Impact Factor:					
ISRA (India)	= 3.117	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	РИНЦ (Russia)	= 0.156	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.716	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

SOI: 1.1/TAS DOI: 10.15863/TAS
International Scientific Journal
Theoretical & Applied Science
p-ISSN: 2308-4944 (print) **e-ISSN:** 2409-0085 (online)
Year: 2019 **Issue:** 07 **Volume:** 75
Published: 10.07.2019 **http://T-Science.org**

QR – Issue QR – Article

Vadim Andreevich Kozhevnikov
Peter the Great St.Petersburg Polytechnic University
Senior Lecturer
vadim.kozhevnikov@gmail.com

Elina Vladimirovna Zhorova
Peter the Great St.Petersburg Polytechnic University
student
elinius@yandex.ru

THE SYSTEM OF THEMATIC AND STATISTICAL INFORMATION PROCESSING OF PRINT AND ONLINE PUBLICATIONS

Abstract: *The work is devoted to the system of thematic and statistical processing of information of printed and online publications using MongoDB DBMS. The purpose of creating such a system is to increase the efficiency of information and analytical departments of institutions. The overview of existing solutions was conducted. The creation of a thematic and statistical information processing system for print and online publications is described. A description of the subject area and a review of existing information retrieval systems of news arrays are done. The functional requirements for the system are described, the development and testing process is considered.*

Key words: *Internet publications, information retrieval, keywords, word statistics.*
Language: *Russian*
Citation: Kozhevnikov, V. A., & Zhorova, E. V. (2019). The system of thematic and statistical information processing of print and online publications. *ISJ Theoretical & Applied Science, 07 (75)*, 67-78.
Soi: http://s-o-i.org/1.1/TAS-07-75-14 ***Doi***: crossref https://dx.doi.org/10.15863/TAS.2019.07.75.14
Classifiers: *Computer science, computer engineering and automation.*

СИСТЕМА ТЕМАТИЧЕСКОЙ И СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ИНФОРМАЦИИ ПЕЧАТНЫХ И ИНТЕРНЕТ-ИЗДАНИЙ

Аннотация: *Работа посвящена системе тематической и статистической обработки информации печатных и интернет-изданий с использованием СУБД MongoDB. Цель создания такой системы - повышение эффективности работы информационно-аналитических отделов учреждений. В процессе работы проводился обзор существующих решений. Описано создание системы тематической и статистической обработки информации печатных и интернет-изданий. Приведено описание предметной области, обзор существующих систем поиска информации новостных массивов. Описаны функциональные требования к системе, рассмотрен процесс разработки и тестирования.*

Ключевые слова: *интернет-издания, информационный поиск, ключевые слова, статистика слов.*

Введение

В складывающейся мировой политической обстановке продолжает расти роль СМИ, как инструмента воздействия на массовое сознание населения. Развитие современных средств телекоммуникации и информационных технологий позволило существенно расширить возможности по представлению информационных массивов потребителю. Значительно увеличилась интенсивность информационного потока, под которым находится большинство граждан. Сложно переоценить влияние интенсивности и содержания новостного потока на эффект воздействия на человека. В связи с этим, перед различными государственными и коммерческими информационно-аналитическими учреждениями

возникают задачи по исследованию данного потока. Изучению подлежат содержание сообщений, их интенсивность, взаимная корреляция, характеристики изданий и авторов, полнота и достоверность информации, представляемой потребителям и т.д. Поэтому крайне важным является повышение эффективности работы информационно-аналитических отделов учреждений.

Постановка задачи и анализ предметной области

При организации и ведении информационно-аналитической деятельности (ИАД) специалистам приходиться решать ряд специфических проблем, таких как:

- отсутствие актуальных данных об источниках достоверной тематической информации и предметной области;
- значительные временные и трудовые затраты, выделяемые для определения достоверности источников;
- выделение значительного временного ресурса на отбор и ввод сведений;
- представление большого количества разрозненной, зачастую недостоверной, информации по определенным тематикам;
- отсутствие возможности автоматизированного ввода;
- отсутствие специальных инструментов обработки информации, необходимых при осуществлении ИАД;
- отсутствие специализированного автоматизированного поиска по источникам информации в зависимости от тематики исследования;
- представление информационных блоков посредством различных носителей в широком многообразии форматов.

Средства сбора информации, которые имеются в распоряжении профессионального аналитика, не отличаются многообразием. Основные - сеть Интернет и публикации в различных письменных источниках. Но задача поиска, отбора, обработки данных, а также их последующего анализа является весьма сложной даже при работе с первичными данными, полученными из каналов СМИ. Важнейшей особенностью сети Интернет, как источника информации, является принцип предоставления информации по требованию [1]. Что делает ее на данный момент самым легким и быстрым источником данных. В традиционной системе автор сталкивается с экспертизой своего произведения перед его публикацией: научные издательства оценивают научную ценность и достоверность, коммерческие – актуальность и популярность темы, оригинальность формы [2] и т.д. Ложная, повторяющаяся, неактуальная информация в значительной мере отсеивается. Однако, информация, попадающая в сеть Интернет, в большинстве случаев не проходит проверок. Актуальность и достоверность информации, получаемой из сети Интернет, является одной из проблем информационной аналитики. Одним из ее решений является использование заслуживающих доверия источников. Накопленные к настоящему времени объемы информации вместе с темпами ее роста определяют актуальность и значимость информационного поиска, в решение многих задач ИАД. Информационный поиск – совокупность логических и технических операций, имеющих конечной целью нахождение документов, сведений о них, фактов, данных, релевантных запросу потребителя [3].

