. Запуск перепроверки. Скорость будет снижена». Значение лучше не менять.

Проверка на дорвей – проверка количества файлов.

Например, на небольшом сайте возьмется 5000 файлов, после фильтрации отбросится мусор (скрипты, стили и т.д.), получится 400 ссылок на текст, после парсинга выйдет примерно 150-200 текстов. Для отключения используйте 999999, так-как сайты с большим количеством файлов будут считаться дорвеями и домен запишется в файл «Dorvey.txt» для ручной проверки этого домена и возьмется следующий домен для проверки.

Парсинг текста – минимальное количество текста.

Здесь нужно указать минимальное количество символов, которое должно быть в статье. Если нужны большие статьи, можно установить 1000-2000 символов (не слов), статьи, содержащие меньше, отбросятся.



Чекать домены – проверка спаршенных доменов.

Если включен режим забирать домены во время парсинга текста, то после полной отработки текста начнется проверка работоспособности доменов из файла «Спаршенные домены» и их сортировка. Они будут перемещены в папку «verified domains» и рассортированы. Выглядит это так:

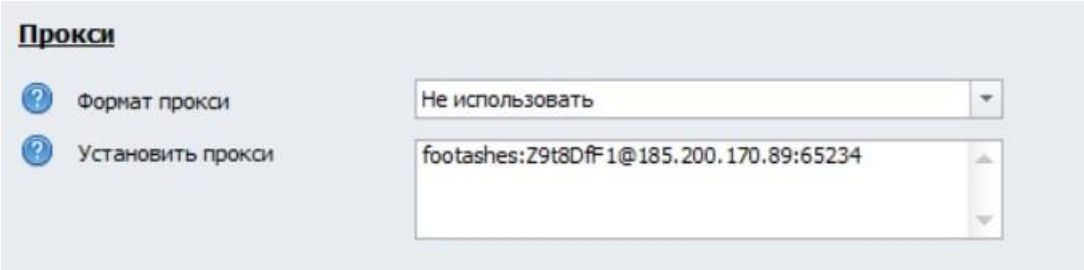
Проверку доменов можно остановить в любой момент, так-как весь текст к этому времени почищен и рассортирован.

Использовать даты – даты по годам.

Если нужно скачать данные по годам, можно использовать в таком виде:

2011
2012
2013
2014

Скачаются ссылки только за эти даты. Если не требуется, то оставьте поле пустым. Тогда скачаются все тексты. Это полезно, если на этом домене дорвей, а ранее был хороший сайт либо нужен текст за определенный год.

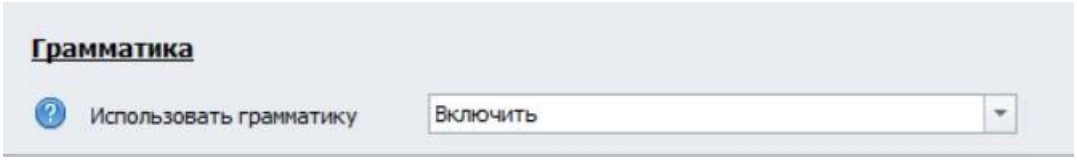


Формат прокси – установить прокси.

Если доступ к Вебархиву ограничен или запущено много краулеров, можно использовать прокси. Прокси используются в формате Socks или Http/Https.

Формат прокси:

http://username:password@ip:port или username:password@ip:port
socks5://username:password@ip:port



Грамматика – использовать грамматику.

Используется для приведения текста в порядок. Убирает двойные\тройные пробелы, пробелы перед запятыми\точками\троеточиями и т.д.

