УДК 004.82

ИНСТРУМЕНТ РАБОТЫ С ДАМПАМИ ВИКИПЕДИИ ДЛЯ ЗАДАЧ КОМПЬЮТЕРНОЙ ЛИНГВИСТИКИ

В.А. Крайванова
Алтайский государственный технический университет им. И.И. Ползунова
г. Барнаул

В статье проведен анализ существующих возможностей работы с Википедией как с корпусом текстов, и предложен автоматизированный инструмент получения корпусов текстов на основе Википедии для различных задач компьютерной лингвистики.

Ключевые слова: инструмент создания лингвистического корпуса, Википедия, автоматический анализ текстов.

«Википедия» – общедоступная многоязычная универсальная интернет-энциклопедия со свободным контентом, реализованная на принципах вики[1]. Не первый год в научных кругах ведется полемика о том, можно ли считать Википедию заслуживающим доверия источником информации. Сегодня, в связи с активным развитием машинного обучения, Википедия становится для ученых не просто возможностью найти ссылки на литературу, но и уникальным набором частично размеченных данных. Особенно это важно для сферы компьютерной лингвистики, так как создание корпусов – невероятно трудоемкая задача, в то время как из Википедии такие корпуса можно извлекать автоматически или полуавтоматически. Да, безусловно, разметка таких корпусов будет существенно отставать по качеству от сделанной вручную, и зачастую она не соответствует в точности потребностям ученых. Однако данный недостаток частично компенсируется размером корпуса. В связи с этим является актуальной задача эффективного программного доступа к Википедии, в первую очередь, для такого популярного среди исследователей языка, как Python.

Чтобы понимать все преимущества, которые дает работа с Википедией, рассмотрим основные словари и корпусы, доступные для русского языка. Такие ресурсы должны по возможности удовлетворять следующим условиям.

− Ресурс должен быть открытым, бесплатным, свободно распространяемым.

− Ресурс должен быть актуальным и пополняемым. Такой ресурс более точно соответствует современной лексике и со временем будет только улучшаться, что будет приводить к улучшению модели на его основе.

− Ресурс по возможности должен содержать необходимую разметку. Для словарей необходима морфологическая разметка. Для корпусов текстов – различная синтаксическая и семантическая разметка.

Большая коллекция ресурсов для задач анализа естественных языков (включая звучащую речь) представлена в каталоге NLPub[3]. Рассмотрим некоторые из них.

− Opencorpora [4] – большой открытый свободный корпус с разметкой морфологии и именованных сущностей, пополняемый сообществом.

− Словарь АОТ [5] – это русский морфологический словарь, базируется на грамматическом словаре А. А. Зализняка.

− Ресурсы-аналоги технологии WordNet [6] – электронные тезаурусы в виде семантических сетей. Англоязычный WordNet разработан в Принстонском университете и выпущен под некопилефтной свободной лицензией [3]. Базовой словарной единицей в WordNet является не отдельное слово, а синонимический ряд. Было сделано несколько попыток [7-10] реализовать WordNet для русского языка. Один из них представляет собой частично переведенный англоязычный WordNet [7]. Yet Another RussNet (YARN) - проект создания нового открытого электронного тезауруса русского языка, разрабатывается усилиями представителей УрФУ, ВШЭ, ИММ УрО РАН и Kontur Labs. [8-9]. Третий проект RussNet с 1999 года разрабатывается группой под руководством И. В. Азаровой (СПбГУ) [10].

− Генеральный Интернет-корпус Рус-

ского Языка (ГИКРЯ) – огромный корпус, созданный при помощи полностью автоматической технологии сбора и морфологической разметки текстов российского сегмента интернета. Кроме морфологии содержит метаданные об источнике и авторе текста. Большая часть корпуса является закрытой [11-12].

− Национальный корпус русского языка (НКРЯ) – коллекция корпусов, доступная (за некоторым исключением) только через веб-интерфейс, и не разрешенная к коммерческому использованию [13].

Таким образом, большая часть современных лингвистических ресурсов для русского языка содержит, в первую очередь, морфологическую и синтаксическую разметку. Для других языков, особенно английского, варианты разметки более разнообразны. Большая коллекция размеченных корпусов для различных языков доступна на сайте Linguistic Data Consortium [14]. Эта организация с 1992 года занимается сбором таких корпусов. К сожалению, для русского языка имеются только параллельные корпусы.

