электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

 # 06, июнь 2016

УДК 004.55

Извлечение информации из текста толковых словарей с использованием Apache UIMA RUTA

***Гутников И.Е.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»

Научный руководитель: Иванов А.М., старший преподаватель
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»
amivanoff@gmail.com

Одна из глобальная проблем XXI века – это так называемый «информационный взрыв» или рост диспропорции между объёмом информации, произведённой человечеством, и объёмом информации, которую люди способны потребить и усвоить.

Информационный взрыв порождает избыток неиспользуемой информации и хаос неструктурированной информации. Поэтому один из главных путей выхода из сложившейся ситуации – превращение неструктурированной информации в структурированную. Превращение неструктурированной информации в структурированную – главная цель парсинга текстов.

Структурированные словари на естественном языке являются основой для множества технологий обработки естественного языка, таких как машинный перевод текстов, голосовое управление и прочие.

В настоящее время в открытом доступе отсутствуют инструменты, которые позволяют из текста словарных статей различных словарей извлекать информацию для последующей машинной обработки. Для решения этой проблемы используется Apache UIMA RUTA.

Пакет Apache UIMA RUTA

Apache UIMA RUTA состоит из двух частей – скриптового языка и расширения для Eclipse. Как язык, UIMA RUTA – это правило-ориентированный скриптовый язык, который интерпретируется базовым Analysis Engine [1]. Analysis Engine (AE,

Аналитический Двигатель) - блок анализирующий документ, выводящий и записывающий атрибуты, которые описывают документ [2].

На рисунке 1 приведена структура вызова скрипта в UIMA RUTA. На этом рисунке показан процесс последовательной загрузки различных частей пакета UIMA RUTA: AE загружает основной скрипт, который загружает один или несколько дочерних скриптов, а также создает временный словарь Ruta, систему типов и свой собственный AE.

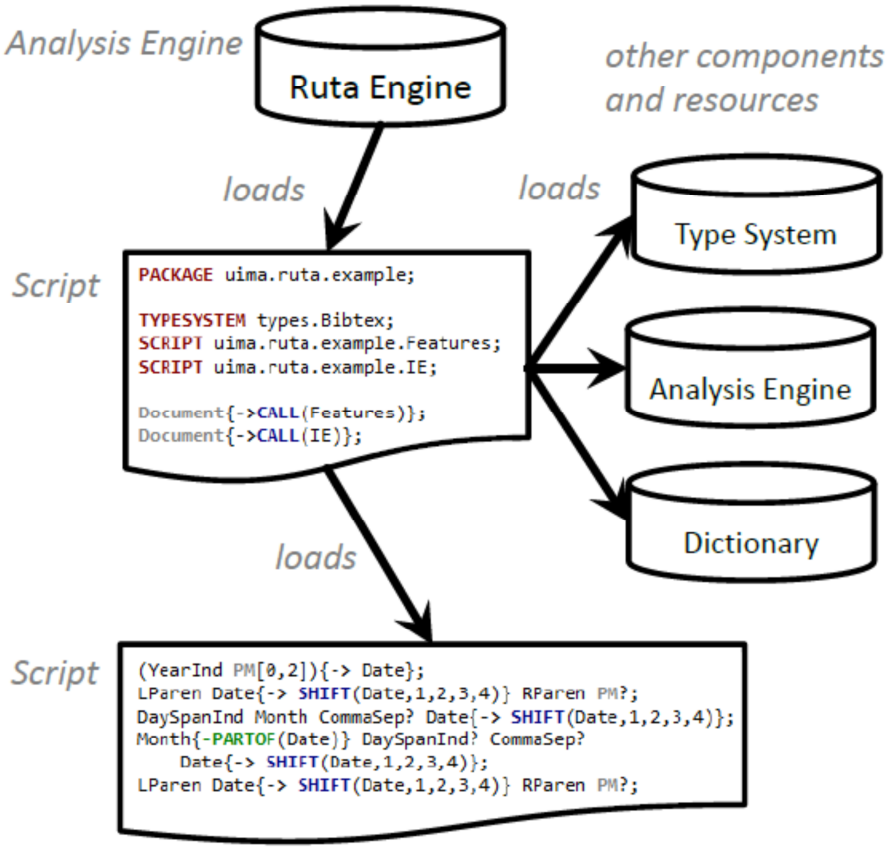

Рис. 1. Структура вызова скрипта

Синтаксис скриптов UIMA Ruta представлен на рисунке 2 и состоит из следующих секций:

- Package – пакет, в котором находится текущий скрипт;
- Import – указывает какие скрипты нужно включить в текущий скрипт;
- Declaration – объявление переменных;
- Rule – какие правила нужно выполнить над входным документом;
- Block – определяет какие правила нужно выполнить над определенным блоком внутри документа.

```
Script     → Package? Import* Statement*
Import     → (“TYPESYSTEM” | “SCRIPT” | “ENGINE” | “UIMAFIT”)
             Identifier “;”
Statement  → (Declaration | Rule | Block)
Block      → “BLOCK” “(” Identifier “)” RuleElement “{” Statement+ “}”
```

Рис. 2. Синтаксис скриптов UIMA Ruta

На рисунке 3 показаны основные принципы составления правил и синтаксис правил. Правила синтаксиса состоят из следующих частей:

— Condition – одно из условий;

— Action – действие, которое необходимо совершить над текстом, который отвечает заданным условиям;

— Quantifier – квантор; ограничивает область истинности условий;

— Wildcard – джокер; # позволяет отметить последний подходящий под условие текст;

— TypeExpression – тип текста, который нужно найти.

```
Rule           → (RuleElement+ | RegExpRule | ConjunctRules) “;”
RuleElement    → MatchReference Quantifier? ( “{” Conditions?
                 “->” Actions? “}” )? InlinedRules?
MatchReference → (TypeExpression | StringExpression | ComposedRE | WildCard)
ComposedRE     → “(” RuleElement ((“&” | “|”)? RuleElement)* “)”
```

Рис. 3. Синтаксис правил UIMA RUTA

Помимо самого скриптового языка в пакет UIMA Ruta входит специализированная среда разработки, включающая в себя ряд инструментов, позволяющих поддерживать отладить, протестировать и запустить скрипты:

— Ruta Explain Perspective – инструмент который позволяет по шагам пройти каждое из правил, которое описано в исполняемом файле. При помощи данного инструмента можно узнать: какое правило проаннотировало данный участок текста, где правило применено успешно, а где нет и почему;

— Ruta Query View – данный инструмент позволяет отлаживать правила, запуская любое их количество как запросы к указанному тексту;

— Annotation Testing View – инструмент создания unit-тестов для ruta-скриптов и сравнивать полученные результаты с документами-эталонами (gold documents). Документ-эталон, это документ, который однозначно положительно пройдет unit-тест;

— Check Annotations View – поддерживает создание документов эталонов, управляет настройкой CAS и помнит обработанные типы;

— Annotation Browser View – позволяет просмотреть аннотации, которые получились на основе входного документа. Рабочий интерфейс инструмента с извлеченной информацией из тестового русскоязычного словаря показан на рисунке 4.