Одним из видов информационного поиска является тематический поиск, который ориентирован на нахождение документов по их содержанию. Общая схема такого поиска заключается в формулировании пользователем некоторого запроса относительно содержания документа и отборе из множества доступных документов тех, которые удовлетворяют запросу. Такой вариант поиска удобен прежде всего, тем, что нет необходимости в предварительном разделении документов по различным категориям. Особенно это актуально при значительном объеме доступных документов, высокой динамике их обновления или отсутствии некоторых реквизитов. Основная проблема тематического поиска – это сложность однозначной автоматической интерпретации содержания текстов документов и формулировок информационных потребностей пользователей. Эта проблема обусловлена отсутствием какой-либо регулярной структуры у текстовых документов на естественном языке. Информационные ресурсы, содержащие такие документы, принято называть неструктурированными или слабоструктурированными. При работе с большим количеством информации, специалист сталкивается с проблемой ее обработки и хранения. Множество разнообразных форматов хранения информации затрудняет её обработку. Специалисту приходится производить различные манипуляции по поиску информации, структуризации, хранению. Одним из решений данной проблемы является создание хранилища информации, позволяющее сохранять и обрабатывать данные различных форматов. Во многих поисковых системах для уменьшения времени поиска используется полнотекстовый поиск, который является разновидностью тематического поиска.

Процесс поиска текстовой информации, реализуемый типичной поисковой системой, включает в себя следующие этапы:

– формализация пользователем поискового запроса (представление пользователем в том или ином виде своих информационных потребностей);

– предварительный отбор документов по формальным признакам наличия интересующей информации (например, наличие в тексте документа одного из слов запроса, если запрос формулируется на естественном языке);

– анализ отобранных документов (лингвистический, статистический);

– оценка соответствия смыслового содержания найденной информации требованиям поискового запроса (ранжирование).

Всю совокупность представленных на сегодняшний день методов тематического анализа текста можно разделить на две группы:

– лингвистический анализ;

– статистический анализ.

Первый ориентирован на извлечение смысла текста по его семантической структуре, второй – по частотному распределению слов в тексте. В свою очередь, лингвистический анализ можно разделить на четыре взаимодополняющих вида анализа: лексический, морфологический, синтаксический, семантический. Статистический анализ – это, как правило, частотный анализ в тех или иных его вариациях. Суть такого анализа заключается в подсчете количества повторений слов в тексте и использовании результатов подсчета для конкретных целей, например, вычисление весовых коэффициентов ключевых слов [4].

Нами был произведен анализ существующих решений в этой области - агрегатор новостных и медийных порталов Agregator.PRO [5], информационно-аналитическая система InfoStream [6], информационно-аналитическая система ПрессИндекс [7], локальная поисковая система Архивариус 3000 [8], Интернет-библиотека русскоязычных СМИ Public.Ru [9]. Изучив существующие системы сбора, обработки и хранения информационных массивов новостных изданий, можно выделить их основные недостатки:

- большинство систем не позволяет загружать в информационное хранилище статьи в виде текстовых файлов. Такие статьи не могут быть использованы в поиске и составлении статистики наравне с загруженными с сайтов интернет-изданий;

- большинство систем не предоставляет возможность делать выгрузку результатов поиска в текстовом формате, что не позволяет использовать результаты для дальнейшей работы вне ИС;

- рассмотренные системы не позволяют предприятию самостоятельно определять используемые источники публикаций, а также частоту и время добавления статей в БД.

Поэтому построение системы тематической и статистической обработки информационного массива, предоставляемым печатными и интернет-изданиями, позволяющей пользователю получать информацию из источников, производить быстрый поиск по различным критериям и получать полную и удобную статистику, является очень актуальной задачей.

Общие требования к системе

Нами были изучены существующие модели систем тематической и статистической обработки информационного массива, предоставляемым печатными и интернет-изданиями. Были проанализированы различные функции и методы работы данных систем и выделены особенности, негативно сказывающиеся на работе и репутации систем среди пользователей подобных ресурсов. Требования, улучшающие работу системы и позволяющие создать хранилище данных, позволяющие получать корректную и актуальную информацию из информационных массивов:

- система должна хранить данные интересующих пользователя интернет-изданий;

- система должна своевременно получать данные интернет-изданий;

- система должна предоставлять возможность пользователю вручную загружать в хранилище текстовые файлы;

- система должна предоставлять возможность поиска статей по:

- источнику;

- автору;

- рубрике, с использованием синонимов;

- дате публикации,

- названию статьи;

- слову, содержащемуся в тексте статьи, с использованием его словоформ;

- система должна предоставлять возможность автоматического выделения ключевых слов и тематики из текста публикации;

- система должна предоставлять возможность автоматического статистического анализа публикаций по различным критериям и вывода полученной статистики;

- система должна предоставлять возможность выгрузки результатов поиска в текстовом формате.

Система должна включать в себя информационно связанные между собой, но разделенные по типу информационного пространства, подсистемы.

Информационное пространство системы делится на следующие подсистемы, предполагающие различный уровень доступа:

- подсистема пользовательского доступа – инструменты и разделы ИС к функциональности которых имеют доступ все пользователи ИС;

- подсистема административного доступа - инструменты и разделы для управления содержанием и наполнением ИС, к функциональности которых имеют доступ только администраторы ИС.

Для получения пользовательского или административного доступа в подсистему, пользователь должен пройти процедуру аутентификации на уровне операционной системы.

Прикладное программное обеспечение ИС состоит из:

- подсистема «Пользовательский инструментарий»;
- подсистема администрирования и управления содержанием «Инструментарий администратора».

Подсистема «Пользовательский инструментарий» выполняет функции:

- добавления новых статей в БД системы;
- поиска и чтение, хранимых в системе статей;
- автоматического выделения ключевых слов и тематик новостных статей;
- сбора статистической информации по новостным статьям печатных и интернет-изданий.

