DOI: 10.17323/1998-0663.2019.4.60.72

Проектирование структуры программной системы обработки корпусов текстовых документов

В.Б. Барахнин [a,b]
E-mail: bar@ict.nsc.ru

О.Ю. Кожемякина [a]
E-mail: olgakozhemyakina@mail.ru

Р.И. Мухамедиев [c,d,e]
E-mail: ravil.muhamedyev@gmail.com

Ю.С. Борзилова [a]
E-mail: i.borzilova@alumni.nsu.ru

К.О. Якунин [c,d]
E-mail: yakunin.k@mail.ru

[a] Институт вычислительных технологий, Сибирское отделение Российской академии наук
Адрес: 630090, г. Новосибирск, пр-т Академика Лаврентьева, д. 6

[b] Новосибирский национальный исследовательский государственный университет
Адрес: 630090, г. Новосибирск, ул. Пирогова, д. 1

[c] Satbayev University
Адрес: Казахстан, 050013, г. Алматы, ул. Сатпаева, д. 22а

[d] Институт информационных и вычислительных технологий
Адрес: Казахстан, 050010, г. Алматы, ул. Пушкина, д. 125

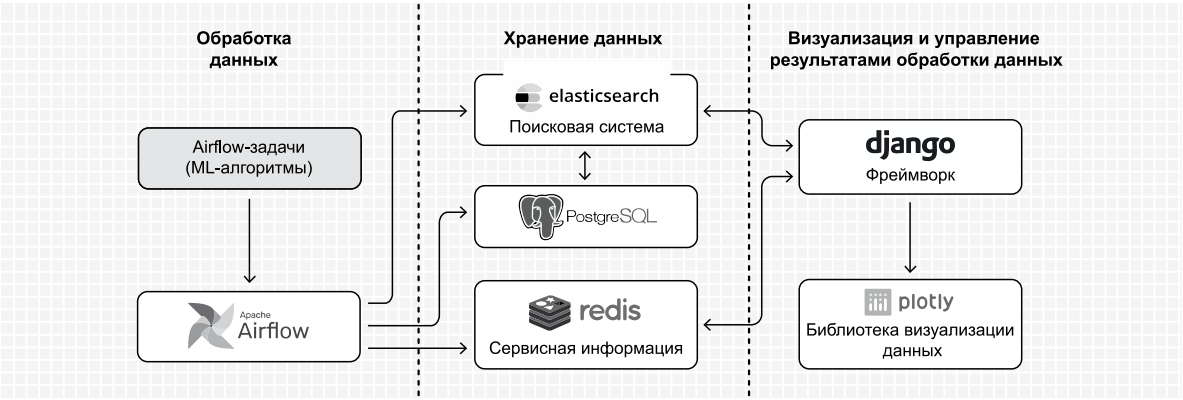
[e] Университет ISMA
Адрес: Латвия, LV-1019, г. Рига, ул. Ломоносова, 1

Аннотация

Одной из труднорешаемых задач в области интеллектуального анализа данных является разработка универсального инструментария для анализа текстов художественного и делового стиля. Популярным направлением развития алгоритмов обработки корпусов текстовых документов является использование методов машинного обучения, которые позволяют решать задачи обработки естественных языков. Основанием для проведения исследований в этой области являются такие факторы, как специфика структуры текстов художественного и делового стиля (что требует формирования отдельных наборов данных и, в случае использования методов машинного обучения, – дополнительных параметров при обучении), а также отсутствие укомплектованных систем массовой обработки корпусов текстовых документов для русского языка (в отношении научного сообщества в коммерческой среде существуют системы меньших масштабов, решающие узкоспециализированные задачи, например, определение тональности текста). Целью текущего исследования является проектирование и последующая разработка структуры системы обработки корпусов текстовых документов. При проектировании учитывались требования, предъявляемые к широкомасштабным системам: модульность, возможность масштабирования компонентов и их условная независимость. Проектируемая система представляет собой совокупность компонентов, каждый из которых сформирован и используется в виде Docker-контейнеров. Уровни системы:

обработка данных, хранение данных, визуализация и управление результатами обработки данных. На уровне обработки данных выполняется сбор (скраппинг) текстовых документов (например, новостных событий) и их дальнейшая обработка с помощью ансамбля методов машинного обучения, каждый из которых реализован в системе как отдельная Airflow-задача. Полученные результаты помещаются для хранения в реляционную базу данных, а для увеличения быстродействия поиска по данным (более 1 млн. единиц) используется инструмент ElasticSearch. Визуализация статистики, полученной в результате работы алгоритмов, осуществляется с использованием плагина Plotly. Администрирование и просмотр обработанных текстов доступны через веб-интерфейс с использованием фреймворка Django. Общая схема взаимодействия компонентов организована по принципу ETL (extract, transform, load). В настоящее время система используется для анализа корпусов новостных текстов с целью сравнительного анализа параметров текстов и средств массовой информации в целом. В перспективе планируется усовершенствование системы и опубликование компонентов в открытом репозитории GitHub для доступа научного сообщества.

Графическая аннотация



Ключевые слова: обработка естественных языков; потоковая обработка текстов; информационная система анализа текстов; разработка системы обработки корпусов текстов.

Цитирование: Барахнин В.Б., Кожемякина О.Ю., Мухамедиев Р.И., Борзилова Ю.С., Якунин К.О. Проектирование структуры программной системы обработки корпусов текстовых документов // Бизнес-информатика. 2019. Т. 13. № 4. С. 60–72. DOI: 10.17323/1998-0663.2019.4.60.72.