Рассмотрим теперь Википедию с точки зрения формирования корпусов текстов. Википедия активно используется в качестве источника данных во многих проектах [15,16].

Документы в Интернете состоят из собственно текста и метаданных. Текст включает в себя линейно упорядоченные заголовки, списки, изображения, таблицы и другие элементы. Метаданные представляют собой элементы разметки html или semantic web. Для большинства документов метаданные невелики. Как правило, это основной заголовок, источник текста и, возможно, дата создания и автор (если документ является новостной статьей или постом в блоге).

В отличие от других возможных источников Википедия имеет следующие достоинства.

− Большой объем. Русский сегмент энциклопедии содержит более 1 429 тысяч статей[17].

− Свободная доступность и бесплатность (лицензия Creative Commons Attribution-ShareAlike 3.0 Unported) [17].

− Актуальность и регулярная пополняемость. Википедия пополняется и поддерживается в актуальном состоянии силами огромного сообщества. Кроме того, тексты в Википедии становятся только лучше.

− Частичная разметка. Тексты из Википедии снабжены достаточно подробными и стандартизованными метаданными. Также имеется подробная иерархия категорий.

− Для Википедии доступна статистика посещений статей.

В Викисловаре есть морфологическая разметка для некоторого количества слов, но в целом Википедия подходит для обучения синтаксических анализаторов только на основе алгоритмов без учителя. В сравнении с семантически размеченными корпусами тестов, Википедия содержит очень упрощенную разметку, лишь частично подходящую под нужды той или иной решаемой задачи, но эта разметка очень разнообразна, включает не только информацию о целых текстах, но и об их внутренней структуре, и в качестве первого приближения может быть адаптирована под большой круг задач.

Не смотря на эти преимущества, Википедия остается недоступной для рядовых исследователей по причине отсутствия качественных программных средств работы с ее дампами. Это связано со следующими проблемами.

Большинство формальных языков, с которыми сталкивается программист, имеют очень строгую структуру. Если они и позволяют вставлять в тексты произвольные конструкции, то они должны быть соответствующим образом обособлены (например, кавычками в языках программирования или специальной конструкцией CDATA в XML) и экранированы. При этом общая структура текстов остается строго формализованной. Некоторые языки разметки устроены по другому принципу. Большая часть текста – это просто набор символов, а формальные конструкции достаточно произвольно разбросаны по тексту. Примерами таких языков являются HTML и Markdown. Язык, на котором размечены страницы современной Википедии – это компиляция этих двух языков, которая постоянно дополняется новыми конструкциями.

Грамматика вики-разметки, в том или ином движком, зависит от версии движка и установленных плагинов. Кроме того, некоторые формальные конструкции, например, перенаправления, зависят от применяемой локали. Более-менее полный список формальных конструкций современной вики-разметки можно найти в статье [18]. Нас интересуют в первую очередь заголовки, ссылки и шаблоны, которые позволяют определить внутреннюю структуру и метаинформацию документа, в частности, принадлежность к категориям. Элементы страницы могут быть вложены друг в друга и содержать произвольный текст (в том числе с элементами, которые совпадают по синтаксису с формальными конст-

рукциями, но ими не являются).

Википедия доступна как через web API [19], так и в виде дампов [20]. Получение данных по сети - задача, очень затратная по времени, кроме того, Википедия настоятельно не рекомендует скачивать ее непосредственно с основных серверов, так как это серьезно повышает нагрузку на серверы, поэтому разработанный нами инструмент базируется на дампах. Выгрузка производится примерно раз в месяц, в два формата: sql и xml.

Для использования Википедии в качестве корпуса размеченных текстов необходимо иметь возможность эффективно извлекать формализованные данные из вики-разметки. Перечислим требования к парсеру Википедии.

− Парсер должен работать с xml-дампом статей Википедии.

− Парсер должен извлекать заголовки, ссылки, шаблоны и перенаправления.

− Парсер должен не просто извлекать элементы статей, а давать доступ к их местоположению в тексте, в идеале - строить полное дерево разбора.

− Парсер должно быть достаточно просто модифицировать под изменения в вики-разметки.