Подсистема администрирования и управления содержанием «Инструментарий администратора» предназначена для анализа и обеспечения функционирования ИС, резервного копирования и восстановления, своевременного добавления информации в ИС.

Так как подсистемы разрабатываются в рамках одного технического контура и используют одну базу данных (БД), требования к способам и средствам связи для информационного обмена не предъявляются.

ИС должна предусматривать в себе механизм для сбора информации из сети Интернет, а также других систем в формате HTML, XML, TXT для формирования базы данных новостных статей.

Одним из инструментов, который должна предоставлять ИС - диагностирование основных процессов системы. При возникновении ошибок или аварийных ситуаций, инструменты диагностики должны сохранять информацию, необходимую разработчику для идентификации.

При разработке ИС должна предусматриваться возможность модернизации программного обеспечения. В том числе, необходимо предусмотреть возможность дальнейшего увеличения производительности системы, путем масштабирования.

После критического сбоя серверной операционной системы или СУБД, в процессе выполнения пользовательских задач, должно быть обеспечено восстановление данных в базе данных до состояния на момент окончания последней нормально завершенной перед сбоем транзакции. Проверка целостности данных и настройка резервного копирования должна обеспечить восстановление данных на момент окончания последней нормально завершенной транзакции.

Требования к функциям, выполняемым системой

Подсистема «Пользовательский инструментарий» предназначена для:

- добавления новых статей в БД системы;
- поиска и чтение, хранимых в системе статей;
- автоматического выделения ключевых слов и тематик новостных статей;
- сбора статистической информации по новостным статьям печатных и интернет-изданий.

Доступ к ИС предоставляется только зарегистрированным в операционной системе пользователям. В рамках данной подсистемы должны быть предусмотрены инструменты, позволяющие зарегистрированным пользователям ИС просматривать, добавлять и получать статистические данные по новостным статьям, размещенных в БД.

Размещение статей должно производиться пользователями при помощи электронной формы, содержащей следующие поля:

- название источника;
- рубрика;
- название статьи;
- текст статьи (прикрепляется файл формата TXT);
- автор(ы) статьи;
- дата публикации.

Каждая статья, загруженная в базу данных ИС, должна содержать набор атрибутов:

1. Источник:

- для статьи, загруженной из интернет-издания, источник должен быть определён автоматически, исходя из названия статьи на странице новостного сайта;
- для статьи, загруженной пользователем, название источника заполняется пользователем, либо помечается значением «undefined».

2. Рубрика (тэг):

- для статьи, загруженной из интернет-издания, рубрика должна определяться автоматически, исходя из названия рубрики, в которой опубликована статья на сайте интернет-издания;
- для статьи, загруженной пользователем, название рубрики заполняется пользователем, либо определяется системой автоматически исходя из содержания публикации.

3. Название статьи:

- для статьи, загруженной из интернет-издания, название должно определяться

автоматически, исходя из названия статьи на сайте интернет-издания;

- для статьи, загруженной пользователем, название заполняется пользователем, либо помечается меткой «undefined».

4. Текст статьи:

- для статьи из интернет-издания текст загружается автоматически;

- для статьи, загруженной пользователем, текст загружается пользователем из файла с форматом TXT.

5. Автор(ы) статьи:

- для статьи, загруженной из интернет-издания, имя и фамилия автора(ов) загружаются автоматически. При их отсутствии ставится метка «undefined»;

- для статьи, загруженной пользователем, имя и фамилия автора(ов) заполняется пользователем, либо помечается меткой «undefined».

6. Дата публикации:

- для статьи, загруженной из интернет-издания, дата публикации должна определяться автоматически, исходя из даты публикации статьи на сайте интернет-издания.

- для статьи, загруженной пользователем, дата публикации заполняется пользователем, либо заполняется автоматически текущей датой.

7. Статистика слов:

- заполняется автоматически, при загрузке статьи в БД.

8. Статистика имен собственных:

- заполняется автоматически, при загрузке статьи в БД.

9. Ключевые слова:

- заполняется автоматически, при загрузке статьи в БД.

10. Дата загрузки статьи в БД:

- заполняется автоматически, при загрузке статьи в БД.

В подсистеме должны быть реализованы следующие функции:

1. Сортировка. Позволит сортировать документы по названию, дате, тематике и т.д.

2. Поиск статей, размещенных в базе данных, по:

- источнику;
- дате;
- автору (авторам);
- словам, поиск которых должен производиться в тексте статьи, при этом результаты поиска должны выводиться по порядку релевантности статьи относительно искомого слова.

3. Вывод статистики встречаемости слова по критериям:

- источник;
- период времени публикации статьи;

4. Поиск по словам должен включать в себя:

- поиск по словосочетанию;
- поиск по слову с использованием различных словоформ одного слова и с удалением коротких и стоп-слов, включающих в себя:
- знаки препинания;
- отдельно стоящие буквы алфавита;
- союзы, междометия, причастия, предлоги, местоимения.

Подсистема «Инструментарий администратора» предназначена для обеспечения поддержки ИС, резервного копирования и восстановления, своевременного добавления информации в ИС, анализа использования.

Подсистема должна обеспечивать возможность:

- добавления шаблонов для автоматической загрузки с сайтов интернет-изданий, настройка которых еще не произведена;
- изменение шаблонов для автоматической загрузки с сайтов интернет-изданий, уже входящих в список используемых;
- удаление шаблонов для автоматической загрузки с сайтов интернет-изданий, информация которых больше не используется.

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Уровень хранения данных в системе должен быть построен на основе современных СУБД.

Доступ к данным должен быть предоставлен только авторизованным пользователям с учетом их служебных полномочий, а также с учетом категории запрашиваемой информации.

