Раздел II. Математическое и программное обеспечение суперкомпьютеров

УДК 004.931 DOI 10.18522/2311-3103-2016-12-2942

А.А. Белозеров, Д.В. Вахлаков, С.Ю. Мельников, В.А. Пересыпкин, Е.С. Сидоров

ТЕХНОЛОГИЧЕСКИЕ АСПЕКТЫ ПОСТРОЕНИЯ СИСТЕМЫ СБОРА И ПРЕДОБРАБОТКИ КОРПУСОВ НОВОСТНЫХ ТЕКСТОВ ДЛЯ СОЗДАНИЯ МОДЕЛЕЙ ЯЗЫКА

Предложена и реализована программная система сбора и предобработки корпуса новостных текстов из Интернет-источников, предназначенного для создания статистических моделей языка. Система подразумевает первоначальное участие лингвистов и обеспечивает большую скорость набора и высокую чистоту собираемого корпуса. В качестве источников используются материалы RSSлент и карт сайтов. Описана методика поиска и выбора информационных источников на примере создания корпуса арабского языка. Система сбора включает программные модули сбора ссылок, скачивания статей, хранения, выделения текстов из html*-документов и управления. В системе реализовано два интерфейса управления: администратора и лингвиста. Используемый метод выделения текстов основан на оригинальной статистике количества текста в* html*-документе. Подсистема предобработки предназначена для очистки собранных корпусов и включает в себя программные модули поиска нечетких дубликатов и текстовых вкраплений не на целевом языке. Система разработана на языке Python, с использованием ряда фреймворков и компонентов с открытым исходным кодом, и работает под управлением ОС Ubuntu на двух серверах с использованием 16 процессорных вычислительных ядер. К августу 2016 года в обработке находилось более 20000 новостных источников на 14 языках из 70 стран мира. Весь перечень источников обходится каждые 2 часа. Для различных языков собраны корпуса от 500 Мб до 20 Гб. Созданная технология позволяет получать корпуса текстов, структурированные по стране происхождения, дате написания, тематике, источнику, а также наращивать объемы уже созданных корпусов для построения более точных языковых моделей с использованием актуальных новостей. Приведены экспериментальные данные по перплексии обученных триграммных моделей на собранном англоязычном корпусе общественно-политической тематики и свободно распространяемых корпусах OANC и Europarl_v7.*

Корпус текстов; парсер; очистка корпуса; перплексия; модель языка.

A.A. Belozerov, D.V. Vakhlakov, S.Yu. Melnikov, V.A. Peresypkin, E.S. Sidorov

ENGINEERING ASPECTS OF BUILDING THE SYSTEM OF COLLECTION AND PREPROCESSING OF NEWS TEXT CORPUSES FOR CREATION OF THE LANGUAGE MODELS

The paper describes an approach to building a software system for collecting and preprocessing text corpuses for natural language modelling. The system assumes that the list of sources is prepared by language experts that allows to increase collection speed and raise quality of the resulting corpus. The corpuses are collected from different Internet sources (mainly news web portals) by parsing and crawling RSS feeds, sitemap files and data from social networks. An example of how to collect such sources for Arabian language is given in the paper. The software system consists of several logical modules: links collection module, crawler, HTML parsing and

text extraction module and web interface for two types of users - language expert and administrator. The original text extraction approaches based on "text quantity metric" as well as additional preprocessing step are also discussed. The preprocessing step applies fuzzy duplicates search algorithms and a filtering algorithm to remove repeated pieces of text and filter out articles that do not belong to the target language. The software system is implemented in Python with the use of several open source frameworks. The system works under Ubuntu OS on two dedicated servers of 16 CPU cores in total. In August 2016 the system was processing more than 20000 news sources on 14 languages from 70 countries. The whole list of sources is crawled during two hours. Text corpuses with sizes ranging from 500Mb till 20Gb were collected for all these languages. The described technology allows collecting text corpuses classified by country of origin, writing date, topics, type of source and also enriching the existing corpuses to build more precise natural language models. As an experiment, the collected data was used to build three-gram models for English language (political topic) and compared in terms of perplexity to the similar ones built using well-known OANC and Europarl_v7 corpuses.

Text corpora; Parsing; Corpus Quality; Perplexity; Language model.

Введение. В современных системах распознавания речи, печатного или рукописного текста, интеллектуальной обработки, синтеза и автоматического перевода текстов используются так называемые модели языка, которые позволяют оценивать правдоподобие вариантов распознанного текста, вычислять степень их согласия с основными статистическими характеристиками языка [1–3]. Модели языка ([4,5]) строятся с помощью того или иного способа статистической обработки набора текстов большого объема на данном языке (такие наборы называются корпусами). Если для наиболее распространенных мировых языков доступны представительные корпуса, распространяемые свободно или на коммерческой основе, то для редких и, так называемых, «малоресурсных» языков ([6]) таких корпусов может и не быть. В ряде приложений необходимо иметь языковые корпуса, находящиеся в актуальном состоянии, прежде всего по лексическому составу. В работе предлагается решение задачи создания корпусов текстов на основе построения системы их сбора по новостнымИнтернет-источникам.

Задача сбора корпусов по Интернет-источникам решается многими исследователями, для этого используются различные способы. В работе [7] описана автоматическая система сбора текстовых корпусов для целей распознавания речи. Программное средство позволяет от заданного «начального» URL идти по ссылкам на заданную глубину, собирая html-страницы и проводя после этого их «парсинг» (разбор) для выделения текста. К недостаткам такого подхода относится непредсказуемость и плохая контролируемость состава собираемого корпуса (по ссылке можно уйти на рекламные сайты, сайты на других языках и др.). В [8] изучаются вопросы оптимизации автоматических процедур для сбора корпусов. Наличие в обучающем корпусе свежих новостных материаловзначительно повышает точность автоматического распознавания речи в телевизионных новостях, и в [9] предложены способы дополнения существующих корпусов актуальными новостными текстами, собираемыми с помощью технологий Web 2.0 с использованием возможностей RSS и Twitter.