Парсер, встроенный в ядро вики-движка, разбирает и преобразует в HTML лишь самые базовые конструкции: теги nowiki, шаблоны, ссылки, заголовки и некоторые другие элементы – а также производит очистку от html[21]. Остальные элементы страницы разбираются специальными плагинами. Полное дерево разбора не строится, осуществляется только перевод конструкций. Оригинальный парсер медиавики имеет сложную многопроходную архитектуру, и для задачи построения дерева объектов неприменим.

Существует множество парсеров, альтернативных тому, что встроен в движок Википедии. Длинный список проектов для различных языков вы можете найти в статье [22]. Многие из них заброшены, не дойдя до стадии первого релиза. Большинство этих проектов трансформируют вики-разметку в HTML-разметку или другие форматы. Некоторые из них требуют, чтобы сама Википедия или другой проект на движке mediawiki, информацию с которого собираются использовать, был доступен онлайн. Например, библиотека mwlib [23] от компании PediaPress реализует конвертацию вики-статей в различные форматы, в первую очередь, в pdf. Библиотека MediaWiki Utilities предоставляет первичный доступ к Mediawiki-движку через web-API, коннект к базе MySQL или итерирование по xml-дампу [24]. Библиотека WikiExtractor [25] позволяет извлекать из дампа Википедии различные объекты: списки изображений, шаблоны, заголовки и другие элементы. Однако, эта библиотека не учитывает особенности разметки русской Википедии, например, использование русских ключевых слов наравне с английскими («перенаправление» и «redirect»). Библиотека Wikipedia: Python Library [26] предоставляет возможность доступа к web-API Википедии, но не работает с дампами.

Таким образом, требуется разработка парсера вики-разметки и фреймворка для доступа к элементам статей из программ анализа текстов. Дампы Википедии представляют собой xml и sql файлы.

Для того, чтобы сделать их пригодными в нашей задаче, необходимо обеспечить высокую скорость чтения по ряду запросов:

− по идентификатору статьи получить ее заголовок и текст;

− по заголовку статьи получить ее идентификатор;

− эффективно работать с деревом категорий;

− определить, является ли страница перенаправлением и другие.

Чтобы решить эту задачу, необходимо либо загрузить дампы в реляционную базу данных или любую другую систему индексации, либо реализовать систему файловых индексов. Использование готового хранилища, в частности, реляционного, имеет сравнительно невысокую скорость на чтение и создает дополнительную зависимость от сторонних библиотек, а также запуск сервера. Кроме того, Википедия содержит большое количество информации, и готовое хранилище потребует большое количество ресурсов, которых может не быть на компьютерах исследователей. Так как индексы должны быть доступны только на чтение, и нет потребности в дополнительной поддержке консистентности, было принято решение реализовать базовые индексы в виде файлов. На рисунке 1 представлена концепция архитектуры разработанной библиотеки.