Технические средства, обеспечивающие хранение информации, должны располагаться на территории предприятия и использовать современные технологии, позволяющие обеспечить повышенную надежность хранения данных и оперативную замену оборудования.

В состав системы должна входить специализированная подсистема резервного копирования и восстановления данных. Данные должны быть защищены от разрушений при авариях и сбоях в электропитании системы путем создания резервных копий.

Построение решения задачи

Система тематической и статистической обработки информации печатных и интернет-изданий обладает наиболее востребованными функциональными свойствами и позволяет решать ежедневные задачи по мониторингу и поиску публикаций при ведении ИАД. Модель информационной системы состоит из следующих модулей:

- информационное хранилище;
- автоматический алгоритм получения публикаций из сети Интернет;

- автоматический алгоритм тематической обработки публикаций;
- автоматический алгоритм статистической обработки публикаций;
- модуль автоматизированного поиска публикаций;
- веб-интерфейс.

Каждая составляющая модели выполняет особые функции и имеет структуру, позволяющую реализовывать алгоритмы для решения поставленных задач.

Построение решения задачи: информационное хранилище

Основной функцией информационного хранилища является хранение текста публикации с сопутствующими атрибутами. Информационное хранилище должно обеспечивать высокую отказоустойчивость и масштабируемость. При этом оно должно быстро осуществлять операции чтения-записи и обладать возможностью полнотекстового поиска. Лучше всего для этого подходит NoSQL документно-ориентированное СУБД.

Разработанная модель содержит единственную коллекцию Articles, которая хранит тексты публикаций с атрибутами:

- sourceArticle - источник статьи;
- tagArticle - рубрика, к которой относится статья;
- headlineArticle - заголовок статьи;
- textArticle - текст статьи;
- dateArticle - дата публикации;
- authorArticle - подколлекция, содержащая автора или авторов статьи;
- statisticsUseSynonims - подколлекция, представляющая собой статистику для поиска с синонимами и содержащая поля word – слово и occurrence – частота вхождения слова в текст;
- statisticsProperName - подколлекция, представляющая собой статистику для поиска имен собственных и содержащая поля word - слово и occurrence - частота вхождения слова в текст;
- keyWordCollection - подколлекция, содержащая ключевые слова, автоматически выделенные из текста публикации.

При загрузке публикации из сети Интернет, все поля проверяется на корректность типа данных и заполняются автоматически, при отсутствии какого-либо поля оно помечается значением “undefined”, что позволяет в дальнейшем проследить либо наличие некорректных данных, либо полное отсутствие этих данных в источнике.

При ручной загрузке публикаций, пользователем заполняются поля:

- источник статьи;
- рубрика;
- заголовок;
- текст статьи;
- дата публикации;
- автор(ы) статьи.

При этом поля «Источник статьи», «Заголовок» и «Текст статьи» являются обязательными для заполнения.

Поле «Автор(ы) статьи» может содержать коллекцию авторов одной статьи.

При появление нового документа на запись в БД поля «Статистика для поиска с синонимами», «Статистика для поиска имен собственных» и «Коллекция ключевых слов» заполняются системой автоматически. Первые два поля организованы в виде словарей, где ключами являются все слова текста, а значениями - встречаемость слов в тексте. Данные поля служат для получения статистики.

Построение решения задачи; автоматический алгоритм получения публикаций из сети Интернет

Информационная система тематической и статистической обработки новостных массивов позволяет в автоматическом режиме получать данные с сайтов новостных изданий.

Структура сетевого новостного текста содержит следующие элементы:

1) Рубрика - результат максимального сжатия содержания текста, отражающий его главный предмет.

2) Тема - обобщённое содержание текста. Тема указывается автором в заголовочной части новостного текста и первом предложение статьи.

3) Подтемы - компоненты содержания текста, направленные на детализацию описываемой в тексте ситуации, раскрытие аспектов основной темы новостного текста.

Комплекс, образованный заголовком и подзаголовком статьи, входит в перечень опубликованных новостных сообщений в определенной рубрике и представляет собой анонс содержания текста. Его цель – кратко представить информацию, передаваемую текстом. Комплекс является гиперссылкой, при переходе по которой на следующий уровень сайта, происходит отображение основного текста статьи.

Заголовок, расположенный вместе с основным новостным текстом, может совпадать или отличаться от заголовка-гиперссылки. Его особенность в качественных интернет-изданиях заключается в краткости (не более 10 слов) и точной передаче текстового содержания, поэтому он лишен экспрессивной окраски [10]. Заголовок новостной статьи используется для загрузки в поле «Заголовок» БД.

Дата публикации статьи, как правило, ставится автоматически при добавлении на сайт интернет-издания и публикуется вместе с текстом

и заголовком статьи. Однако, недобросовестные интернет-издания, вместо даты публикации статьи могут выводить дату последнего изменения для увеличения посещаемости сайта [11]. Качественные интернет-издания обычно имеют в своей HTML-разметке и дату публикации, и дату последнего изменения.

Исходными данными для модуля автоматического получения публикаций из сети интернет, является массив URL рубрик источника публикаций. После этого производится разбор HTML-страницы на наличие ссылок на отдельные публикации. В итоге система получает HTML-страницы новостных статей, при обработке которых получаются поля. Данные поля в дальнейшем обрабатываются и записываются в БД.

Структуры веб-источников различаются, однако можно выделить схожие черты, которые позволяют использовать одинаковый функционал для получения URL отдельной публикации. Для каждой статьи ИС получает HTML-код, из которого выделяются необходимые для загрузки в БД поля. Структура HTML-кода статей для одного источника однородна, поэтому для получения данных, достаточно выбрать содержащие их контейнеры и в дальнейшем использовать для всего сайта. При этом контейнер должен включать хотя бы один атрибут со значением, однозначно его определяющим и выделяющим среди множества других тэгов.