В настоящей работе предложен иной подход к сбору новостных текстовых корпусов, требующий участия экспертов-лингвистов, но обеспечивающий высокое качество собираемого текстового материала, описаны основные модули реализующей этот подход программной системы. Приведены экспериментальные данные о характеристиках собираемых корпусов и точности обученных языковых моделей.

1. Поиск и выбор информационных ресурсов. На первом этапе лингвисты с помощью универсальных поисковых систем и каталогов (Google, Bing, Ask, Yahoo), а также национальных систем составляют максимально полный перечень мировых информационных ресурсов на целевом языке. Значительно повышает

эффективность поиска использование запросов и механизмов расширенного поиска (advancedsearch), позволяющих, например, искать только сайты на определённом языке или только сайты в определенной географической зоне. Полезно также варьировать запросы по морфологическим формам языка (к примеру, варьирование морфологических форм для арабского языка заметно влияет на релевантность поисковой выдачи).

Создаваемые корпуса текстов, с одной стороны, должны быть достаточно универсальны (для построения адекватных моделей языков), а с другой – отражать лишь определённые пласты языка, например, современную письменную лексику социально-политической направленности. Наиболее соответствующими указанным требованиям информационными ресурсами в сети Интернет являются сайты сетевых и печатных СМИ, телеканалов, радиостанций, информационных агентств, общественно политические сайты и новостные ленты. Отдельным источником информации могут служить различные Wiki-ресурсы – сетевые энциклопедии, совместно разрабатываемые и модерируемые самими пользователями, однако у них есть существенный минус – низкая актуальность, так как многие упоминаемые издания давно прекратили существование.

В качестве примера опишем результаты применения предложенного механизма для арабского языка. Поиск в Google по тематическим запросам "сайты, касающиеся арабских газет" (на арабском языке) обнаруживает ряд медиа-каталогов – как панарабских, содержащих списки газет, журналов, радиостанций, телеканалов и сайтов всех арабских и ряда неарабских стран (http://allnewspapers.com,http://newspaperglobal.com, http://arabic-media.com, http://www.arabo.com и др.), так и национальных (иракский http://www.tenaa.net/site1093.html, сирийские http://www.syrianmedia.com/media/newspaper/ и http://syriagate.com/347/, ливийский http://www.reinventinglibya.org/ar/media.php). Поиск Yandex в русскоязычном сегменте интернета находит каталоги «Ближний восток» http://middleeast.org.ua/catalog/pressa.htm, Arabic.ru http://arabic.ru/read/arab/press и сайты информационных агентств: ИноСМИ http://inosmi.ru/asia/, РИА-НОВОСТИ http://ria.ru/asia/ и ИТАР-ТАСС http://itar-tass.com.

Существуют также арабоязычныеагрегаторы новостей. Например, суданские "Аль-Машахир" http://www.almshaheer.com и "Сударесс" http://www.sudaress.com, йеменский "Ас-Сиджль" http://www.alsjl.com, иорданский "Ахбар аль-Урдун" http://www.akhbar-jo.com и др.

В результате изучения вышеназванных источниковбыло найдено более 4700 сайтов различных СМИ, организаций, партий, политических движений и др. Из них более 2250 пригодны для сбора материала в корпус арабского языка.

2. Получение ссылок на страницы. Для получения потока ссылок на страницы отобранных ресурсов существует несколько подходов, отличающиеся разной степенью сложности, генерируемой нагрузкой на информационный ресурс, уровнем сложности реализации и т.д. Одним из распространенных подходов ([8]) является разработка робота, рекурсивно обходящего сайт с заданной глубиной. Достоинством этого подхода является относительная простота его реализации. Однако возникает задача отделения страниц, содержащих статьи, от страниц с другим контентом. Такой робот создает существенную нагрузку на сайт, что может привести в действие различные механизмы защиты.

По результатам анализа материалов новостных сайтов были выбраны три способа получения ссылок на статьи, опубликованные на этих сайтах:

1. Получение и разбор RSS лент с последующим выкачиванием статей по ссылкам из этих лент.

2. Получение ссылок на статьи из файлов Sitemap.xml (карт сайтов).

3. Получение ссылок на новые материалы из Twitter.

RSS ленты

RSS – семейство XML-форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т.д. Информация из различных источников, представленная в формате RSS, может быть собрана, обработана и представлена пользователю при помощи специальных программ-агрегаторов или онлайн сервисов. RSS является распространенным средством доставки информации об обновлениях на сайтах, и практически все популярные CMS (системы управления контентом) и фреймворки для веб-разработки содержат функционал для генерации RSS лент.Содержимое RSS лент обычно не защищается от автоматического скачивания контента. Крупные новостные порталы содержат несколько разделенных по тематике RSS лент, что упрощает распределение собранного контента по темам.

Карты сайтов (файлы Sitemaps)

Карты сайтов (Sitemaps) – XML-файлы с информацией для поисковых систем о страницах веб-сайта, которые подлежат индексации. Карты сайтов помогают поисковым роботам определить местонахождение страниц сайта, время последнего обновления страниц, частоту обновления, важность размещенного контента и т.д. К их достоинствам можно отнести гораздо больший (относительно RSS) объем ссылок. К недостаткам – то, что карты сайтов значительно реже обновляются и не содержат дополнительной информации (в частности, информации о тематике статьи). Таким образом, RSS ленты являются предпочтительным инструментом сбора свежего контента, а карты сайтов подходят для сбора архивов или для случаев, когда сайт не содержит функционала RSS лент.