Введение

Современные методы интеллектуального анализа данных позволяют обрабатывать большие корпуса текстовых документов (объемом более 1 млн документов) с целью выявления как тех или иных свойств отдельных документов, входящих в корпус, так и закономерностей, характеризующих их совокупность. Поскольку указанные алгоритмы предполагают извлечение из текстов широкого спектра разноплановых характеристик (что нередко само по себе является комплексной задачей, решение которой подразумевает использование сложных и отнюдь не всегда быстродействующих алгоритмов), возникает необходимость хранения извлеченных характеристик (наряду с самими документами) в справочно-информационном фонде создаваемой программной системы. При этом информационная модель хранилища документов и их характеристик будет в значительной мере зависеть от типа исследуемых корпусов текстов и характера решаемых задач. Например, системы обработки новостных сообщений с целью выявления деструктивной информации [1] существенно отличаются от систем обработки научной информации [2] и, тем более, художественных текстов, как прозаических, так и поэтических [3].

Следует отметить, что одной из труднорешаемых задач является разработка универсального инстру-

ментария для анализа текстов художественного и делового стиля. Как указано в [4], «при распознавании слов делового текста наиболее существенным является фактор знакомства с текстом (его темой, структурой и наиболее частотными словами), ключевые слова и элементы темы распознаются сравнительно неплохо, конец текста предсказуем и хорошо распознается. Для художественного текста большая «опорность» приходится на начальный (преамбула) и срединный (развитие сюжета) композиционные фрагменты и по-разному соотносится с компонентами коммуникативного и смыслового членения: с темой для преамбулы, с диалогом (особенно ключевыми словами или ремой) для срединного фрагмента. Таким образом, говоря о структурах текста и процедурах анализа, мы должны учитывать разнообразные виды контекста, в частности, функциональный стиль, композиционную структуру и риторическую связность текста» [4].

В настоящее время обработка текстовой информации является активно развивающейся отраслью информационных технологий. Обзор работ в этой области имеется, например, в [5–8]. Отметим, что в последнее десятилетие основным направлением развития алгоритмов обработки корпусов текстовых документов является использование методов машинного обучения (см., например, [9–11]). В общем же случае для автоматического анализа текста можно выделить следующие подходы [12]:

1. На основе правил с использованием шаблонов (rule-based with patterns). Этот подход использует такие инструменты, как part-of-speech-taggers и parsers. Другой вариант – использование N-grams для выделения часто используемых сочетаний, которые объединяются в слова. В частности, при решении задач определения тональности текста этим N-grams присваиваются положительные или отрицательные оценки;

2. Машинное обучение без учителя (unsupervised learning). Главное отличие от машинного обучения с учителем – отсутствие ручной разметки для обучения модели. В случае статистической модели корпуса текстов наибольший вес в тексте имеют такие термины, которые чаще встречаются в этом тексте и одновременно встречаются в небольшом количестве текстов всей выборки;

3. Машинное обучение с учителем (supervised learning). Обучающая выборка размечается вручную экспертами в предметной области или dataset-инженерами. Затем размеченный корпус текстов используется для обучения различных классификаторов, среди которых часто применяются Naive Bayes Classifier [13], Support Vector Classifier (SVM) [14], а также композиции алгоритмов, например, бустинг [15], когда несколько методов машинного обучения могут объединяться в ансамбль (совокупность), в которой каждый следующий метод обучается на ошибках предыдущего, и искусственные нейронные сети (artificial neural networks, ANN) различных конфигураций [16, 17];

4. Гибридный метод (hybrid method). Данный подход может сочетать методы машинного обучения, а также использовать шаблоны правил;

5. Метод, основанный на теоретико-графовых моделях. При таком подходе используется разбиение корпуса текстов на слова, при этом каждое слово имеет свой вес. Такой вес используется, например, в задачах анализа определения тональности текста: некоторые слова имеют больший вес и сильнее влияют на тональность текста;

6. Предобученные модели на основе глубоких нейронных сетей (transfer learning), когда заранее обученная модель дообучается для решения специфических задач, например, весьма популярная в последнее время BERT [18].

В частности, задача анализа тональности текста неоднократно решалась и активно используется в коммерческих разработках. К последним можно отнести, например, систему лингвистического анализа текстов модульного типа Eureka Engine, которая позволяет извлекать новые знания и факты из неструктурированных данных больших объемов[1]. Кроме определения тональности документа, система решает задачи определения тематики текста (т.е. классификации), выделения именований и имен собственных (named-entity recognition, NER). Модуль автоматической классификации текстов TextClassifier реализован на основе машинного обучения, также имеются модули автоматического определения именованных сущностей, нормализации слов и морфоанализатор. Внутренняя структура системы не приводится. Система использовалась как инструмент для определения тональности текста в СМИ относительно одного и того же события [19]. Кроме того, можно отметить работу [20], в которой приведены результаты исследования методики ана-

[1] Eureka Engine: http://eurekaengine.ru/

лиза тональности текста на примере анализа сообщений сети Твиттер и рецензий портала «Кинопоиск». В качестве инструментария авторы использовали алгоритмы машинного обучения: метод опорных векторов, наивный Байесовский классификатор, методы случайных деревьев (random forest). Дополнительно в работе приведен обзор аналогичных работ в задачах анализа тональности текста.

Разнообразие алгоритмов обработки текстов на естественном языке предполагает возможность их реализации в виде отчуждаемого программного продукта. В силу этого структура создаваемой программной системы должна быть нацелена на ее взаимодействие как с конечным пользователем, так и с другими системами. В данной статье формулируются требования к структуре создаваемой программной системы и определяются ролевые функции пользователей. После этого описывается разработанная структура, включающая в себя подсистему обработки данных, хранилище и подсистему построения аналитических отчетов.

Основным особенностями разрабатываемой системы, выделяющими ее среди аналогов, являются:

1. Автоматическое тематическое моделирование, которое позволяет определять тренды в реальном времени, без ручного формирования списка ключевых слов или запросов. Это позволяет в оперативном режиме автоматически определять актуальные, социально значимые темы, что является критически важным при принятии управленческих решений в различных областях;

2. Экспертная разметка на уровне тематик позволяет на порядки уменьшить объем необходимых для разметки объектов (особенно по сравнению с использованием сетей глубокого обучения);

3. Из предыдущего пункта следует возможность оперативной и относительно недорогой разметки по произвольному набору критериев, не ограниченной только тональностью. Критерии могут быть выбраны индивидуально под конкретные требования клиента (например, оценка инновационности, социальной значимости, оппозиционности, социального доверия, инфляционных ожиданий и т.д.).