Построение решения задачи: автоматический алгоритм тематической обработки текста публикаций

Система тематической и статистической обработки информации новостных массивов позволяет в автоматическом режиме обрабатывать текст статьи, выделяя ключевые слова и тематику статьи. Ключевые слова в контексте анализа текста представляют собой важные, общепонятные, ёмкие слова, набор которых дает полное описание текста для читателя. Другими словами, ключевые слова - это такой набор слов, который позволяет на своей основе восстановить смысл текста. Множество ключевых слов представляет собой свертку текста, поэтому поиск по ключевым словам эффективнее, чем поиск по любому слову из текста, потому что поиск в данном случае производится именно по семантической нагрузке.

Рубрикация текста необходима для его систематизации. Она позволяет выделить главный предмет и в дальнейшем использовать его для поиска публикаций одинаковой направленности.

Автоматическое выделение рубрики статьи необходимо в том случае, когда пользователь при ручном занесение публикации в хранилище не указывает рубрику, к которой относится статья. Задача поиска ключевых слов текста схожа с задачей выделения тематики, поэтому для решения обеих задач используют одинаковые методы. Одними из самых эффективных методов решения задачи являются алгоритмы кластеризации [12]. В текущей версии системы задача автоматического выделения рубрики статьи ещё не решена до конца, ведутся исследования, в настоящее время планируется применение самоорганизующихся карт Кохонена (SOM). К тому же последние работы показывают (см., например, [13]), что на больших массивах данных методы машинного обучения дают неплохие результаты по обработке текстов на русском языке.

Модуль автоматической обработки текста публикаций принимает на вход текст статьи, после чего из него удаляются лишние пробелы, знаки табуляции, знаки перевода строки. После обработки текст готов к внесению в БД.

Для автоматического заполнения полей «Статистика для поиска с синонимами» и «Статистика для поиска имен собственных» необходимо разбить обработанный текст на слова. Разделителями являются знак пробела, перевода строки и всевозможные знаки пунктуации. После разбиения текста на слова производится подсчет частоты вхождения слова в тексте. При этом слова из коллекции поля «Статистика для поиска с синонимами» приводятся к одному регистру. Это необходимо сделать из-за того, что для кодирования заглавных и строчных символов используются разные ASCII-коды, и система в дальнейшем может воспринять одно слово в разных регистрах, как два разных.

Число, показывающее сколько раз встречается слово в тексте, называется частотой вхождения слова. Если расположить частоты по мере убывания и пронумеровать, то порядковый номер частоты называется рангом частоты. Вероятность обнаружения слова в тексте равно отношению частоты вхождения слова к числу слов в тексте. Джордж Ципф (G.K. Zipf) популяризовал эмпирическую закономерность, которая стала носить название закона Ципфа, заключающуюся в том, что если взять достаточно длинный текст на естественном языке, то частота слова с n-ным рангом окажется приблизительно обратно пропорциональной n. Иными словами, если умножить вероятность обнаружения слова в тексте на ранг частоты, то получившаяся величина приблизительно постоянна для всех текстов на одном языке [14]: $f \cdot r/n = const$, где f – частота вхождения слов, r – ранг частоты, n – число слов в тексте. Если нарисовать график зависимости ранга слова от его частоты вхождения, то, как показали исследования вышеуказанных зависимостей для различных текстов, наиболее значимые слова текста лежат в средней части графика, так как

слова с максимальной частотой, как правило, являются предлогами, частицами, местоимениями (так называемые «стоп-слова»), а редко встречающиеся слова в большинстве случаев не имеют решающего значения. Таким образом, данная особенность может помочь правильно выбрать ключевые слова из текста. Процедура оптимального выбора ключевых слов, основанная на алгоритмах кластеризации и законе Ципфа, заключается в следующем:

1. «Стоп-слова» удаляются из текста;
2. Выделяются предварительные ключевые слова на основе методов кластеризации;
3. Вычисляется частота вхождения каждого предварительного ключевого слова и составляется список, в котором слова расположены в порядке убывания их частоты;
4. Выбирается диапазон частот, лежащий в середине списка. Слова, соответствующие данным частотам с наибольшей вероятностью являются ключевыми.

Ключевые слова сохраняются в поле «Коллекция ключевых слов» БД.

Рубрика анализируемого текста выбирается с помощью коллекции ключевых слов, имеющихся в БД названий рубрик и подключаемой внешней БД синонимов.

Построение решения задачи: автоматический алгоритм статистической обработки публикаций

С помощью полей «Статистика для поиска с синонимами» и «Статистика для поиска имен собственных» производится подсчет частоты употребления слова в статьях с использованием различных критериев. В набор критериев входят:

- промежуток даты публикации статьи;
- источник статьи;
- автор(ы) статьи.

Основной проблемой при подсчете частоты употребления слова является принадлежность его к именам собственным. Для ее решения необходимо выделить имена собственные, встречающиеся в тексте на этапе обработки. Критерии определения имен собственных [15]:

- имена собственные в кавычках пишутся с заглавной буквы (хотя бы первое слово: медаль «За отвагу на пожаре») и не имеют перед знаком кавычек двоеточия, а после знака кавычек - тире;
- имена собственные, не оформленные кавычками, в середине предложения пишутся с большой буквы;
- имена собственные могут иметь при себе инициалы с точкой;
- имена собственные могут быть написаны в виде одной заглавной буквы с точкой (инициалы), а после слово с заглавной буквы (фамилия);
- инициалы часто стоят после фамилии, поэтому при обнаружении в тексте заглавных букв с последующими точками следует обратить внимание на то, с какой буквы пишется предыдущее слово. Если оно тоже с заглавной, а перед ним нет точки, то это фамилия. Если предыдущее слово со строчной буквы – проверить последующее слово.