Twitter

Twitter – сервис для публичного обмена короткими (до 140 символов) сообщениями с использованием веб-интерфейса, программ-клиентов и т.д. В настоящий момент, данный сервис является очень популярным средством «микроблоггинга» - публикации коротких сообщений, которые доступны всем пользователям вне зависимости от того, подписаны они на данные сообщения, или нет. Как правило, новостные порталы имеют собственную учетную запись в Twitter, где они публикуют ссылки на новые материалы и некоторые короткие комментарии. По аналогии с RSS лентами, крупные новостные порталы могут иметь несколько учетных записей для публикации сообщений различной тематики, что позволяет получать ссылки на статьи по интересующей тематике.

Несмотря на относительное удобство получения ссылок через Twitter API, этот способ ставит разработчиков системы в зависимость от стороннего сервиса и ограничений в работе с ним, что может негативно повлиять на работоспособность системы сбора корпусов текстов. При росте нагрузки и объемов собираемой информации Twitter может ограничить доступ к учетной записи или API.

Тем не менее, существует и другой подход к работе с Twitter, который состоит в создании нескольких аккаунтов и подписке их на интересующие источники. Данный подход является более надежным, однако скорость сбора ссылок на новостные ресурсы является довольно низкой.

В текущей версии системы реализованы работа с RSS лентами и картами сайтов, модуль для работы с Twitter будет интегрирован в ближайшее время.

Таблица 1

Сравнительные характеристики источников информации

Характеристика	RSS	Sitemaps	Twitter
Сложность получения данных	Низкая	Низкая	Средняя
Скорость обновления (в среднем)	Высокая	Средняя в зависимости от конкретного сайта	Высокая
Использованиестороннего API	Нет	Нет	Да
Подходит для сбора архивных материалов	Нет	Да	Не всегда
Разбиение статей по тематике	В некоторых случаях	Нет	В некоторых случаях

3. Описание программной системы сбора корпусов текстов. Система сбора корпусов текстов содержит несколько модулей: модуль сбора ссылок на новостные материалы и скачивания материалов, модуль хранения информации, модуль интерфейса управления, модуль извлечения текстов. Система разработана на языке Python, с использованием ряда фреймворков и компонентов с открытым исходным кодом, и работает под управлением ОС Ubuntu.

3.1. Модуль сбора ссылок на новостные материалы и скачивания статей. Модуль сбора ссылок на новостные материалы и скачивания материалов содержит в себе функционал для выкачивания и разбора RSS лент и карт сайтов, выделяет из них ссылки на новостные материалы, делает запрос к подсистеме хранения для предотвращения повторного скачивания и далее получает содержимое веб-страниц, размещенных по этим ссылкам. Модуль сбора ссылок и скачивания контента работает в многопоточном режиме.

Модуль использует компонент feedparser для разбора RSS лент, который имеет открытые исходные коды, что позволяет легко модифицировать некоторые его части и подстраивать его под нужды системы. Функционал скачивания контента в системе построен на основе библиотеки с открытыми исходными кодами requests. Эта библиотека предоставляет удобный способ работы с протоколом HTTP(S), а также содержит базовый функционал для определения кодировок веб-страниц, на основе которого разработан модуль распознавания кодировки текста. В модуле распознавания кодировки реализована следующая последовательность действий:

- анализируется HTTP заголовок “Content-Type”. Если такой заголовок присутствует в ответе сервера, то значение поля “charset” берется в качестве кодировки.
- если данного заголовка нет, то анализируется тело HTML документа на предмет наличия тега <meta> с атрибутом “http-equiv” равным “Content-Type” или с атрибутом “charset”. Если такой тег в теле документа отсутствует, то документ отбрасывается и не сохраняется в подсистеме хранения информации. Если же такой тег присутствует в документе, то он используется в качестве значения для кодировки.

Данный алгоритм в целом соответствует поведению современных браузеров по распознаванию кодировок веб страниц, однако веб-браузеры не отбрасывают содержимое веб-страниц в ситуациях, когда нет полной уверенности, что кодировка документа определена верно. Для сбора корпусов текстов кодировка должна определяться с максимальной точностью.

Модуль хранения информации

Модуль хранения информации состоит из нереляционной СУБД MongoDB и набора скриптов для обработки данных в этой СУБД. В модуль хранения также входит брокер, распределяющий задания на скачивание контента по потокам модуля скачивания. В СУБД хранятся ссылки, полученные из RSS лент и карт сайтов, скачанные документы, работы системы, промежуточные итоги извлечения текстов, списки RSS лент и карт сайтов.

При сохранении скачанных документов производится их первичная очистка (удаление тегов <object>, <script>, <img>, <svg> и т.д.). Это позволяет без снижения объемов текстов уменьшить объем сохраняемых документов на 50–70 % процентов.

3.2. Интерфейс управления. Интерфейс управления предусматривает два типа пользователей – администратор и эксперт – лингвист. Администратор обладает неограниченными правами (рис. 1), в том числе на добавление новых пользователей, изменения очередей обработки данных, просмотра логов работы системы, удаления данных и т.д. Интерфейс лингвиста позволяет вносить и редактировать адреса RSS лент и карт сайтов, которые впоследствии будут проверены администратором (рис. 2). Также интерфейс позволяет просматривать различную статистическую информацию и метрики системы.