1. Постановка задачи

Процесс анализа текстов на естественном языке сводится к следующим последовательным шагам, на которых осуществляется анализ характеристик текста:

✦ инициализация – формирование корпуса текстов и его предобработка для последующего анализа;

✦ структурный анализ (только для поэтических текстов) – определение низкоуровневых характеристик текста (фонетика и метроритмика стихотворения);

✦ семантический анализ – определение смысловых конструкций с учетом синонимии и связывание именованных сущностей (named entity linking, NEL). Анализ научных текстов обычно ограничивается этим уровнем;

✦ прагматический анализ – определение жанрово-стилевых особенностей для художественных текстов; конструкции, определяющие деструктивное влияние для новостных сообщений и т.д.

✦ синтез проведенных исследований – определение влияния низших уровней на более высокие, а также агрегация результатов в удобном для восприятия и поиска виде.

Сформулируем требования к функциональности системы, исходя из ее целевого назначения: скрапинг, хранение, потоковая аналитика и формирование аналитических отчетов с визуализацией.

1. Надежное хранение корпусов текстов больших объемов, при этом система должна быть настроена на работу с разностилевыми текстами: научными, публицистическими, художественными;

2. Быстрый параллельный доступ, фильтрация и агрегация данных с целью потоковой обработки: предобработка, построение тематических моделей и классификаторов, агрегация и выгрузка для отчетов в реальном времени и т.д.;

3. Гибкость и возможность хранения неструктурированных и слабоструктурированных данных для поддержки возможности хранения и доступа к произвольным структурам данных с целью проведения статистического анализа и различных вычислительных экспериментов на основе современных методов анализа текстов.

Структура программной системы должна позволять решать масштабные задачи, состоящие в хранении корпусов объемов нескольких миллионов текстов и пакетной обработки в режиме онлайн нескольких тысяч документов. Таким, например, является проект мониторинга в реальном времени русскоязычных СМИ Республики Казахстан [21], предназначенный для создания отчетов следующих типов:

1. Тематическая структура новостных публикаций в казахстанских электронных СМИ – как на уровне крупных тем (экономика, образование, политика) и подтем (дошкольное образование, единый государственный экзамен, высшее образование и наука), так и на уровне информационных поводов

(конкретных узких тем, описывающих конкретное событие или группу близко связанных событий);

2. Оценка отдельных публикаций, тематик и СМИ по произвольному набору критериев. Такая оценка предполагает предварительную разметку экспертом или коллегией экспертов;

3. Отчеты и оповещения по выявленным аномалиям. Аномалии рассматриваются на двух уровнях: на уровне динамики (например, резкий рост публикаций по определенной теме или резкий рост публикаций с отрицательной оценкой по критерию «тональность») и на тематическом уровне – появление групп публикаций с нестандартной, «аномальной» тематикой, которая раннее не встречалась (например, тема криптовалют, тема феминизма в Казахстане и т.д.);

Первые два типа отчетов могут быть получены как в динамике по времени, так и статично (например, оценка СМИ по определенным критериям за последний год). Отчет по аномалиям предполагает анализ динамики с привязкой публикации ко времени.

Концептуальное проектирование включало формирование возможностей создаваемой программной системы. Создаваемая программная система должна обладать следующими возможностями:

1. Обеспечение доступа к корпусам текстов;

2. Автоматизированная обработка корпуса текстов, хранящихся в базе данных;

3. Занесение полученных характеристик в хранилище (базу данных);

4. Возможность гибкого планирования выполнения различных задач по обработке данных;

5. Статистическая обработка полученных характеристик и их представление в удобном для исследователя виде;

6. Обновление и улучшение применяемых алгоритмов для анализа корпуса текстов.

В рамках текущего исследования поставлена задача проектирования структуры системы обработки корпусов текстов на естественном языке. Область применения данной системы начинается с анализа корпусов текстов публицистического стиля. В дальнейшем область применения системы может будет распространена и на художественные тексты, в силу модульности системы и гибкости применяемых технологий.

Проектируемая система состоит из следующих подсистем:

1. Подсистема обработки данных. Используется совокупность гибридных методов (машинное обучение с учителем и словари);

2. Хранилища данных. Для обеспечения быстрого взаимодействия с пользователем, а также снижения потребления ресурсов используется несколько видов хранилищ;

3. Подсистема построения аналитики по полученным данным.

Информационная система должна учитывать этапы анализа текстов. Структура системы состоит из компонентов, перечисленных при описании постановки задачи. На этапе предобработки выполняется предварительная обработка текста для его дальнейшего анализа. Применяемые методы предобработки зависят от алгоритма, который работает с этими данными (обучение и анализ). Можно выделить следующие виды обработки:

✦ дающие в результате “bag of words”; к этому виду также можно отнести метод TF-IDF;

✦ дающие в результате обработки каждой смысловой единицы корпуса (например, новости) его “embedding” (векторизацию), например, распределение по токенам / словам / фразам / предложениям. В этом случае возможно использование рекуррентных нейронных сетей (recurrent neural networks, RNN);

✦ дающие в результате обработки каждой смысловой единицы корпуса один “text embedding”; для такой предобработки возможно использование стандартных методов классификации.

Структурный анализ применяется для текстов художественного стиля и может быть выполнен существующими на текущий момент инструментами, например, [22]. Семантический анализ может выполняться как на этапе предобработки текста (например, лемматизация слов), так и не выполняться вообще – выбранный инструментарий будет зависеть от методов машинного обучения и может изменяться с течением времени. Прагматический анализ в системе осуществляется с использованием совокупности алгоритмов машинного обучения и составленных частотных словарей. Синтез результатов обеспечивается путем агрегирования результатов в некоторые хранилища и вывод этих результатов в таком виде, который будет с наибольшей точностью удовлетворять потребностям пользователя.