Построение решения задачи: модуль автоматизированного поиска публикаций

Система тематической и статистической обработки новостных массивов позволяет в автоматическом режиме производить поиск текста статьи по заданным критериям в БД. Для поиска текстовых данных в БД существуют различные инструменты:

- регулярные выражения;
- встроенные операторы поиска (например, оператор LIKE в некоторых СУБД) и т.д.

Для разрабатываемой ИС наиболее эффективным инструментом поиска текста является полнотекстовый поиск. Полнотекстовый поиск - это автоматизированный поиск документов, при котором отбор ведётся не по именам документов, а по их содержимому, всему или существенной части [16].

Все технологии полнотекстового поиска работают по схожему принципу. На основе текстовых данных строится индекс, который способен быстро искать соответствия по ключевым словам. Обычно, сервис поиска состоит из двух компонент - индексатора и поисковика. Индексатор получает текст на вход, делает обработку текста (получение нормальной формы слова (леммы), удаление стоп-слов и т.п.) и сохраняет все в индексе. Устройство такого индекса позволяет быстро проводить поиск по нему. Поисковик - интерфейс поиска по индексу - принимает от клиента запрос, обрабатывает слово или словосочетание и ищет его в индексе [17]. Для организации поиска в информационном хранилище системы тематической и статистической обработки новостных массивов используются следующие виды поиска:

- полнотекстовый поиск для поля textArticle;
- поиск по полному соответствию для поля sourceArticle;
- поиск с использованием регулярных выражений для поля authorArticle;
- поиск по промежутку для поля dateArticle.

Реализация системы

Проанализировав сформулированные требования и отобрав необходимые для реализации модулей системы алгоритмы, были выбраны следующие программные продукты:

- СУБД MongoDB 3.4;
- ASP.NET MVC 4.0 Framework. Язык программирования - C#;
- MongoDB .NET 2.2 Driver;

- SDK PullEnti.

Для работы с информационным хранилищем системы была выбрана NoSQL документо-ориентированная СУБД MongoDB 3.4. Выбор данного продукта обусловлен следующими причинами:

- СУБД MongoDB является свободно распространяемым продуктом;
- данная СУБД является документо-ориентированной (для нашей системы реляционная СУБД является избыточной) и поддерживает создание текстовых индексов;
- на основе текстовых индексов можно производить полнотекстовый поиск, поддерживающий морфологию русского языка;
- для хранения данных используются формат BSON, который позволяет быстро производить поиск и обработку данных;
- способность к горизонтальному масштабированию.

MongoDB .NET 2.2 Driver является официальным драйвером MongoDB для взаимодействия с платформой .NET. Основными достоинствами продукта являются:

- MongoDB .NET 2.2 Driver предоставляет возможность асинхронного взаимодействия с MongoDB;
- поддержка BSON сериализации;
- возможность построения запросов любых уровней сложности, включая запросы с использованием текстовых индексов. Использование для запросов нотации LINQ;
- возможность обращаться к объектам БД не только как к BSON-документам, но и как к объектам классов, определенных в C#;
- возможность внесения изменений в схему БД, а также добавление новых полей или подколлекций в документы.

Подключение к БД начинается с запуска сервера БД, после этого, с помощью конструктора по умолчанию ArticleContext, приложение создает экземпляр клиента подключения с указанием строки подключения. При создании коллекции Aricles БД для поля textArticle был создан текстовый индекс, который позволяет производить полнотекстовый поиск.

Для получения информации со страниц новостных сайтов используется библиотека C# HtmlAgilityPack [18]. Библиотека использует для разбора HTML-страниц XPath - язык запросов к элементам XML документа. Разбор HTML-страниц начинается с вызова метода GetListArticle класса, определённого для каждого новостного сайта. Класс содержит в себе шаблоны для разбора страниц, а также URL новостного сайта. Метод GetListArticle вызывает метод GetURL класса ToolParsing, который возвращает весь HTML-код разбираемой страницы. Далее начинается проход по всем узлам документа, для поиска ссылок для перехода на страницы статей. Узлом в данном случае является любой тэг HTML-кода, причем библиотека HtmlAgilityPack при поиске узла позволяет учитывать вложенность узла в другой узел и т.д. При проходе по всем узлам учитывается значение атрибута искомого тега. При нахождении необходимого тэга происходит считывание значения URL страницы статьи.

Для получения информации из HTML-кода страницы новостной статьи используется метод GetComponent класса ToolParsing. Метод использует вызов вспомогательных методов GetString, GetData и GetCollection того же класса для получения необходимых полей, а также метод CountOccurence класса StatisticsTool для сбора статистики текста. По окончанию работы модуля происходит сохранение статей с сопутствующими полями в БД с помощью асинхронного метода InsertDocsInArticlesCollection класса ArticleContext.

Для получения ключевых слов статьи используется SDK PullEnti [19]. Встроенные инструменты SDK PullEnti позволяют получать именные группы из текста - имена существительные или имена существительные с относящимися к ним прилагательными. SDK выделяет среди именных групп ключевые слова и подсчитывает их ранг - число, показывающее значимость слова относительно текста. Так, например, название географических объектов имеют наибольший ранг. Ранг определяется при семантическом анализе на основе выделения в тексте объектов и их взаимоотношений. Выявленные ключевые слова помещаются в поле keyWordCollection БД. Как уже отмечалось, в настоящее время проводится реализация алгоритма автоматического выделения из текста рубрики статьи. Первичная обработка текста для получения статистики происходит еще на этапе загрузки данных в БД. Затем, при введении пользователем необходимых критериев статистики, происходит поиск слова в полях statisticsUseSynonims и statisticsProperName. Вывод статистики происходит с помощью метода CreateChart контроллера HomeController. Метод принимает на вход коллекции значений, по которым строится график.