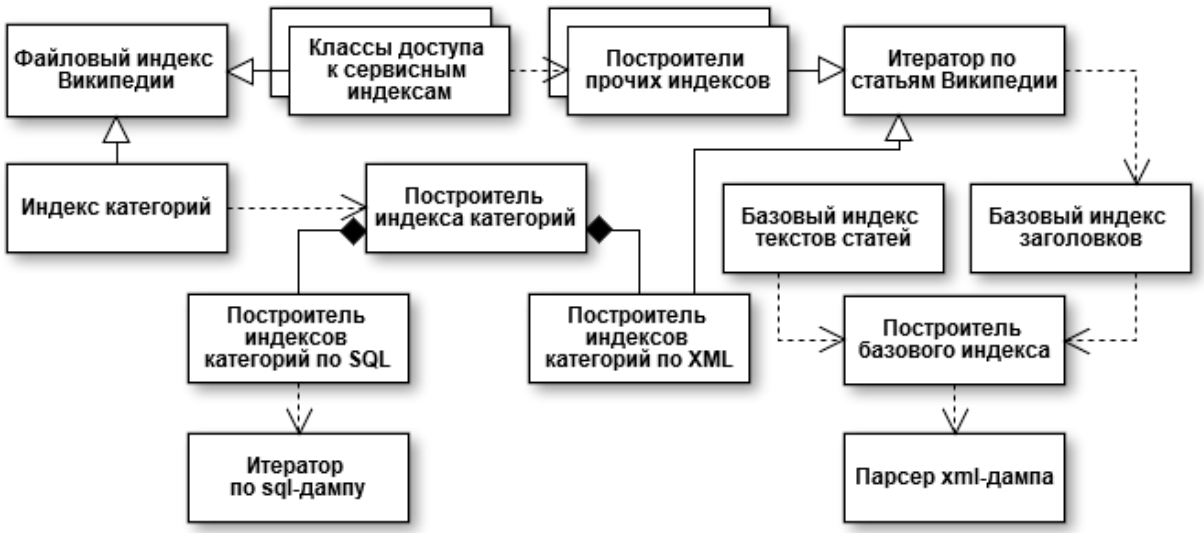

Рисунок 1 – Обобщенная диаграмма классов разработанной библиотеки

На основе xml-дампа строится базовый индекс доступа к заголовкам и текстам статей.

После этого пользователем могут быть реализованы сервисные индексы. Некоторые из них уже реализованы в библиотеке, и речь о них пойдет далее.

Сервисные индексы являются наследником от базового класса файловых индексов, который обеспечивает их загрузку, проверку консистентности и, в случае необходимости, запускает их построение. Построители индексов реализуются на базе итератора по идентификаторам статей Википедии. Индекс категорий кроме стандартного итератора задействует также итератор по SQL-дампам.

Исходными данными для системы базовых индексов являются два файла дампа Википедии[20]:

- articles.xml;
- catlinks.sql.

Файл articles.xml содержит титулы и тексты всех Вики-статей со всей Вики-разметкой, в том числе категориями. Однако, не все категории указываются непосредственно в Вики-разметке. Часть из них может добавляться автоматически при использовании специальных шаблонов, например, категория «Персоналии» добавляется при использовании огромного количества шаблонов. Чтобы получить более полный граф категорий, используется файл catlinks.sql. Этот файл также содержит не все категории, так как, видимо, sql-связи строятся не при сохранении статьи Википедии после редактирования, а на основе какого-то другого алгоритма.

На первом этапе построения индексов Википедии файл articles.xml преобразуется в индекс текстов и титулов статей, что позволяет получать быстрый доступ к статьям. Далее на основе итератора по базовому индексу строятся другие индексы. Все индексы сохраняются в файлах.

На рисунке 2 представлена схема взаимосвязи между индексами.

Индекс доступа к «сырым» текстам статей реализует быстрый доступ к дампу Википедии без необходимости парсинга текстовых форматов. Состоит из text.pkl - сериализованного python-словаря, хранящего пары «индекс статьи - позиция в файле текстов» и text.dat – файл текстов статей в виде потока байтов, в формате «длина статьи, байты статьи». Данные индексы строятся при непосредственном парсинге articles.xml.

Параллельно с ними строится индекс title_RawTitleIndex.pkl, содержащий сериализованный python-словарь, хранящий пары «индекс статьи - титул статьи».

Индекс поиска по титулам статей реализует возможность быстро определять по идентификатору статьи ее титул и наоборот. Строится на основе базового индекса титулов. Титулы хранятся в нормализованном виде (удалены знаки подчеркивания, которыми иногда заменяются пробелы в заголовках, все буквы переведены в нижний регистр). Помимо уже описанного файла индекс содержит два словаря: title_IdToTitle.pkl, хранящий пары «идентификатор статьи – титул статьи» и title_TitleToId.pkl, хранящий пары «титул статьи – идентификатор статьи».