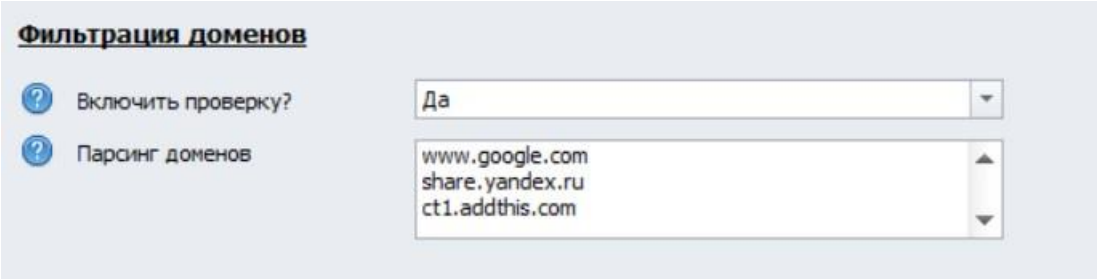
Было:

Мама мыла раму .

Стало:

Мама мыла раму.

Не исправляет ошибки в словах.



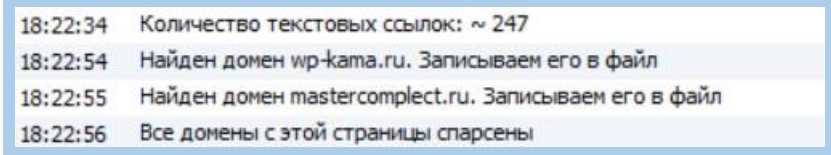
Фильтрация доменов – парсинг дропов со страниц.

Это паук, бывший «спайдер», ставший потом «ParserDomens» и теперь внедренный в «WebArchiveMastersV3.0».

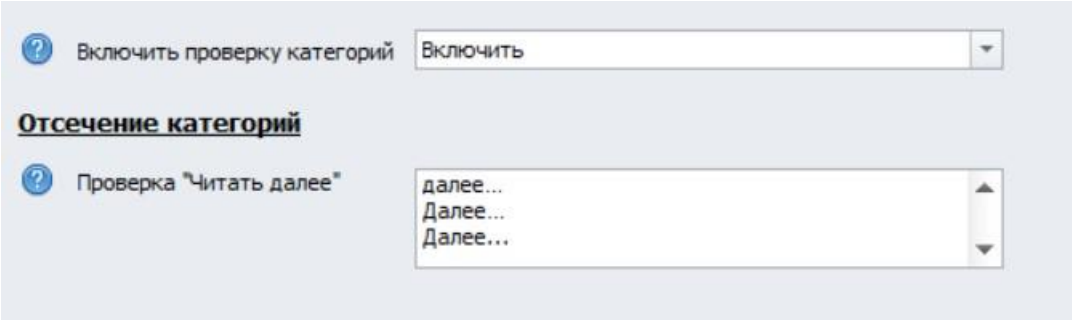
При парсинге текста одновременно забирает все домены со страницы (из комментариев, чатов и т.д.). Работает, используя фильтрацию по ненужным доменам:

www.google.com
share.yandex.ru
ct1.addthis.com
stg.odnoklassniki.ru
nolix.ru
connect.mail.ru
cdn.connect.mail.ru
twitter.com…
…и другим. Рекомендуется периодически добавлять те домены, которые не нужны, и они будут пропускаться. Немного замедляет парсинг текста и дает лишний запрос к Вебархиву. Используется для автоматизации методики поиска брошенных доменов.

Данные пишутся в файл «Спаршенные домены.txt» и выводятся в лог:



Если домены парсить не нужно, можете отключить проверку и поиск.