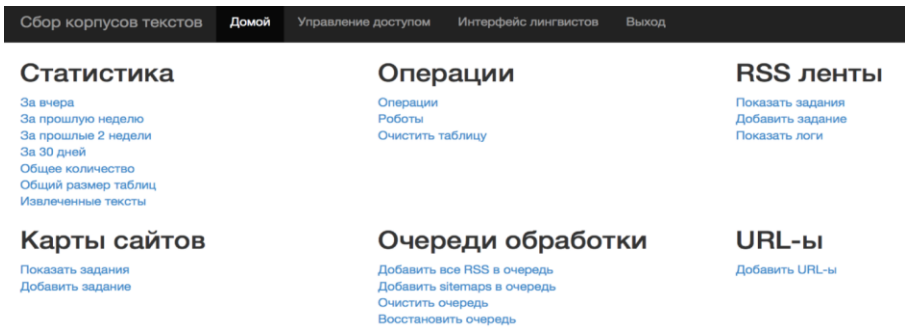

Рис. 1. Интерфейс администратора системы

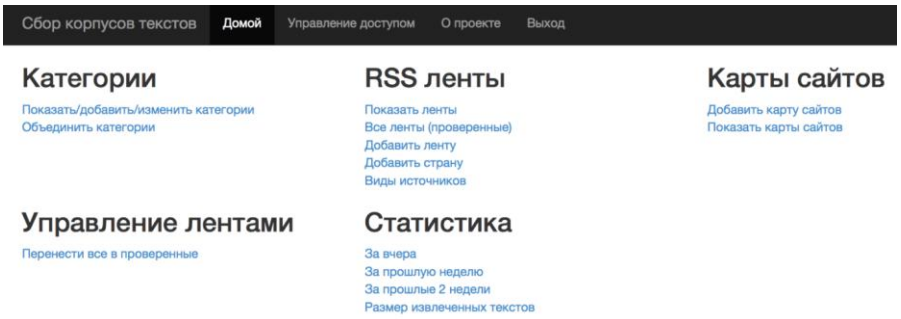

Рис. 2. Интерфейс пользователя системы

Интерфейс управления разработан на основе веб-фреймворкаDjango, для верстки страниц использован фреймворкBootstrap.

3.3. Модуль извлечения текстов. Модуль извлечения текстов предназначен для обработки скачанных веб-страниц. В модуле извлечения текстов реализованы алгоритмы поиска элементов веб-страницы, которые содержат текст статьи, алго-

ритмы выделения итогового текста из таких элементов, алгоритм анализа и извлечения шаблонов документов. Все алгоритмы построены на основе описанной ниже оригинальной оценке количества текста.

В реализации модуля обработки текстов для представления HTML документов в виде дерева используются библиотеки BeautifulSoup и lxml. Обе программных библиотеки имеют высокую производительность, и могут корректировать ошибки в HTML разметке.

Оценка количества текста

Любой HTML документ представим в виде дерева, в котором корнем является тег <html>. Для поддеревьев этого дерева введем характеристику количества текста, которая будет рассчитываться только для тегов , <p>, и некоторых других редко встречающихся тегов, которые используются небольшим количеством новостных агентств. Значения оценки количества текста лежат в диапазоне от 0 до 1.

Алгоритм вычисления оценки количества текста для одиночного тега div, p, span:

- Из HTML документа извлекается поддерево, вершиной которого является данный тег.
- Из данного тега и всех дочерних тегов в поддереве удаляются атрибуты и их значения.
- Поддерево конвертируется в строку A.
- Из строки A удаляются все открывающие и закрывающие теги, например “<a>” и т.д., в результате получается строка B, содержащая только текст, который был в значениях тегов.
- Отношение len(A)/len(B) длины строки A к длине строки B и является оценкой количества текста для данного тега.

Оценка количества текста лежит в основе алгоритма извлечения текста из HTML документа, а также служит для принятия решения о том, содержит ли данный HTML документ полезный для включения в корпус текст или нет.

Оценка количества текста по всему HTML документу проводится следующим образом. Дерево HTML документа обходится в глубину, и для каждого узла, для которого определена эта оценка, выделяется все поддерево HTML документа. Из тега этого поддерева и всех дочерних его тегов удаляются все атрибуты и их значения. Числовым значением оценки служит отношение длины (в символах) всего извлеченного текста из такого поддерева (в данном случае извлеченный текст – текстовая информация, которая располагается внутри тегов) к длине (в символах) этого поддерева (с учетом названий тегов).

В большинстве случаев, данная оценка для элемента HTML документа, содержащего фрагменты текста, близка к 1, так как доля количества символов в названиях тегов после удаления атрибутов дочерних тегов весьма мала.

Базовый метод извлечения текста из одиночного HTML документа

В основе способа лежит описанная выше характеристика количества текста, а также способ фильтрации блоков меню, рекламных блоков и блоков ссылок, которые могут приводить к искажению результатов и не должны появляться в итоговом тексте.

Фильтрация блоков меню, рекламных блоков и блоков ссылок:

- HTML документ естественным образом представляется в виде дерева тегов. Данное дерево обходится рекурсивно по тегам div, span.

- Если родителем текущего тега является тег ul или li, то обработка данного поддерева прекращается и осуществляется переход к следующему.
- Для поддерева с вершиной в теге div, span рассчитывается количество дочерних тегов в узлах первого уровня p и br и число дочерних тегов a по всем уровням поддерева.
- Если число тегов a, рассчитанное упомянутым способом, более чем на 5 больше числа подсчитанных тегов p или br, то также осуществляется переход к обработке следующего тега, а текущий тег отбрасывается.
- Если тег прошел все проверки, то для него рассчитывается оценка количества текста, описанная выше.

Метод извлечения текста из одиночного HTML документа:

- Для HTML документа применяется описанная выше процедура, после работы которой имеем набор тегов и оценки количества текста в них.
- На выход подается тег, на котором оценка количества текста принимает максимальное значение.