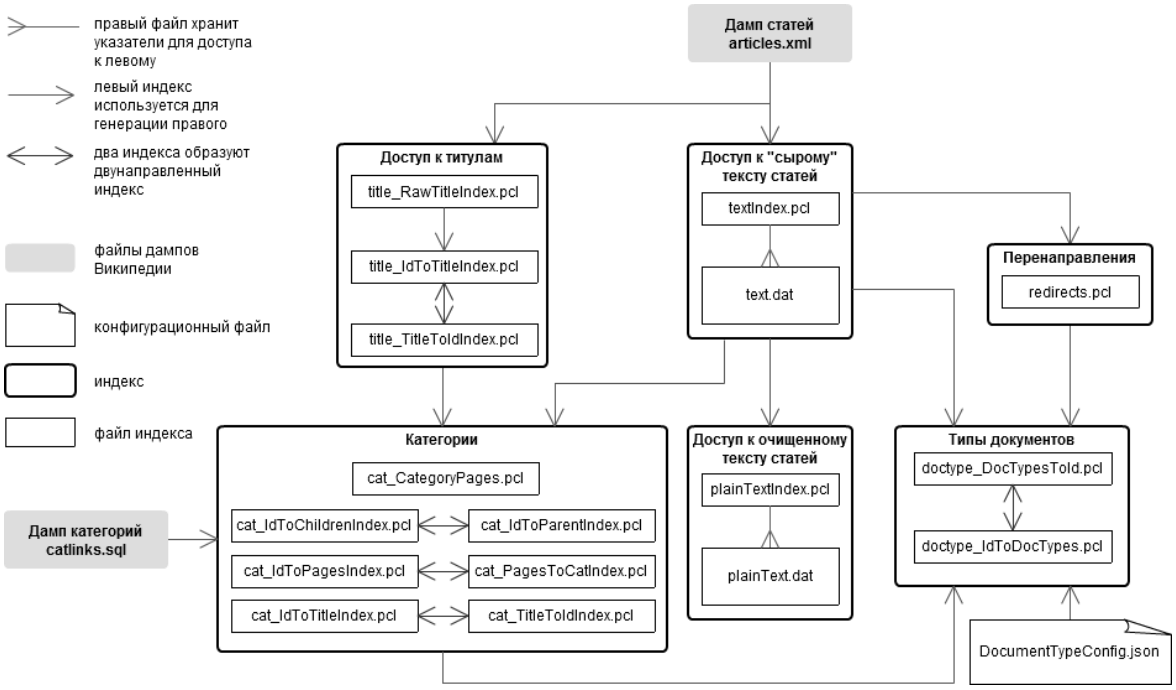

Рисунок 2− Схема базовых индексов для доступа к дампам Википедии

Индекс очищенных текстов статей содержит тексты статей, очищенные от викиразметки, и состоит из следующих файлов: `plainTextIndex.pkl` – сериализованный python-словарь, хранящий пары «индекс статьи – позиция в файле текстов», и `plainText.dat` – тексты статей в виде потока байтов, в формате «длина статьи, байты статьи».

Многие статьи Википедии являются перенаправлением на другие статьи или части статей. Индекс перенаправлений реализует возможность работать с такими статьями отдельно, или исключать их из рассмотрения, и хранится в файле `redirects.pkl` в виде сериализованного python-словаря, хранящего пары «индекс статьи – перенаправление (пара из идентификатора статьи и якоря внутри нее, если он установлен)».

Индекс категорий содержит граф категорий Википедии в виде набора сериализованных python-словарей.

− cat_TitleToIdIndex хранит пары «титул категории – индекс категории»;

− cat_IdToTitleIndex хранит пары «индекс категории – титул категории»;

− cat_IdToChildrenIndex хранит пары «индекс категории – список дочерних категорий»;

− cat_IdToParentIndex хранит пары «индекс категории – список родительских категорий»;

− cat_IdToPagesIndex хранит пары «индекс категории – список статей, непосредственно приписанных к данной категории «;

− cat_PagesToCatIndex хранит пары «индекс статьи - список категорий, в которые непосредственно вложена данная статья».

− cat_CategoryPages хранит пары «индекс категории – страница данной категории» для тех категорий, у которых есть отдельные страницы.

Индекс типов документов, строго говоря, не является базовым индексом Википедии, и может строиться для каждой задачи отдельно. Этот индекс предназначен для того, чтобы составлять списки статей по определенному набору признаков. Например, нам необходим список персоналий или список географических локаций. Принадлежность к таким группам можно определить по следующим метаданным в статье:

− используемый шаблон, например, «космонавт» или «нп», то есть населенный пункт);

− приписанная категория, например, «Персоналии по алфавиту», «Города России»;

− свойству, имеющемуся в шаблоне, например, «дата рождения»;

− префиксу перед заголовком статьи (для технических статей, например, «Проект:» или «Википедия:»).