На основе вышеописанных возможностей системы можно выделить следующие требования к разрабатываемой системе:

✦ обеспечивать работу подсистем в виде отдельных независимых компонентов, каждый из которых можно оперативно заменить при необходимости;

✦ организовывать распараллеливание вычислений, в том числе на нескольких машинах;

✦ реализовывать автоматизированную обработку корпуса текстов по запросу пользователя;

✦ вести мониторинг выполнения задач в реальном времени, в том числе обеспечивать оперативное информирование об исключениях;

✦ выводить данные по результатам анализа текстов в интерфейсе пользователя;

✦ обновлять применяемые в системе алгоритмы для улучшения качества анализа и расширения их области применения.

2. Структура системы

Все компоненты системы организованы в виде Docker-контейнеров. Все контейнеры имеют доступ к одной виртуальной сети, что обеспечивает возможность обмена данными с использованием стандартных сетевых протоколов (TCP). Такая реализация обеспечивает работу подсистем в виде независимых компонентов, каждый из которых можно заменить при необходимости.

Взаимодействие компонентов, – подсистемы построения аналитики и подсистемы обработки данных, – осуществляется с помощью системы хранения. Общая схема взаимодействия компонентов организована по принципу ETL (extract, transform, load): от пользователя поступает запрос на получение данных в ElasticSearch (если данные используются редко) или в Redis (если данные используются часто). Кроме того, подсистема обработки использует Airflow-scheduler, который записывает в Redis информацию о распределении задач по “workers”; они, в свою очередь, отчитываются в Redis о статусе выполнения своих задач. В процессе проектирования могут применяться компоненты сообразно их целевому назначению.

Визуализация структуры системы показана на *рисунке 1.*

Анализ текстовых корпусов (на данном этапе – корпуса новостных сообщений в казахстанских русскоязычных интернет-СМИ объемом 1,5 млн документов с постоянным пополнением) осуществляется по загрузке “workers”. Новые документы

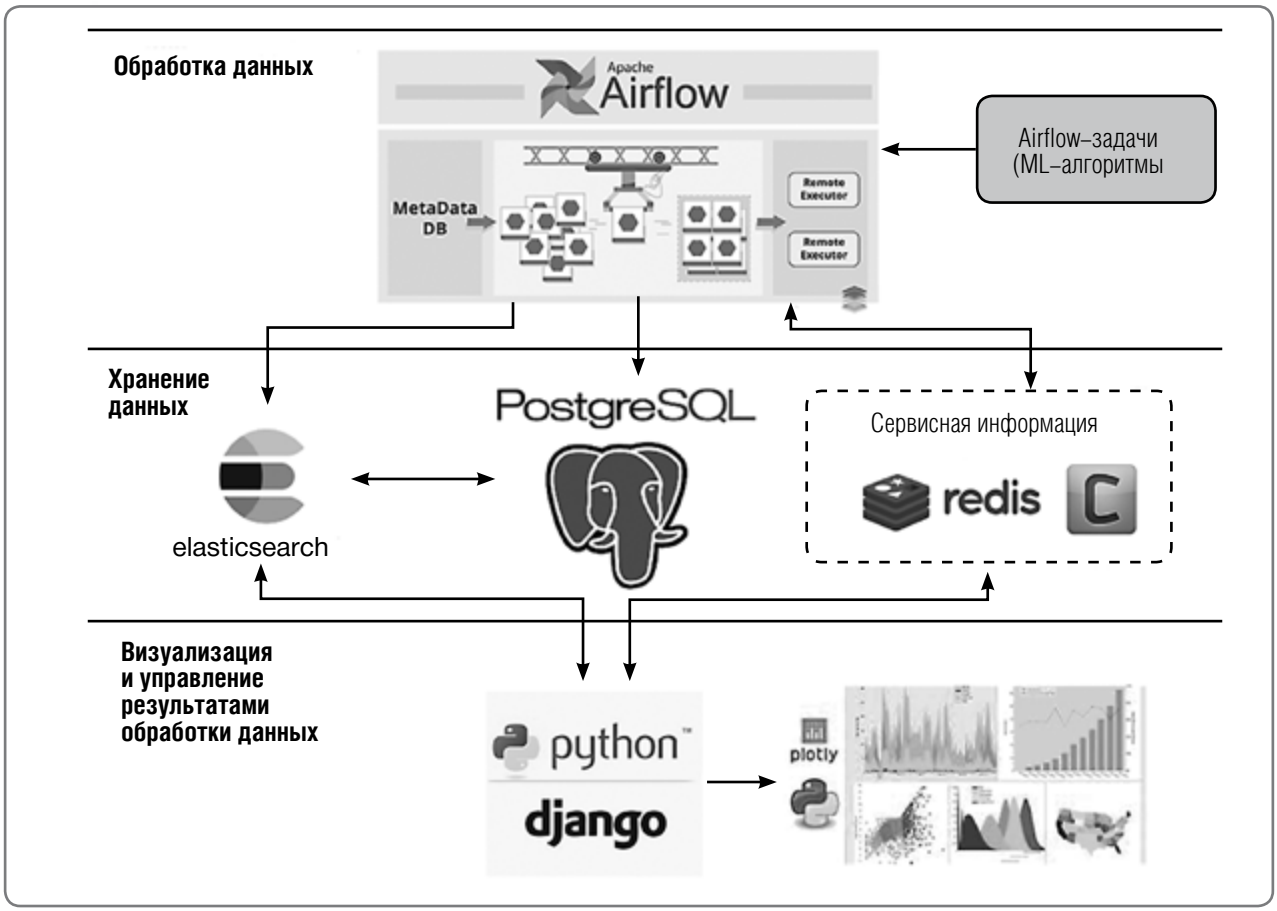


Рис. 1. Структура системы

загружаются в подсистему обработки данных с помощью специального парсера: на данном этапе загрузка происходит вручную по запросу пользователя, в будущем получение новых новостей будет настроено по расписанию. С заданной периодичностью будет выполняться генерация отчетов, требующих большого вычислительного времени; результаты будут размещены в хранилище (такой подход уменьшит время ожидания результатов от подсистемы обработки данных). На основе собранных данных будет выполняться дополнительное обучение модели (1–2 раза в месяц), которое будет включать в себя пересчет набора ключевых характеристик текстового корпуса. В случае, если дообучение модели не будет приводить к увеличению показателя точности (например, в задаче определения тональности текста), предусмотрено использование других ML-алгоритмов или их совокупности.