СУБД MongoDB и разработанный для взаимодействия с ней MongoDB .NET 2.2 Driver позволяют производить поиск данных в БД. Асинхронный метод FindArticles контроллера HomeController принимает на вход коллекцию введенных пользователем фильтров, после этого происходит поиск по фильтрам в БД.

Тестирование системы

Тестирование разработанной системы проводилось на 1614 публикациях, загруженных

из интернет-изданий «Комсомольская правда» [20] и «Lenta.ru» [21]. Статистика коллекции Articles изображена на Рис. 1.

```
> db.articles.stats()
{
        "ns" : "articledb.articles",
        "count" : 1614,
        "size" : 37255130,
        "avgObjSize" : 23082,
        "storageSize" : 16703488,
        "capped" : false,
        "wiredTiger" : {
                "metadata" : {
                        "formatVersion" : 1
                },
```

Рис.1 Статистика коллекции Articles

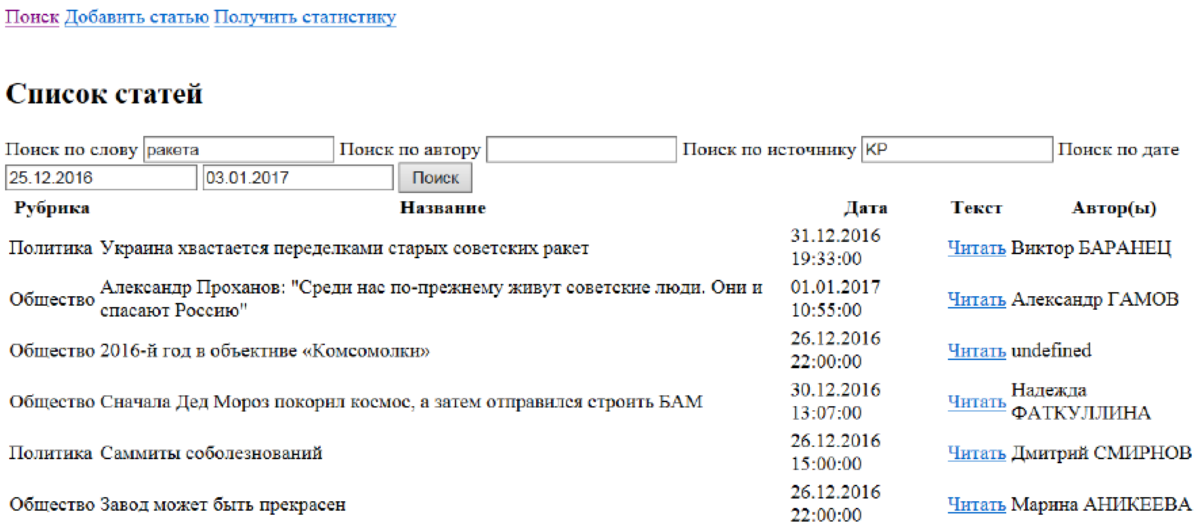
1) Было проведено выделение из текста публикаций ключевых слов. Пример результата работы алгоритма выделения ключевых слов из текста приведен на Рис. 2.

Найден гигантский гусь ростом 1,5 метра [22]. Палеонтологи, ведомые Марко Павиа (Marco Pavia) из Университета Турина (University of Turin), исследовали кости, обнаруженные в центральной Италии. И пришли к выводу, что они принадлежали огромному гусю, который водился тут 5-9 миллионов лет назад. Его назвали Garganornis ballmanni. Кости доисторического гуся, которые позволили восстановить его облик. Как полагают ученые, обнаруженный гусь был самым крупным в семействе утиных - крупнее всех других доисторических гусей, уток и даже лебедей. Рост этого Garganornis ballmanni достигал полутора метров. Почти, как у страуса. Вес - 22 килограмма. Жаль, что гусь вымер, а то запекали бы с яблоками. Ведь в те времена, когда он жил, запекать было еще некому. Крупнее, чем этот гусей на Земле не было. Огромный гусь не летал. Крылья имел маленькие, хотя и сильные. Вес нынешних гусей не превышает 7 килограммов, рост - полуметра.	6,12: ГУСЬ 3,97: Павиа М. 3,41: ДОИСТОРИЧЕСКИЙ ГУСЬ [ГУСЬ ДОИСТОРИЗМ] 3,15: ОГРОМНЫЙ ГУСЬ [ГУСЬ ОГРОМНОСТЬ] 2,96: Университет Турина 2,94: University Of Turin 2,86: Италия [IT]

Рис.2. Пример результата работы алгоритма выделения ключевых слов из текста

2) Был проведен статистический анализ встречаемости слов по определенным критериям.

3) Был проведен поиск публикаций в БД по различным критериям. Пример поиска приведен на Рис. 3.



Поиск Добавить статью Получить статистику

Список статей

Поиск по слову ракета Поиск по автору Поиск по источнику КР Поиск по дате
25.12.2016 03.01.2017 Поиск

Рубрика	Название	Дата	Текст	Автор(ы)
Политика	Украина хвастается переделками старых советских ракет	31.12.2016 19:33:00	Читать	Виктор БАРАНЕЦ
Общество	Александр Проханов: "Среди нас по-прежнему живут советские люди. Они и спасают Россию"	01.01.2017 10:55:00	Читать	Александр ГАМОВ
Общество	2016-й год в объективе «Комсомолки»	26.12.2016 22:00:00	Читать	undefined
Общество	Сначала Дед Мороз покорил космос, а затем отправился строить БАМ	30.12.2016 13:07:00	Читать	Надежда ФАТКУЛЛИНА
Политика	Саммиты соболезнований	26.12.2016 15:00:00	Читать	Дмитрий СМИРНОВ
Общество	Завод может быть прекрасен	26.12.2016 22:00:00	Читать	Марина АНИКЕЕВА

Рис.3. Пример поиска публикаций в БД по различным критериям

Результаты тестирования были признаны хорошими.