—

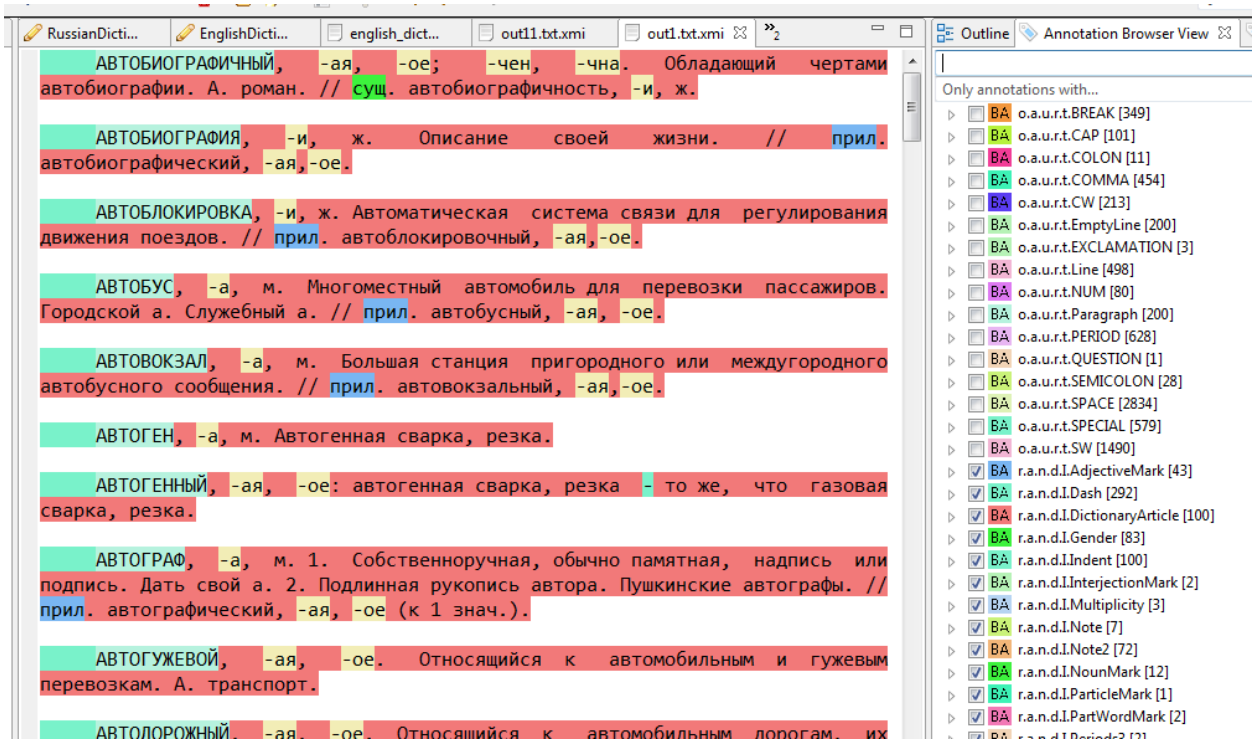

Рис. 4. Инструмент Annotation Browser View и результаты обработки тестового словаря

Также на рисунке 4 можно увидеть символьную структуру словаря, для которого были выявлены следующие принципы построения скриптов:

— Каждая словарная статья начинается с пяти пробелов и заканчивается символом переноса строки [3];

— Каждое словарное слово в словарной статье написано заглавными буквами;

— Для обозначения части речи словарного слова используется специальные сокращения;

— Для обозначения многозначных слов используются арабские цифры, которые стоят сразу после словарного слова;

— Обозначения разных значений словарного слова нумеруются в словарной статье.

Тест производительности

Тестирование системы было произведено при следующих исходных параметрах:

— Ноутбук с операционной системой OS X Yosemite, процессором Core i5 и 8 гигабайтами RAM, на Java VM выделено 4096 МБ RAM;

— Словарь Ожегова;

— Словарь Webster's Revised Unabridged Dictionary.

Результаты тестирования представлены в графиках на рисунках 5 и 6.