Определение шаблонов HTML документов

Алгоритм определения шаблонов позволяет значительно улучшить качество извлечения текстов за счет использования информации, полученной в ходе анализа множества документов от одного ресурса для обработки последующих документов, скачанных с этого ресурса. Эффективность использования алгоритмов выделения шаблонов объясняется тем ([10]), что современные веб-приложения используют шаблонизаторы для построения итоговой веб-страницы перед непосредственной отправкой ее пользователю, что уменьшает вариативность верстки страницы.

Шаблоном будем называть какой-либо тег в HTML документе и его атрибуты, которые позволяют идентифицировать (найти) данный тег в другом документе. Определение шаблонов проводится непосредственно перед извлечением текстов из набора страниц. Алгоритм определения шаблонов обрабатывает документы и в каждом из них пытается выделить шаблон при помощи описанного выше способа извлечения текста из одиночного HTML документа. Для извлечения шаблона выполняются следующие шаги:

- Из каждого отдельного HTML документа в коллекции, полученной из одного источника описанным выше способом извлекается шаблон и получается оценка количества текста.
- Если оценка количества текста больше определенной экспериментально величины, то данный документ попадает в дальнейшую обработку, его шаблон запоминается.
- Если оценка количества текста меньше определенной экспериментально величины – документ отбраковывается.
- Шаблоном коллекции документов считается тот шаблон, который получен в большинстве одиночных HTML документов коллекции после ее обработки.

Извлечение текста из коллекции HTML документов

Извлечение текстов из коллекции HTML документов проводится в два этапа. На первом этапе для коллекции документов определяется шаблон. На втором этапе происходит извлечение текста из каждого документа коллекции при помощи полученного шаблона.

4. Предобработка (очистка) корпусов текстов. Нецелевая информация, попадающая в корпус, может, например, содержать многократное дублирование каких-то сообщений, нетекстовые фрагменты или фрагменты текстов на других языках [11]. Для формирования языковых моделей эту избыточную информацию необходимо убирать, проводя очистку корпуса. Для больших корпусов процедуры очистки проводить вручную невозможно, поэтому возникает необходимость создания автоматических или, по крайней мере, удобных интерактивных процедур.

Методы очистки автоматически собираемых корпусов рассматривались рядом авторов. В [12] приводятся данные, как те или иные методы уменьшают размер очищаемого корпуса. В [13] сравнивается эффективность различных процедур очистки для корпусов английского, арабского и бенгальского языков. В [14] описаны этапы очистки корпуса чешского языка, содержащего, помимо статей, материалы соцсетей.

При анализе собранных корпусов выявились две основных причины появления нецелевой информации: нечеткое дублирование одних и тех же текстов, а также наличие фрагментов на нецелевых языках и нетекстовой информации (содержание и оформление таблиц, остатки html разметки и подобная информация). Сильно влияет на объем корпуса наличие дубликатов текстов. Явление дублирования связано с перепечаткой новостными мировыми агентствами материалов друг друга с минимальным редактированием или без него. В дублирующих текстах могут быть ссылки на источник, а могут и не быть. Степень дублирования определяется политикой новостных агентств, газет, журналов и организаций – источников новостей, их связями друг с другом, важностью и особенностями самого перепечатываемого материала.

4.1. Удаление нечетких дубликатов. Подробный обзор и анализ современных алгоритмов поиска нечетких дубликатов Web-документов приведен в [15] и [16]. Для описываемой системы сбора новостных текстов достаточно использовать метод поиска нечетких дубликатов по словарному составу. Опишем его. Для каждого текста из корпуса строится словарь употребленных в нем слов. Пусть V_1 и V_2 – словари текстов t_1 и t_2. Если величина $\frac{|V_1 \cap V_2|}{|V_1 \cup V_2|}$, где в числителе и знаменателе стоят мощности соответственно пересечения и объединения словарей, превосходит заданный порог, то тексты t_1 и t_2 считаются дубликатами. Однако количество таких сравнений квадратично зависит от числа статей в корпусе и для больших корпусов обнаружение нечетких дубликатов по такому правилу является вычислительно трудоемким. В самом деле, пусть T – длительность периода (количество дней), за который собран корпус. Для простоты предположим, что интенсивность поступления материалов в корпус является постоянной и составляет K статей в день. Тогда всего в корпусе содержится KT статей. Тогда количество необходимых попарных сравнений равно $\binom{KT}{2} = O\left(T^2\right)$. Однако особенности нашего корпуса таковы, что дубликатами могут являться только статьи с близкими датировками. Поэтому, задав максимум модуля разности возможных дат дублирующих текстов, данное правило поиска дубликатов можно уточнить, считая, что если даты текстов t_1 и t_2 различаются больше, чем на заданный максимум, то t_1 и t_2 не являются дубликатами. Обозначим выбранный максимум Δ. В таком случае все попарные сравнения происходят в промежутке Δ, и их количество оценивает-

ся как $O\left(\Delta^2\right)$, а число таких промежутков за период T, очевидно, можно оценить как $O\left(\frac{T}{\Delta}\right)$. Таким образом, общее число попарных сравнений оценивается как $O\left(\Delta^2\right)O\left(\frac{T}{\Delta}\right)=O\left(\Delta T\right)$.

4.2. Отбор текстов по списку редко встречающихся слов. Список формируется из редко встречаемых слов (с частотой встречаемости во всем корпусе меньше заданной) и непереведенных системой автоматического перевода слов из оставшихся. Отсекаются тексты, в которых доля слов из сформированного списка превышает заданный порог.

4.3. Отбор текстов по частоте встречаемости символов алфавита. Процедура направлена на поиск текстов, содержащих большое количество символов, отсутствующих в алфавите языка. Таким текстом может оказаться, например, некорректно скачанный текст с присутствующей в нем html разметкой в виде тегов. Задается базовый алфавит символов и вычисляется процентное отношение этих символов в тексте к общему числу символов. Отбор текстов происходит по фиксированному порогу.