Ролевая система включает в себя следующие роли:

1. Обычный пользователь – имеет доступ к базовой функциональности системы: поиск, фильтрация, цифровые информационные панели (так называемые дашборды);

2. Расширенный пользователь – имеет доступ к настраиваемым отчетам, автоматическим оповещениям о «горячих темах», возможность проводить фильтрацию по именованным сущностям (например, человек, организация, регион) в статьях. Такое разделение пользователей обусловлено последующим использованием системы государственными органами;

3. Разработчик – имеет доступ к панели администратора Airflow и к репозиторию, в котором хранятся Airflow DAG. Может добавлять и менять свои задачи, запускать и отслеживать их выполнение;

4. Администратор – супер-пользователь, имеет полный набор прав по работе с системой.

Распределение доступной функциональности по ролям представлена на *рисунке 2.*

В последующих подразделах более подробно описан выбранный инструментарий для каждой из перечисленных подсистем.

2.1. Подсистема обработки данных

В ходе анализа для удовлетворения перечисленных потребностей была выбрана программная платформа с открытым исходным кодом Apache Airflow. Основные компоненты данной платформы:

1. Airflow-worker – основной компонент, выполняющий обработку данных. Может быть горизонтально масштабирован, в том числе на отдельные сервера или облачные виртуальные машины. В

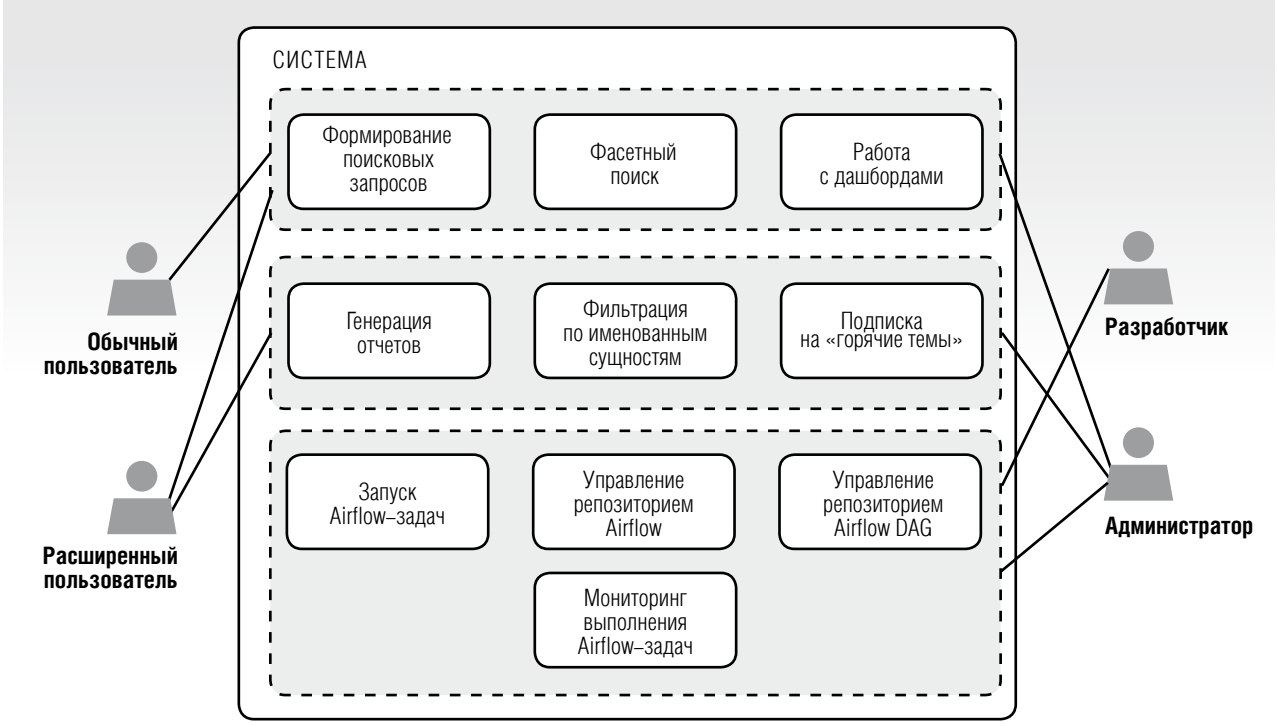


Рис. 2. Распределение доступа к системе по ролям

текущем варианте архитектуры в образ контейнера Airflow-worker заранее встраиваются необходимые зависимости. Однако принципиально процесс инъекции зависимостей может происходить различным образом, в том числе, путем динамичного получения Docker-контейнеров из публичных или приватных репозиториев;

2. Airflow-scheduler – компонент, отвечающий за назначение задач Airflow-workeram в порядке, определенном Airflow DAG-ами. Airflow DAG – нециклический направленный граф, описывающий порядок выполнения определенных задач, а также содержащий информацию о расписании, приоритетах, поведении в случае исключений и т.д.;

3. Airflow web server – веб-интерфейс, позволяющий отслеживать и контролировать ход выполнения задач.

Алгоритмы машинного обучения реализованы в системе как отдельные Airflow-задачи.

2.2. Хранилище

В системе предусмотрено три вида хранилищ:

1. PostgreSQL – выполняет роль персистентного хранилища для структурированных данных. Его использование обусловлено широкими возможностями данной реляционной базы данных (среди свободно распространяемых продуктов) и взаимодействием с широким кругом инструментов. Основные типы данных, хранящиеся в этой базе:

✦ новости и метаданные;

✦ обработанные данные на уровне разных базовых единиц анализа (токен / слово / фраза / предложение / текст), в том числе векторизации, результаты лемматизации, очистки и др.;

✦ результаты проведения тематического моделирования;

✦ результаты классификации новостей по различным признакам (тональность, политизированность, социальная значимость и др.)