Заключение

В ходе разработки системы были произведен анализ существующих реализаций систем тематической и статистической обработки информации печатных и интернет-изданий, соответствующих проблем автоматического получения информации с новостных сайтов, выделения ключевых слов и тематики из текста, полнотекстового поиска. Разработанная система призвана повысить эффективность работы информационно-аналитических отделов государственных и коммерческих учреждений. Были разработаны и реализованы модули автоматического получения публикаций из сети Интернет, тематической и статистической обработки текста публикаций, автоматизированного поиска публикаций. Для представления полученных результатов был разработан веб-интерфейс. Разработанная система была протестирована на публикациях интернет-изданий «Комсомольская правда» и «Lenta.ru». Были получены корректные данные с сайтов газет, произведено выделение ключевых слов из текстов статей, выбрана текстовая статистика, а также выполнен поиск по различным критериям. Таким образом, применение разработанной системы можно считать оправданным.

References:

1. Kurnosov, Y. V., & Konotopov, P. Y. (2004). *ANALITIKA: metodologiya, tekhnologiya i organizatsiya informatsionno-analiticheskoy raboty*. (p.550). Moscow: Izdatel'stvo «Rusaki».
2. Zubets, V. V., & Il'ina, I. V. (2011). Otsenka dostovernosti setevoy informatsii. *Vestnik Tambovskogo universiteta. Seriya: Estestvennye i tekhnicheskie nauki. Vyp. №1, t. 16.* (p.410). Tambov: Izd-vo TGU.
3. Zinov'eva, N. B. (2001). *Dokumentovedenie*. Uchebno-metodicheskoe posobie. (p.208). Moscow: Profizdat.
4. Serebryanaya, L. V. (2011). *Informatsionnoe obespechenie finansovykh struktur*. Metodicheskoe posobie k laboratornym rabotam dlya studentov spetsial'nosti «Programmnoe obespechenie informatsionnykh tekhnologiy» vsekh form obucheniya. (p.43). Minsk: BGUIR.
5. (n.d.). Agregator Agregator.PRO: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://agregator.pro
6. (n.d.). Informatsionnaya sistema InfoStream: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://infostream.ua

7. (n.d.). PressIndeks: [Elektronnyy dokument]. Retrieved July 07, 2019, from https://pressindex.ru
8. (n.d.). Arkhivarius 3000: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://www.likasoft.com
9. (n.d.). Internet-biblioteka russkoyazychnykh SMI Public.Ru: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://www.public.ru
10. Bazhenova, E. Y. (2011). Strukturnaya organizatsiya setevogo novostnogo teksta. *Vestnik Chelyabinskogo gosudarstvennogo universiteta. Seriya: Filologiya. Iskusstvovedenie. Vyp. №85*. (p.98). Chelyabinsk: Izd-vo ChGU.
11. Shakin, M. (n.d.). *SEO khitrost' s datoy publikatsii*: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://shakin.ru/seo/publication-date.html
12. Soloshenko, A. N., Orlova, Y. A., & Rozaliev, V. L. (2014). Avtomaticheskoe vydelenie syuzhetov i tem iz potoka novostnykh soobshcheniy. *Izvestiya VolgGTU. Volgograd: VolgGTU*, p.250.
13. Kozhevnikov, V. A., & Oborin, P. A. (2019). Development of the automated student testing system using Python and NLP methods. ISJ Theoretical & Applied Science, 06 (74), 301-306. Doi: https://dx.doi.org/10.15863/TAS.2019.06.74.36
14. Popov, A. (n.d.). *Poisk v Internete - vnutri i snaruzhi:* [Elektronnyy dokument]. Retrieved July 07, 2019, from http://citforum.ru/pp/search_03.shtml
15. Lavrenenko, A. V. (2012). Algoritm opredeleniya imen sobstvennykh pri avtomaticheskom analize teksta. *Karpovskie nauchnye chteniya: sb. nauch. st. Vyp. 6: v 2 ch. Ch. 1*, redkol.: A.I. Golovnya (otv. red.) [i dr.] (Eds.). (pp.201-203). Minsk: Belorusskiy Dom pechati.
16. (n.d.). Polnotekstovyy poisk: [Elektronnyy dokument]. Retrieved July 07, 2019, from https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BF%D0%BE%D0%B8%D1%81%D0%BA
17. (n.d.). Polnotekstovyy poisk: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://ruhighload.com/post/%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9+%D0%BF%D0%BE%D0%B8%D1%81%D0%BA
18. (n.d.). HTMLAGILITYPACK: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://htmlagilitypack.codeplex.com/
19. (n.d.). PullEnti: [Elektronnyy dokument]. Retrieved July 07, 2019, from www.pullenti.ru
20. (n.d.). Komsomol'skaya pravda: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://www.spb.kp.ru/
21. (n.d.). LENTA.RU: [Elektronnyy dokument]. Retrieved July 07, 2019, from https://lenta.ru/
22. (n.d.). Nayden gigantskiy gus' rostom 1,5 metra: [Elektronnyy dokument]. Retrieved July 07, 2019, from http://www.spb.kp.ru/daily/26629/3648654/