Процедуры реализованы программно и позволяют обрабатывать текстовые корпуса объемом в десятки гигабайт в приемлемое время.

При обработке процедурой удаления нечетких дубликатов, в зависимости от числа и аффилированности собираемых источников, от 30 % (для малоресурсных языков, например, татарского) до 60 % (для английского) текстов были определены как дубликаты и удалены из корпуса.

Процедура отбора текстов по списку редко встречающихся слов оказалась эффективной при удалении текстов на языках, отличных от языка корпуса, а также текстов, не подходящих по содержанию из-за большого присутствия имен собственных, аббревиатур, слов с ошибками написания.

Предложенные процедуры очистки корпусов позволяют значительно уменьшить размер строящихся языковых моделей и сократить время их построения, сохраняя при этом точность моделей.

5. Итоги работы системы. Сравнение точности обученных языковых моделей. В основе разработанной программной системы лежит микросервисная архитектура, что позволяет легко проводить горизонтальное масштабирование и разворачивать систему сбора корпусов текстов на современных облачных платформах. Исходя из анализа необходимых на текущий момент вычислительных ресурсов и требований по удобству работы пользователей, используемые аппаратные средства реализованы в виде двух серверов в стоечном исполнении. Используются процессоры серии IntelXeon E3, суммарное количество ядер – 18, суммарный объем оперативной памяти – 64 ГБ, объем дисковой подсистемы, включающей в себя быструю и медленную составляющие, – 5 ТБ.

К середине 2016 года в обработке находилось более 20000 новостных источников на 14 языках из 70 стран мира. Весь перечень источников обходится каждые 2 часа. Для различных языков собраны корпуса от 500 Мб до 20 Гб, при этом список языков и источников постоянно дополняется.

Опишем эксперимент, характеризующий качество собираемых корпусов на примере английского языка. С октября 2014 по январь 2015 гг. описанной системой по 376 источникам из 8 стран был собран англоязычный новостной корпус по общественно-политической тематике, содержащий приблизительно 281 Мб текста. Он был разбит на две части, обучающую, размером 250 Мб, и тестовую, размером 31 Мб.

Собранный корпус сравнивался с двумя англоязычными корпусами, TheOpenAmericanNationalCorpus (OANC) ([17]), корпус американского английского, включающий различные жанры и стили письма, собранный в период 1990–2012 гг., содержащий приблизительно 55 Мб текста, и англоязычной частью корпуса EuropeanParliamentProceedingsParallelCorpus 1996–2011, release 7 (2012) ([18]), содержащей приблизительно 290 Мб текста. Для этого по всем трем корпусам был составлен общий словарь объемом 200000 слов, и по каждому корпусу построены триграммные модели на словах, с использованием сглаживания Кнессера-Нея ([19], [20]).

Для всех трех корпусов на материале тестовой части собранного корпуса были вычислены перплексии построенных моделей. Для текста, представленного последовательностью слов $w_1 w_2 ... w_K$, перплексия вычисляется следующим образом:

$$PPL = \sqrt[K]{\prod_{i=1}^{K} \frac{1}{P\left(w_i \middle| w_{1...i-1}\right)}} = 2^{-\frac{1}{K}\sum_{i=1}^{K} \log_2 P\left(w_i | w_{1...i-1}\right)},$$

где $w_{1...i-1}$ – отрезок текста с первого до $i-1$-го слова, а $P\left(w_i \middle| w_{1...i-1}\right)$ – вычисленная по модели условная вероятность появления на i-ом месте в тексте слова w_i при условии, что перед ним наблюдался отрезок $w_{1...i-1}$. Вычисленные значения перплексий приведены в таблице.

Таблица 2

Значения перплексий

Обучающий корпус	Даты текстов корпуса	Объем обучающего корпуса, Мб	Значение перплексии на тестовом множестве
OANC	1990-2012	55	471,52
Europarl_v7	1996-2011	290	642,83
Собранный	2014-2015	250	151,25

Значительное расхождение в вычисленных значениях перплексий связано с тем, что структура и стиль (новостной формат), тематика (общественно-политическая) и время происхождения у тестовых и обучающих текстов из собранного корпуса весьма близки, в отличие от других корпусов.

6. Основные результаты и выводы. Разработанная и реализованная технология сбора и предобработки корпусов текстов позволяет с высокой скоростью собирать корпуса новостных текстов, структурированные по стране происхождения, дате написания, тематике, источнику инаращивать объемы существующих корпусов для построения более точных языковых моделей с использованием актуальных новостей.

Для новостных текстов точность моделей, построенных на собранных корпусах, существенно превосходит точность моделей, построенных на свободно распространяемых корпусах.