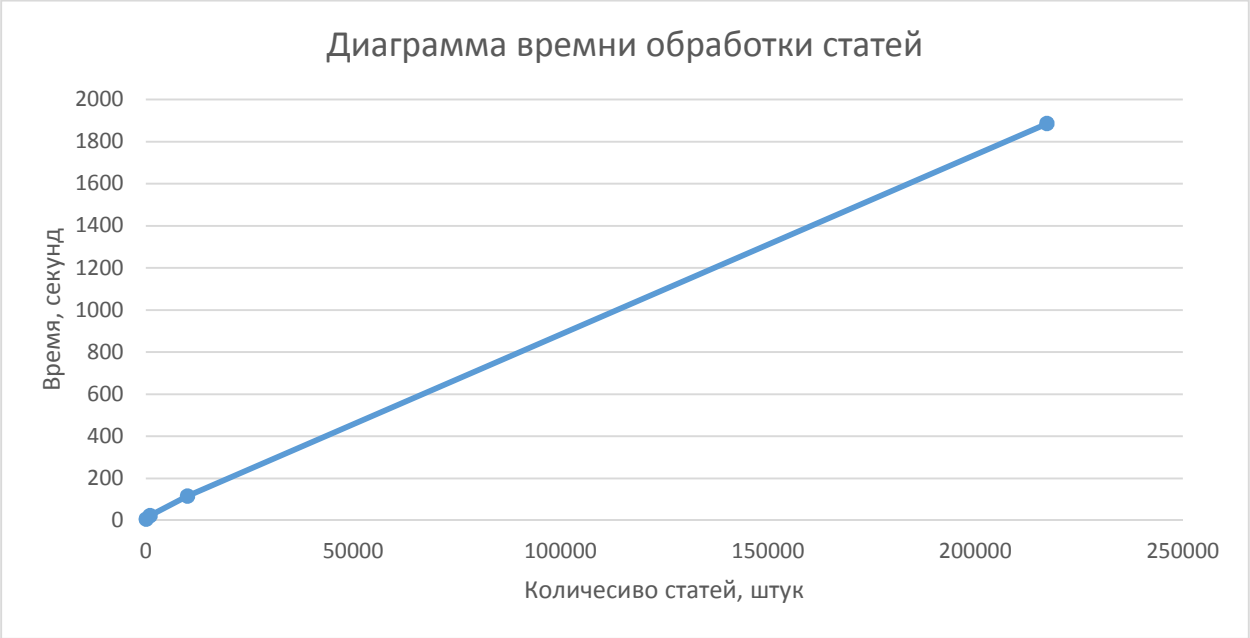

Рис. 5. Диаграмма времени обработки словарных статей

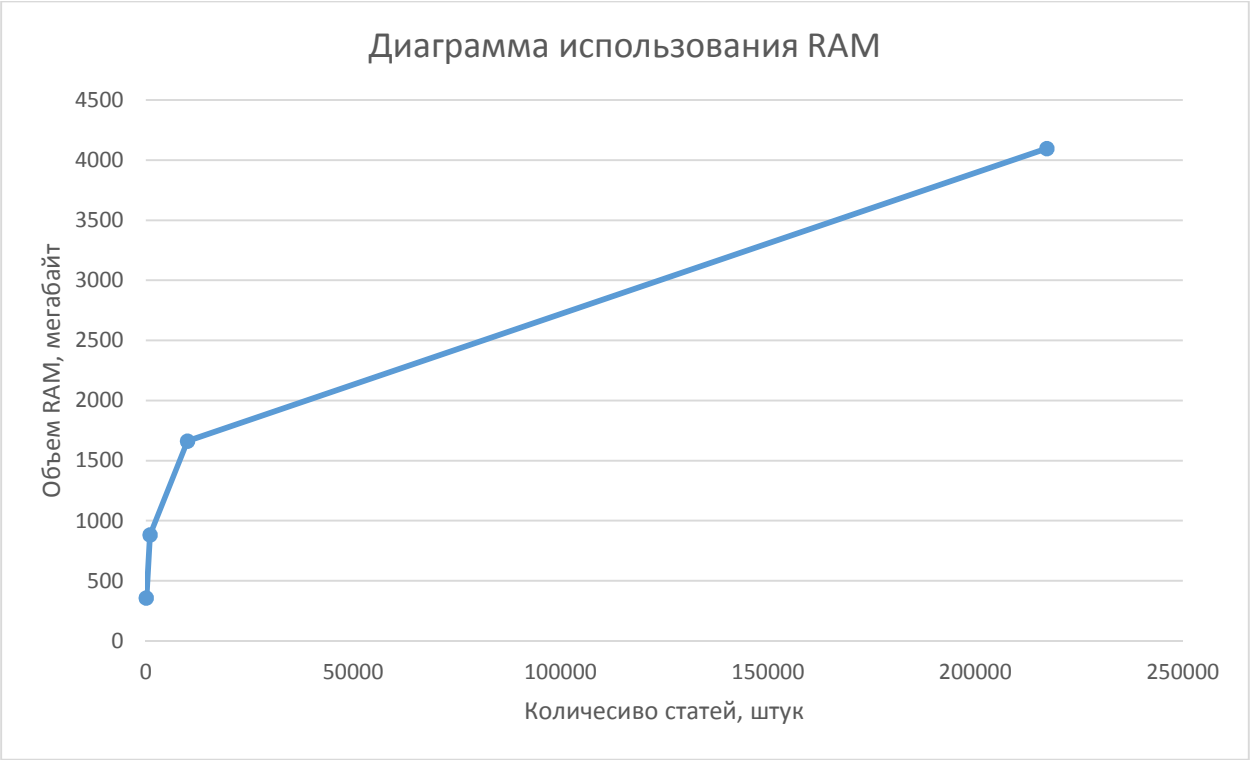

Рис. 6. Диаграмма использования RAM

Выводы

При анализе пакета Apache UIMA RUTA были получены данные о внутреннем устройстве пакета, о принципах построения правил и скриптов, а также инструментах пакета.

В конечном итоге был создан инструмент для извлечения информации из неструктурированных русскоязычных словарей для последующей их машинной обработки и был проведен нагрузочный тест производительности полученного инструмента.

В результате нагрузочного теста производительности было выявлено, что созданный инструмент является надежным решением задачи по извлечению данных из текста толковых словарей. Графики потребления ресурсов центрального процессора и оперативной памяти показали, что с обработкой данных словаря типового объема (размер текстового файла словаря приблизительно равен 10 Мб) может справится любой современный и производительный компьютер.

В дальнейшем планируются использовать полученный инструмент для создания интерфейсов на основе контролируемых естественных языков.

Список литературы

[1]. Apache UIMA Ruta. Available at: https://uima.apache.org/downloads/gscl2013/2013-GSCL-Ruta.pdf, accessed 24.12.2015.

[2]. UIMA Overview. Available at: https://uima.apache.org/d/uimaj-current/overview_and_setup.html, accessed 24.12.2015.

[3]. Гречищев К. М. Применение Apache UIMA при решении задачи выделения имён из текстов документов // Молодежный научно-технический вестник. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 2. Режим доступа: http://sntbul.bmstu.ru/doc/541710.html (дата обращения 24.12.2015).