2. ElasticSearch – in-memory NoSQL хранилище, предназначенное для хранения неструктурированных или слабоструктурированных данных, а также быстрого поиска (в том числе полнотекстового) и фильтрации и потокового доступа. В сравнении с другими NoSQL базами для хранения документов с произвольной структурой, такими как MongoDB и CouchDB, ElasticSearch, выделяется расширенными инструментами для индексирования текста, позволяющими проводить полнотекстовый поиск по большим объемам документов практически в реальном времени. Также ввиду возможности построения продвинутых индексов для данных, возможно выполнение сложных агрегаций в самой базе, в том числе распределенно. ElasticSearch выполняет несколько функций:

✦ основное хранилище для доступа, поиска и фильтрации данных конечным пользователем;

основное хранилище для ETL (extract, transform. Load) процессов обработки данных, в том числе запись любых промежуточных результатов в свободной форме;

✦ хранилище для кэширования определенных результатов вычислений, необходимых для построения дашбордов и отчетов в системе;

ElasticSearch дублирует данные, хранящиеся в PostgreSQL как персистентном хранилище, поскольку ElasticSearch является in-memory базой данных без гарантий относительно персистентности и целостности данных.

3. Redis – быстрое key-value хранилище, используемое для кэширования отдельных страниц и элементов, а также для кэширования сессий авторизации. В Redis хранятся служебные данные, а также кэш страниц и элементов, к которым происходит частый доступ.

Все три основных хранилища системы могут быть легко масштабированы на несколько отдельных компьютеров. Поддерживается как горизонтальное масштабирование, так и репликация, при этом ElasticSearch и Redis показывают близкое к линейному увеличение производительности при горизонтальном масштабировании.

Для хранения служебных данных, таких как состояния выполнения задач, используется отдельный кластер PostgreSQL. Для запуска и отслеживания прогресса задач используется связка Celery+Redis.

2.3. Подсистема построения аналитических отчетов

Интерфейс подсистемы представляет их себя HTML+CSS+JS веб-сайт с доступом по протоколу HTTP. Выбор стека технологий HTML+CSS+JS для интерфейса оправдан тем, что именно веб-интерфейсы являются наиболее распространенной и повсеместно поддерживаемой технологией построения интерфейсов пользователя, с возможностью доступа с любых устройств и операционных систем из любой точки мира, при условии наличия веб-браузера и подключения к сети интернет.

Веб-приложение реализовано на Python фреймворке Django, в качестве веб-сервера выступает Gunicorn, реверс-прокси – Nginx. Веб-приложение

имеет доступ как к персистентному хранилищу PostgreSQL, так и к ElasticSearch. В Django есть встроенный Cache Framework, который позволяет кэшировать страницы и элементы страниц в Redis. Например, если предполагается, что на страницу будут заходить часто, а считается она долго (например, три секунды), то такую страницу лучше кешировать в Redis, что позволит ускорить доступ к необходимым данным.

Фреймворк Django был выбран по следующим причинам:

1. Возможность быстрой Agile-разработки веб-интерфейса и модели хранения данных. Скорость разработки с применением Django значительно выше, чем при использовании таких аналогов, как Spring (Java), Yii (PHP) и Node.js (JavaScript);

2. Ввиду того, что проект предполагает проведение анализа данных и построение моделей машинного обучения, в том числе для обработки естественных языков (natural language processing, NLP), язык Python является оптимальным выбором, так как большая часть “state-of-the-art” моделей и методов ML/AI и NLP разрабатывается сообществом именно на языке Python;

3. Django ORM лучше работает с базой данных PostgreSQL.

Веб-приложение реализует ряд страниц для фильтрации, поиска и доступа к различным дашбордам и отчетам. На первом этапе реализации системы дашборды считаются заранее вручную. При дальнейшем развитии системы будет использоваться фасетный поиск (faceted search) из Elastic Search. Для формирования графиков будет использоваться Python-библиотека визуализации данных Plotly.

Примерами информации, которую могут отображать графики, являются:

Динамика по тональности (а также манипулятивность, политизированность и др.), тематикам, количеству просмотров и комментариев с фильтрацией по СМИ, тематикам, авторам, тегам (включая полнотекстовый поиск);

Распределение тематик, значений тональности и др. в статике, с фильтрациями и поиском;

Выявление выбросов (аномалий) для аналитических отчетов (самые «горячие» темы и др.).

Заключение

В статье сформулированы требования к структуре программной системы, предназначенной для обработки больших (объемом более 1 млн единиц) корпусов текстовых документов, включая, в частности, реализацию автоматизированной обработки корпуса текстов, возможность распараллеливания вычислений и составление аналитических отчетов. Также определены ролевые функции пользователей. На этой основе разработана структура программной системы, включающая подсистему обработки данных на основе сервиса Apache Airflow, несколько видов хранилищ, обеспечивающих быстрый доступ к компонентам системы, и подсистему построения аналитических отчетов, которая формируется в приложении Python Django с использованием библиотеки визуализации Plotly. Гибкость системы позволяет подбирать разную совокупность алгоритмов машинного обучения, обеспечивая прирост качества и точности анализа корпусов текстовых документов.

В настоящее время система используется для анализа корпусов новостных текстов с целью сравнительного анализа новостных корпусов СМИ Казахстана. ■

Благодарности

Работа финансировалась грантами BR05236839 Министерства образования и науки Республики Казахстан, при финансовой поддержке Российского фонда фундаментальных исследований в рамках научного проекта № 19-31-27001 и в рамках темы государственного задания № АААА-А17-117120670141-7 (№ 0316-2018-0009).