БИБЛИГРАФИЧЕСКИЙ СПИСОК

1. *Кипяткова И.С., Карпов А.А.* Автоматическая обработка и статистический анализ новостного текстового корпуса для модели языка системы распознавания русской речи // Информационно-управляющие системы. – 2010. – № 4 (47). – С. 2-8.
2. *Мещеряков Р. В.* Структура систем синтеза и распознавания речи // Известия Томского политехнического университета. – 2009. – Т. 315, № 5. – С. 127-132.
3. *Мельников С.Ю., Пересыпкин В.А.* О применении вероятностных моделей языка для обнаружения ошибок в искаженных текстах // Вестник компьютерных и информационных технологий. – 2016. – № 5. – С. 29-33.
4. *Rosenfeld R.* Two decades of statistic language modeling: where do we go from here? // in Proceedings of the IEEE. – 2000. – Vol. 88, Issue 8. – P. 1270-1278.
5. *Мельников С.Ю., Пересыпкин В.А.* Тенденции развития языковых моделей в задачах распознавания, аспекты точности и вычислительной трудоемкости // Материалы 8-й Всероссийской мультиконференции по проблемам управления МКПУ-2015. с. Дивноморское. – Т. 1. – С. 85-87.
6. Proceedings of the 4th International Workshop on Spoken Language Technologies for Under-resourced Languages SLTU-2014. – St.Petersburg, Russia, 2014. – 268 p.
7. *Vu N.T., Schlippe T., Kraus F., Schultz T.* Rapid Bootstrapping of five Eastern European Languages using the Rapid Language Adaptation Toolkit // In: Proc. of Interspeech 2010, Japan, Makuhari. – P. 865-868.
8. *Biemann C., Bildhauer F., Evert S., Goldhahn D., Quasthoff U., Schäfer R., Simon J., Swiezinski L., Zesch T.* Scalable construction of high-quality web corpora // Journal for Language Technology and Computational Linguistics. – 2013. – No. 28 (2). – P. 23-60.
9. *Schlippe T., Gren L., Vu N.T., Schultz T.* Unsupervised Language Model Adaptation for Automatic Speech Recognition of Broadcast News Using Web 2.0 // Interspeech 2013, 25-29 August 2013, Lyon, France. – P. 2698-2702.
10. *Kim C., Shim K.* TEXT: Template Extraction from Heterogeneous Web Pages // IEEE Transactions on Knowledge and Data Engineering. – 2011. – Vol. 23, Issue 4. – P. 612-626.
11. *Sivakumar P.* Effectual Web Content Mining using Noise Removal from Web Pages // Wireless Personal Communications. – 2015. – Vol. 84 (1). – P. 99-121.
12. *Eckart T., Quasthoff U., Goldhahn D.* The Influence of Corpus Quality on Statistical Measurements on Language Resources // in: Proc. of the 8 Int. Conf. on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 2012. – P. 2318-2321.
13. *Sarkar A., De Roeck A., Garthwaite P.* Easy measures for evaluating non-English corpora for language engineering. Some lessons from Arabic and Bengali // Dep. of Comp., Faculty of Math. and Comp., The Open University, Walton Hall, UK. Tech. Rep. №2004/05. – P. 1-5.
14. *Spoustova J., Spousta M.* A high-quality web corpus of Czech // in: Proc. of the 8 Int. Conf. on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 2012. – P. 311-315.
15. *Зеленков Ю.Г., Сегалович И.В.* Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL’2007, Переславль-Залесский, Россия, 2007 г. – С. 166-174.
16. *Xiao C., Wang W., Lin X., Xu Y. J., Wang G.* Efficient similarity joins for near-duplicate detection // ACM Transactions on Database Systems (TODS). – August 2011. – Vol. 36, No. 3. – P. 1-41.
17. https://www.anc.org/oanc/.
18. http://www.statmt.org/europarl/.
19. *Kneser R., Ney H.* Improved backing-off for m-gram language modeling // In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. – Vol. I. – Detroit, Michigan: 1995. – P. 181-184.
20. *Chen S.F., Goodman J.* An empirical study of smoothing techniques for language modeling // Computer Science Group, Harvard University, Cambridge, Massachusetts, TR-8-98, August, 1998.

REFERENCES

1. *Kipyatkova I.S., Karpov A.A.* Avtomaticheskaya obrabotka i statisticheskiy analiz novostnogo tekstovogo korpusa dlya modeli yazyka sistemy raspoznavaniya russkoy rechi [Automatic processing and statistical analysis of news text corpus for language model recognition systems for Russian speech], Informatsionno-upravlyayushchie sistemy [Information control systems], 2010, No. 4 (47), pp. 2-8.
2. *Meshcheryakov R.V.* Struktura sistem sinteza i raspoznavaniya rechi [The structure of the systems of synthesis and speech recognition], *Izvestiya Tomskogo politekhnicheskogo universiteta* [Bulletin of the Tomsk Polytechnic University], 2009, Vol. 315, No. 5, pp. 127-132.
3. *Mel'nikov S.Yu., Peresypkin V.A.* O primenenii veroyatnostnykh modeley yazyka dlya obnaruzheniya oshibok v iskazhennykh tekstakh [On the application of probabilistic language models to detect errors in distorted texts], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* **[**Herald of computer and information technologies], 2016, No. 5, pp. 29-33.
4. *Rosenfeld R.* Two decades of statistic language modeling: where do we go from here?, *in Proceedings of the IEEE*, 2000, Vol. 88, Issue 8, pp. 1270-1278.
5. *Mel'nikov S.Yu., Peresypkin V.A.* Tendentsii razvitiya yazykovykh modeley v zadachakh raspoznavaniya, aspekty tochnosti i vychislitel'noy trudoemkosti [Trends in the development of language models in pattern recognition, aspects of precision and computational complexity], *Materialy 8-y Vserossiyskoy mul'tikonferentsii po problemam upravleniya MKPU-2015* [Proceedings of 8-th all-Russian multiconference on problems of control, mcpo-2015]. s. Divnomorskoe, Vol. 1, pp. 85-87.
6. Proceedings of the 4th International Workshop on Spoken Language Technologies for Under-resourced Languages SLTU-2014. St. Petersburg, Russia, 2014, 268 p.
7. *Vu N.T., Schlippe T., Kraus F., Schultz T.* Rapid Bootstrapping of five Eastern European Languages using the Rapid Language Adaptation Toolkit, *In: Proc. of Interspeech 2010, Japan, Makuhari,* pp. 865-868.
8. *Biemann C., Bildhauer F., Evert S., Goldhahn D., Quasthoff U., Schäfer R., Simon J., Swiezinski L., Zesch T.* Scalable construction of high-quality web corpora, *Journal for Language Technology and Computational Linguistics,* 2013, No. 28 (2), pp. 23-60.
9. *Schlippe T., Gren L., Vu N.T., Schultz T.* Unsupervised Language Model Adaptation for Automatic Speech Recognition of Broadcast News Using Web 2.0, *Interspeech 2013, 25-29 August 2013, Lyon, France,* pp. 2698-2702.
10. *Kim C., Shim K.* TEXT: Template Extraction from Heterogeneous Web Pages, *IEEE Transactions on Knowledge and Data Engineering,* 2011, Vol. 23, Issue 4, pp. 612-626.
11. *Sivakumar P.* Effectual Web Content Mining using Noise Removal from Web Pages, *Wireless Personal Communications,* 2015, Vol. 84 (1), pp. 99-121.
12. *Eckart T., Quasthoff U., Goldhahn D.* The Influence of Corpus Quality on Statistical Measurements on Language Resources, *in: Proc. of the 8 Int. Conf. on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 2012,* pp. 2318-2321.
13. *Sarkar A., De Roeck A., Garthwaite P.* Easy measures for evaluating non-English corpora for language engineering. Some lessons from Arabic and Bengali, *Dep. of Comp., Faculty of Math. and Comp., The Open University, Walton Hall, UK. Tech. Rep. №2004/05*, pp. 1-5.
14. *Spoustova J., Spousta M.* A high-quality web corpus of Czech, *in: Proc. of the 8 Int. Conf. on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 2012,* pp. 311-315.
15. *Zelenkov Yu.G., Segalovich I.V.* Sravnitel'nyy analiz metodov opredeleniya nechetkikh dublikatov dlya Web-dokumentov [Comparative analysis of methods for determining near-duplicate Web documents], *Trudy 9-y Vserossiyskoy nauchnoy konferentsii «Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollektsii» – RCDL'2007, Pereslavl'-Zalesskiy, Rossiya, 2007 g.* [Proceedings of 9-th all-Russian scientific conference "digital libraries: advanced methods and technologies, digital collections" – RCDL'2007, Pereslavl, Russia, 2007], pp. 166-174.
16. *Xiao C., Wang W., Lin X., Xu Y. J., Wang G.* Efficient similarity joins for near-duplicate detection, *ACM Transactions on Database Systems (TODS),* August 2011, Vol. 36, No. 3, pp. 1-41.
17. Available at: https://www.anc.org/oanc/.
18. Available at: http://www.statmt.org/europarl/.