Википедия – развивающийся проект, по-

этому далеко не у всех статей прописаны все метаданные, например, категории. Может иметь место и обратная ситуация, когда список категорий достаточно полон, а шаблон не применяется. Поэтому для достижения хороших результатов необходимо прописывать разные типы признаков.

Индекс типов документов состоит из двух сериализованных python-словарей: `doctype_DocTypesToId.pcl`, хранящий пары «название типа документа – список включенных в него страниц», и `doctype_IdToDocTypes.pcl`, хранящий пары «индекс статьи – список приписанных к ней типов документов».

Разработанный расширяемый инструмент работы с дампами Википедии поможет многим ученым начать использовать сетевую энциклопедию как набор данных в своих исследованиях. Исходный код библиотеки доступен на GiHub[27].

СПИСОК ЛИТЕРАТУРЫ

1. Википедия // Википедия. Url: https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F.
2. «Википедию» признали эффективным инструментом для продвижения науки // Интернет-издание N+1. 22 сентября 2017. Url: https://nplus1.ru/news/2017/09/22/wiki-science
3. Ресурсы // NLPub. Url: https://nlpub.ru/%D0%A0%D0%B5%D1%81%D1%83%D1%80%D1%81%D1%8B#.D0.9A.D0.BE.D1.80.D0.BF.D1.83.D1.81_.D1.82.D0.B5.D0.BA.D1.81.D1.82.D0.BE.D0.B2.
4. Opencorpora. Url: http://opencorpora.org.
5. Автоматическая обработка текста. Url:http://aot.ru.
6. WordNet: a lexical database for English. Url: http://wordnet.princeton.edu/
7. Русский Wordnet. Url: http://wordnet.ru/
8. Yet Another RussNet. Url: https://russianword.net/
9. Braslavski, P. et al. YARN: Spinning-in-Progress // Proceedings of the Eight Global Wordnet Conference, 2016, Bucharest, Romania, pp.58-65. Url: http://gwc2016.racai.ro/ procedings.pdf\#page= 67
10. RussNet: тезаурус русского языка. Url: http://project.phil.spbu.ru/RussNet/index_ru.shtml
11. Генеральный Интернет-Корпус Русского Языка. Url: http://www.webcorpora.ru"
12. Belikov V., Kopylov N., Piperski A., Selegey V., Sharoff S. Big and diverse is beautiful: A large corpus of Russian to study linguistic variation. // Web as Corpus Workshop (WAC-8). 2013. Pp.24-29.
13. Национальный Корпус Русского Языка. Url:http://ruscorpora.ru/index.html
14. Language Data Consorcium Catalog by Year // Linguistic Data Consorcium Site. Url: https://catalog.ldc.upenn.edu/byyear
15. Зинько, Д. Contropedia – сайт для анализа и визуализации споров внутри статей Википедии // Теплица социальных технологий, 2015.
16. Panchenko, A. et al. Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia // Proceedings of Concept Discovery in Unstructured Data Workshop of International Conference On Formal Concept Analysis, 2012, Belgium. Pp. 78-88.
17. Википедия:Описание // Википедия. Url:http://ru.wikipedia.org.
18. Help:Wiki markup // Wikipedia Url: https://en.wikipedia.org/wiki/Help:Wiki_markup.
19. API:Main page // MediaWiki. Url: https://www.mediawiki.org/wiki/API:Main_page
20. Wikimedia Downloads. Url: https://dumps.wikimedia.org/
21. Mediawiki/includes/parser/Parser.php // Phabricator. Url: https://phabricator.wikimedia.org/diffusion/MW/browse/master/includes/parser/Parser.php
22. Alternative parsers // MediaWiki. Url: https://www.mediawiki.org/wiki/Alternative_parsers
23. Collection Extension for MediaWiki. Url:http://mwlib.readthedocs.io/en/latest/index.html
24. MediaWiki Utilities Documentation. Url:http://pythonhosted.org/mediawiki-utilities/
25. Wiki Extractor. Url: https://github.com/attardi/wikiextractor
26. Wikipedia: Python library. Url: https://github.com/goldsmith/Wikipedia
27. Исходный код разработанной библиотеки PyWikiText. Url: https://github.com/smart-frogg/pywikitext

Крайванова Варвара Андреевна – к.ф.-м.н., доцент, телефон: +7-913-230-34-19, email:krayvanova@yandex.ru.