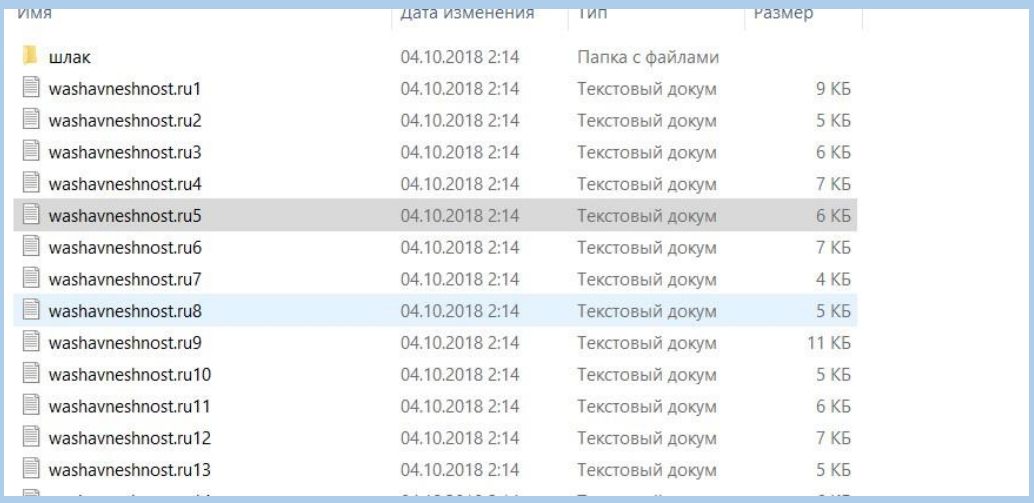


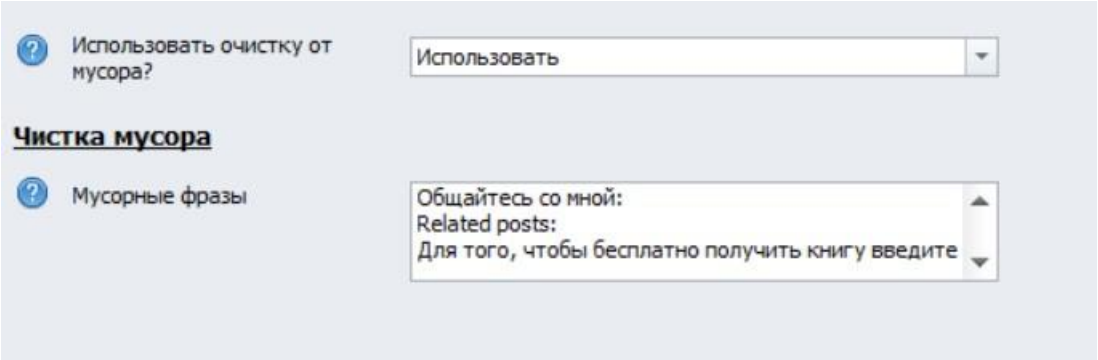
Включить проверку категорий – проверка на присутствие категорий.

Принцип работы – ищет в тексте признаки категорий. Признаки находятся в черном списке «Проверка Читать далее». Вот некоторые из них (рекомендуется добавлять):

далее...
Далее...
Далее...
Далее.. »
Читать дальше
читать далее
читать полностью
read more
Read more
Read More
Read more…
Read the rest of this entry
Читать продолжение записи
Прочитать остальную часть записи
[...]
continue reading

Если хоть одно это слово будет найдено в тексте, текст отправляется в папку «шлак» этого домена. Текст всегда можно посмотреть и при ошибочном срабатывании переместить обратно. Примерная структура после сортировки:





Использовать очистку от мусора – мусорные фразы.

Используется для удаления текста после определенных фраз:

Общайтесь со мной:
Related posts:
Также на эту тему можно почитать:
Поделиться в соц. сетях
Хочешь первым узнать о новых статьях на этом сайте? Внеси своё имя и почту и нажми получить.
Related Posts
Советую также почитать:
Понравилась статья? Поделись с друзьями:
Посмотрите другие записи:
Подпишитесь на рассылку:

Такие фразы идут в конце статьи и не нужны, так как создают информационный шум при проверке на уникальность. Если найдется такая фраза, текст, идущий после этой фразы, будет удален с этой фразой. Используйте аккуратно и осторожно.