Литература

1. Барахнин В.Б., Кучин Я.И., Мухамедиев Р.И. К вопросу о постановке задачи выявления фейковых новостей и алгоритмах их мониторинга // Материалы III Международной научной конференции «Информатика и прикладная математика». Алматы, 26–29 сентября 2018 г. С.113–118.

2. Шокин Ю.И., Федотов А.М., Барахнин В.Б. Технология создания программных систем информационного обеспечения научной деятельности, работающих со слабоструктурированными документами // Вычислительные технологии. 2010. Т.15. № 6. С. 111–125.

3. Барахнин В.Б., Кожемякина О.Ю., Борзилова Ю.С. Проектирование информационной системы представления результатов комплексного анализа поэтических текстов // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 1. С. 5–17. DOI: 10.25205/1818-7900-2019-17-1-5-17.

4. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика / Е.И. Большакова и [др.]. М.: МИЭМ, 2011.

5. Pang B., Lee L., Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques // Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002). Philadelphia, PA, USA, 6–7 July 2002. P. 79–86. DOI: 10.3115/1118693.1118704.

6. Choi Y., Cardie Cl., Riloff E., Patwardhan S. Identifying sources of opinions with conditional random fields and extraction patterns // Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005). Vancouver, British Columbia, Canada, 6–8 October 2005. P. 355–362.

7. Manning C.D. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? // Proceedings of the 12th International Conference "Computational Linguistics and Intelligent Text Processing" (CICLing 2011). Tokyo, Japan, 20–26 February 2011. P. 171–189.

8. Mukhamediev R., et al. Assessment of the dynamics of publication activity in the field of natural language processing and deep learning // Proceedings of the 4th International Conference on Digital Transformation and Global Society. St. Petersburg, 19–21 June 2019. Springer, 2020 (in press).

9. Tarasov D.S. Deep recurrent neural networks for multiple language aspect-based sentiment analysis // Computational Linguistics and Intellectual Technologies: Proceedings of Annual International Conference "Dialogue–2015". 2015. No 14 (21). Vol. 2. P. 65–74.

10. Garcia-Moya L., Anaya-Sanchez H., Berlanga-Llavori R. Retrieving product features and opinions from customer reviews // IEEE Intelligent Systems. 2013. Vol. 28. No 3. P. 19–27. DOI: 10.1109/MIS.2013.37.

11. Mavljutov R.R., Ostapuk N.A. Using basic syntactic relations for sentiment analysis // Proceedings of the International Conference "Dialogue 2013". Bekasovo, Russia, 29 May – 2 June 2013. P. 101–110.

12. Prabowo R., Thelwall M. Sentiment analysis: A combined approach // Journal of Informetrics. 2009. Vol. 3, No 2. P. 143–157. DOI: 10.1016/j.joi.2009.01.003.

13. Dai W., Xue G.-R., Yang Q., Yu Y. Transferring naive Bayes classifiers for text classification // Proceedings of the 22nd National Conference on Artificial intelligence (AAAI 07). Vancouver, British Columbia, Canada, 26–27 July 2007. Vol. 1. P. 540–545.

14. Cortes C., Vapnik V. Support-vector networks // Machine Learning. 1995. Vol. 20. No 3. P. 273–297. DOI: 10.1023/A:1022627411411.

15. Friedman J.H. Greedy function approximation: a gradient boosting machine // Annals of Statistics. 2001. Vol. 29. No 5. P. 1189–1232.

16. Zhang G.P. Neural networks for classification: A survey // IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews). 2000. Vol. 30. No 4. P. 451–462.

17. Schmidhuber J. Deep learning in neural networks: An overview // Neural Networks. 2015. No 61. P. 85–117. DOI: 10.1016/j.neunet.2014.09.003.

18. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding // arXiv:1810.04805. 2018.

19. Vladimirova T.N., Vinogradova M.V., Vlasov A.I., Shatsky A.A. Assessment of news items objectivity in mass media of countries with intelligence systems: The Brexit case // Media Watch. 2019. Vol. 10. No 3. P. 471–483. DOI: 10.15655/mw/2019/v10i3/49680.

20. Романов А.С., Васильева М.И., Куртукова А.В., Мещеряков Р.В. Анализ тональности текста с использованием методов машинного обучения // 2nd International Conference "R. Piotrowski's Readings in Language Engineering and Applied Linguistics (Saint-Petersburg, 2017). 2018. С. 86–95.

21. Methods to identify the destructive information / V.B. Barakhnin [et al.] // Journal of Physics: Conference Series. 2019. Vol. 1405. No 1. DOI: 10.1088/1742-6596/1405/1/012004.

22. Barakhnin V.B., Kozhemyakina O.Y., Zabaykin A.V. The algorithms of complex analysis of Russian poetic texts for the purpose of automation of the process of creation of metric reference books and concordances // CEUR Workshop Proceedings. 2014. Vol. 1536. P. 138–143.

Об авторах

Барахнин Владимир Борисович

доктор технических наук, доцент;

ведущий научный сотрудник, Институт вычислительных технологий, Сибирское отделение Российской академии наук, 630090, г. Новосибирск, пр-т Академика Лаврентьева, д. 6;

профессор факультета информационных технологий, Новосибирский национальный исследовательский государственный университет, 630090, г. Новосибирск, ул. Пирогова, д. 1;

E-mail: bar@ict.nsc.ru

ORCID: 0000-0003-3299-0507

Кожемякина Ольга Юрьевна

кандидат филологических наук;

старший научный сотрудник, Институт вычислительных технологий, Сибирское отделение Российской академии наук, 630090, г. Новосибирск, пр-т Академика Лаврентьева, д. 6;

E-mail: olgakozhemyakina@mail.ru

ORCID: 0000-0003-3619-1120

Мухамедиев Равиль Ильгизович
доктор инженерных наук;
профессор, Satbayev University, Казахстан, 050013, г. Алматы, ул. Сатпаева, д. 22а;
ведущий научный сотрудник, Институт информационных и вычислительных технологий, Казахстан, 050010, г. Алматы, ул. Пушкина, д. 125;
профессор, Университет ISMA, Латвия, LV-1019, г. Рига, ул. Ломоносова, 1;
E-mail: ravil.muhamedyev@gmail.com
ORCID: 0000-0002-3727-043X

