ОЧИСТКА ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ APACHE SPARK

Д.А. Усталов[1,2], П.А. Блинов[2], М.А. Черноскутов[1,2]

[1] *Институт математики и механики им. Н.Н. Красовского Уральского отделения Российской академии наук*
[2] *Уральский федеральный университет имени первого Президента России Б.Н. Ельцина*

Введение. При работе с крупными коллекциями документов, представленными в виде неструктурированного текста на естественном языке, чрезвычайно актуален вопрос поддержания таких данных в качестве, достаточном для их эффективной обработки и хранения. В частности, данные могут иметь авторские опечатки, автоматически экранированные символы, наличие следов специализированной разметки, и многие другие дефекты. Особенно это актуально при работе с текстами, полученными из неконтролируемых источников, таких как Всемирная Паутина или каких-либо крупномасштабных краудсорсинговых проектов. Примерами таких проектов могут быть Википедия и Викисловарь.

В данной работе представлен и апробирован подход к выполнению очистки лингвистических данных на примере очистки базы данных проекта по созданию открытого электронного тезауруса русского языка Yet Another RussNet с использованием инструментальной среды Apache Spark. Новизна и практическая ценность данной работы состоит в применении легковесного инструментария Apache Spark к обработке текстовых данных с определённым видом разметки.

Набор данных. Данные тезауруса Yet Another RussNet на уровне их схемы разделяются на «сырые» и «готовые» [1]. «Сырые» данные получены путём объединения машиночитаемых версий Викисловаря, Словарь русских синонимов и сходных по смыслу выражений, тезауруса WordNet.ru, и ряда других источников, а «готовые» данные создаются волонтёрами на основе «сырых» данных в краудсорсинговом режиме. При работе с тезаурусом невооружённым глазом видно большое количество различных элементов текстовой разметки в словах, текстах их определений, и в примерах их употреблений. Исправление каждой такой записи вручную занимает несколько десятков секунд, что снижает эффективность работы и отвлекает участников от основной задачи — создания *синсетов* (групп квазисинонимов) из «сырых» данных, снабжённых синонимическими связями, определениями и примерами употребления этих слов в том или ином контексте.

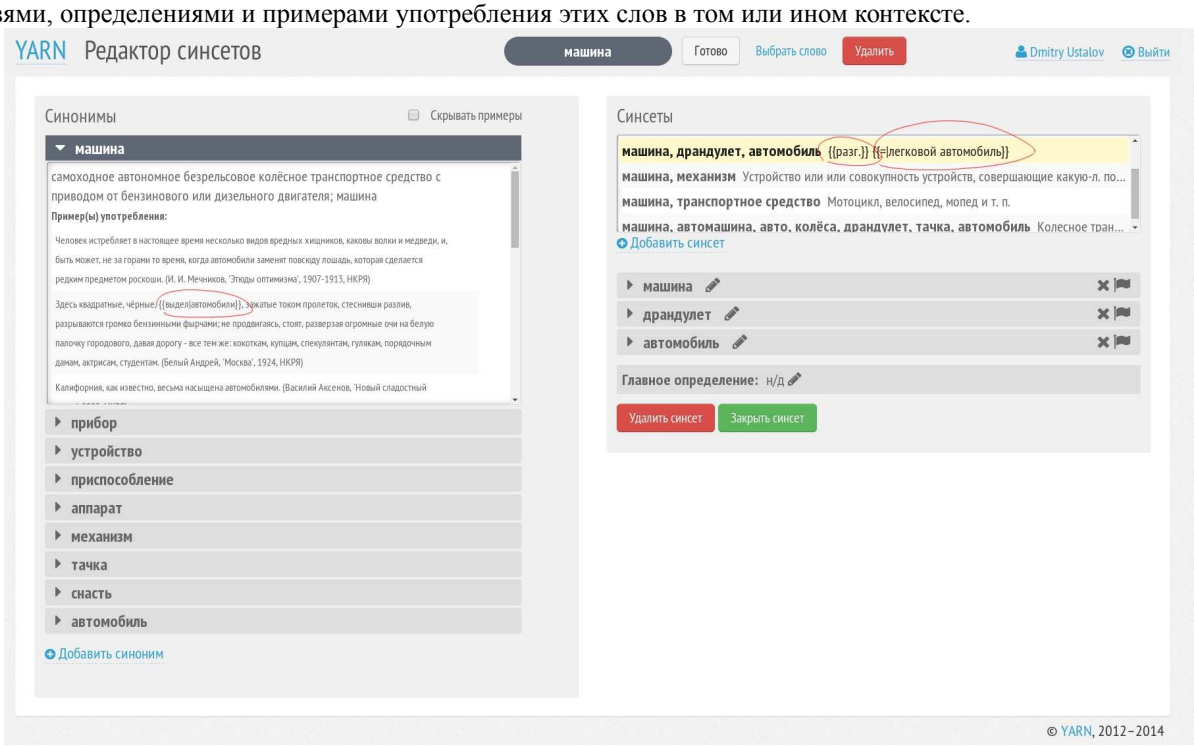

Рис. 1 Редактор синсетов, красным цветом выделены дефекты текстовых данных

На рис. 1 представлен экран интерфейса редактора синсетов, состоящий из двух частей: в левой части представлены «сырые» данные, в правой — «готовые». Выделенный фрагмент рисунка показывает достаточно типичный дефект данных, полученных при импорте из словарей: наличие Викиразметки, оставшейся после

автоматического импорта из Викисловаря. При изучении данных было выделено шесть типов дефектов, подлежащих исправлению:

1. наличие словарных помет в начале определений: "{{техн.}} двигатель транспортного средства";
2. присутствие вики-шаблонов, захватывающих часть определения: "{{=|стартовая площадка|место, где происходит запуск летательных аппаратов}}";
3. вики-разметка для выделения слов: "— Это что же тут за '''цирк''' или комедия?"
4. начало предложения со строчной буквы и отсутствие знака пунктуации в его конце: "тот, кто ищет, занимается поисками";
5. нежелательные аббревиатуры и сокращения: "дальнейшее существование, будущее кого-л., чего-л.";
6. неразрывные пробелы и другие разделители в начале и конце строки: "_частичн.: постулат".