19. *Kneser R., Ney H.* Improved backing-off for m-gram language modeling, *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. I, Detroit, Michigan: 1995, pp. 181-184.
20. *Chen S.F., Goodman J.* An empirical study of smoothing techniques for language modeling, *Computer Science Group, Harvard University, Cambridge, Massachusetts, TR-8-98, August, 1998*.

Статью рекомендовал к опубликованию д.т.н., профессор И.И. Левин.

Белозеров Андрей Александрович – ФГУП «НТЦ «Орион»; e-mail: melnikov@linfotech.ru; 127018, г. Москва, ул. Образцова, д. 38, стр. 1; сотрудник.

Вахлаков Дмитрий Владимирович – сотрудник.

Пересыпкин Владимир Анатольевич – научный консультант; к.т.н.

Мельников Сергей Юрьевич – ООО «Лингвистические и информационные технологии»; зам. директора; к.ф.-м.н.

Сидоров Евгений Сергеевич – сотрудник.

Belozerov Andrey Alexandrovich – FGUP "NTC "Orion""; e-mail: melnikov@linfotech.ru; 38, Obraztsova street, build. 1, Moscow, 127018, Russia; worker.

Vakhlakov Dmitriy Vladimirovich – worker.

Peresypkin Vladimir Anatol'evich – research consultant; cand. of eng. sc.

Melnikov Sergey Yur'evich – OOO "Lingvisticheskie I informatsionye tehnologii" (Limited Liability Company); deputy Director; cand. of phys.-math. sc.

Sidorov Evgeniy Sergeevich – worker.

УДК 004.056.55 DOI 10.18522/2311-3103-2016-12-4254

Л.К. Бабенко, Д.В. Голотин, О.Б. Макаревич

СОЗДАНИЕ И ИССЛЕДОВАНИЕ МАЛОРЕСУРСНОЙ РЕАЛИЗАЦИИ ПОТОЧНОГО ШИФРА TRIVIUM*

В современном мире Интернет вещей особое место занимает легковесная криптография, наиболее эффективно обеспечивающая безопасность сетевых структур различного типа. Одной из проблем легковесной криптографии является оптимизация аппаратных решений с целью повышения эффективности их использования. Легковесная криптография – новое малоисследованное направление. Предметом данной статьи является поточный шифр Trivium. Данный шифр является финалистом проекта eSTREAM, по профилю 2 (поточные шифры для аппаратной реализации). Целью работы является реализация и исследование аппаратной модели легковесного шифра Trivium, в сравнении с другими его реализациями. Результаты работы могут быть использованы для решения задач по защите информации в условиях ограниченных ресурсов. Шифр Trivium является одним из самых эффективных в плане соотношения быстродействия, надёжности и легковесности. Для определения характеристик представленной аппаратной реализации шифра, на разработанной модели проведен анализ экспериментальных данных. Устройство шифрования реализовано на плате Марсоход2bis, на основе ПЛИС cyclone EP4CE6E22C8 с 6272 элементами. Частота внутреннего генератора микросхемы 100 МГц. Шифрующее устройство обрабатывает данные с компьютера, посылаемые по виртуальному com-порту, реализованному через USB. Передача данных из компьютера на ПЛИС и обратно осуществляет программа-клиент Serial Com port v1.2. В статье приводится схема устройства и его бло-

* Работа поддержана грантом РФФИ № 15-07-00595.