Инструментарий. Apache Spark — это инструментальная среда распределённых вычислений, основанная на обобщении модели MapReduce, используемой в Apache Hadoop [2]. Помимо этого, Spark ориентирован на применение модели вычислений в памяти с выполнением большинства операций без дополнительного сохранения на диск. Это несколько снижает теоретическую надёжность системы, но по заявлению разработчиков системы, в десятки раз повышает производительность операций. При этом в Spark встроена возможность автоматического перезапуска задач, завершившихся сбоем. Spark полностью совместим с Hadoop версии 2.0 и выше, и может выполняться с ним на одном и том же кластере с возможностью ввода-вывода в HDFS, S3 и другие файловые системы. Apache Spark написан на языке Scala, работающем поверх Java Virtual Machine, и предоставляет «родной» программный интерфейс для языков Java, Scala и Python. При помощи предоставляемого программного интерфейса (API) можно легко и удобно строить как простые, так и сложные программы для обработки данных.

Решение. Для очистки данных разработано несколько программ, выполняющих обработку данных:

1. программа *extract.rb* на языке Ruby, извлекающая из базы данных Yet Another RussNet данные о словарных входах, определениях и примерах употребления слов и записывающая их в файлы *current_words.csv*, *current_definitions.csv*, *current_examples.csv*;
2. программа *csv_preprocessor.py* на языке Python, выполняющая предварительную обработку и подготовку CSV-файлов к распределённой обработке при помощи Spark;
3. программа *cleaner.py* на языке Python с использованием Spark, выполняющая очистку файлов при помощи регулярных выражений со встроенным рудиментарным парсером вики-разметки и исправлением других обозначенных дефектов данных с сохранением результатов в файлы *current_words.cleaned.csv*, *current_definitions.cleaned.csv*, *current_examples.cleaned.csv*;
4. программа *putback.rb* на языке Ruby, выполняющая загрузку новых CSV-файлов в базу данных Yet Another RussNet.

Помимо очистки данных, программа cleaner.py извлекает обнаруженные словарные пометы из текста определений и записывает их в дополнительную колонку файла *current_definitions.cleaned.csv*. Эти сведения пригодятся при развитии интерфейса пользователя Yet Another RussNet и будут добавлены в базу данных после дополнительного согласования.

Результаты. В результате выполнения очистки данных получены следующие изменения в словарных входах, определениях и примерах: изменён текст 3061 словарного входа (всего: 68 153), изменён текст всех загруженных определений (всего: 78 129) и всех примеров (всего: 10 785). Поскольку вычислительная сложность задачи относительно невелика, Spark использовался как инструмент для упрощения построения конвейера вычислений. Время выполнения задачи составило около одной минуты на каждый набор данных (Linux, x86_64, 8 ядер, 16 ГБ ОЗУ). Для иллюстрирования результата можно привести несколько наиболее заметных изменений в определениях и примерах употребления слов:

- "{{разг.}}, {{пренебр.}} {{помета|, также }}{{собир.}} нечто плохое, негодное; плохие вещи" → «*Нечто плохое, негодное; плохие вещи.*»
- "\'\'преимущ. мн., ед. в том же знач., что мн.\'\' состояние, в котором человек способен сознавать окружающее, владеет своими душевными и умственными способностями" → «*Состояние, в котором человек способен сознавать окружающее, владеет своими душевными и умственными способностями.*»
- "Так, например, несмотря на то, что в течение последних столетий человеческие мозги разбухли в ущерб всем остальным функциям организма, люди догадались выделить из государства только один {{выдел|орган}} — цензуру, для охраны порядка своего мира, выражающегося в государственных формах." → «*Так, например, несмотря на то, что в течение последних столетий человеческие мозги разбухли в ущерб всем остальным функциям организма, люди догадались выделить из государства только один орган — цензуру, для охраны порядка своего мира, выражающегося в государственных формах.*»

Заключение. Из результатов работы видно, что тексты, полученные из Интернета и других неконтролируемых источников хорошо подвергаются первичной автоматической обработке. Полученные очищенные данные загружены в базу данных Yet Another RussNet и доступны как для просмотра, так и для загрузки на официальном сайте проекта http://russianword.net/. Исходный код всех разработанных программ доступен в репозитории на GitHub: https://github.com/pahaz/yarn-spark-cleansing.

Благодарности. Работа выполнена при поддержке проекта РГНФ №13-04-12020 «Новый открытый электронный тезаурус русского языка».

ЛИТЕРАТУРА:

1. P. Braslavski, D. Ustalov, M. Mukhin. A Spinning Wheel for YARN: User Interface for a Crowdsourced Thesaurus // Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics. — Gothenburg, Sweden : Association for Computational Linguistics, 2014. — P. 101–104.
2. M. Zaharia, M. Chowdhury, T. Das, et al. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. Technical Report UCB/EECS-2011-82, EECS Department, University of California, Berkeley, Jul 2011.