Борзилова Юлия Сергеевна
аспирант, Институт вычислительных технологий, Сибирское отделение Российской академии наук, 630090, г. Новосибирск, пр-т Академика Лаврентьева, д. 6;
E-mail: i.borzilova@alumni.nsu.ru
ORCID: 0000-0002-8265-9356

Якунин Кирилл Олегович
аспирант, Satbayev University, Казахстан, 050013, г. Алматы, ул. Сатпаева, д. 22а;
инженер-программист, Институт информационных и вычислительных технологий, Казахстан, 050010, г. Алматы, ул. Пушкина, д. 125;
E-mail: yakunin.k@mail.ru
ORCID: 0000-0002-7378-9212

The design of the structure of the software system for processing text document corpus

Vladimir B. Barakhnin [a,b]
E-mail: bar@ict.nsc.ru

Olga Yu. Kozhemyakina [a]
E-mail: olgakozhemyakina@mail.ru

Ravil I. Mukhamediev [c,d,e]
E-mail: ravil.muhamedyev@gmail.com

Yulia S. Borzilova [a]
E-mail: i.borzilova@alumni.nsu.ru

Kirill O. Yakunin [c,d]
E-mail: yakunin.k@mail.ru

[a] Institute of Computational Technologies, Siberian Branch of the Russian Academy of Sciences
Address: 6, Academician M.A. Lavrentiev Avenue, Novosibirsk 630090, Russia

[b] Novosibirsk State University
Address: 1, Pirogova Street, Novosibirsk 630090, Russia

[c] Satbayev University
Address: 22a, Satbayev Street, Almaty 050013, Kazakhstan

[d] Institute of Information and Computational Technologies
Address: 125, Pushkin Street, Almaty 050010, Kazakhstan

[e] ISMA University
Address: 1, Lomonosova Street, Riga LV-1019, Latvia

Abstract

One of the most difficult tasks in the field of data mining is the development of universal tools for the analysis of texts written in the literary and business styles. A popular path in the development of algorithms for processing text document corpus is the use of machine learning methods that allow one to solve NLP (natural language processing) tasks. The basis for research in the field of natural language processing is to be found in the following factors: the specificity of the structure of literary and business style texts (all of which requires the formation of separate datasets and, in the case of machine learning methods, the additional feature selection) and the lack of complete systems of mass processing of text documents for the Russian language (in relation to the scientific community-in the commercial environment, there are some systems of smaller scale, which are solving highly specialized tasks, for example, the definition of the tonality of the text). The aim of the current study is to design and further develop the structure of a text document corpus processing system. The design took into account the requirements for large-scale systems: modularity, the ability to scale components, the conditional independence of components. The system we designed is a set of components, each of which is formed and used in the form of Docker-containers. The levels of the system are: the data processing level, the data storage level, the visualization and management of the results of data processing (visualization and management level). At the data processing level, the text documents (for example, news events) are collected (scrapped) and further processed using an ensemble of machine learning methods, each of which is implemented in the system as a separate Airflow-task. The results are placed for storage in a relational database; ElasticSearch is used to increase the speed of data search (more than 1 million units). The visualization of statistics which is obtained as a result of the algorithms is carried out using the Plotly plugin. The administration and the viewing of processed texts are available through a web-interface using the Django framework. The general scheme of the interaction of components is organized on the principle of ETL (extract, transform, load). Currently the system is used to analyze the corpus of news texts in order to identify information of a destructive nature. In the future, we expect to improve the system and to publish the components in the open repository GitHub for access by the scientific community.

Key words: natural language processing; streaming word processing; text analysis information system; development of a text corpus processing system.

Citation: Barakhnin V.B., Kozhemyakina O.Yu., Mukhamediev R.I., Borzilova Yu.S., Yakunin K.O. (2019) The design of the structure of the software system for processing text document corpus. *Business Informatics*, vol. 13, no 4, pp. 60–72. DOI: 10.17323/1998-0663.2019.4.60.72

References

1. Barakhnin V.B., Kuchin Ya.I., Mukhamediev R.I. (2018). On the problem of identification of fake news and of the algorithms for monitoring them. Proceedings of the *III International Conference on Informatics and Applied Mathematics, Almaty, Kazakhstan, 26–29 September 2018*, pp.113–118 (in Russian).
2. Shokin Yu.I., Fedotov A.M., Barakhnin V.B. (2010) Technologies for construction of processing software systems dealing with semistructured documents aimed at information support of scientific activity. *Computational Technologies*, vol. 15, no 6, pp. 111–125 (in Russian).
3. Barakhnin V.B., Kozhemyakina O.Yu., Borzilova Yu.S. (2019) The development of the information system of the representation of the complex analysis results for the poetic texts. *Vestnik NSU. Series: Information Technologies*, vol. 17, no 1, pp. 5–17 (in Russian). DOI: 10.25205/1818-7900-2019-17-1-5-17.
4. Bolshakova E.I., Klishinskii E.S., Lande D.V., Noskov A.A., Peskova O.V., Yagunova E.V. (2011) *Automatic natural language text processing and computer linguistics*. Moscow: MIEM (in Russian).
5. Pang B., Lee L., Vaithyanathan S. (2002) Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the *2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), Philadelphia, PA, USA, 6–7 July 2002*, pp. 79–86. DOI: 10.3115/1118693.1118704.
6. Choi Y., Cardie Cl., Riloff E., Patwardhan S. (2005) Identifying sources of opinions with conditional random fields and extraction patterns. Proceedings of the *Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005). Vancouver, British Columbia, Canada, 6–8 October 2005*, pp. 355–362.
7. Manning C.D. (2011) Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? Proceedings of the *12th International Conference "Computational Linguistics and Intelligent Text Processing" (CICLing 2011), Tokyo, Japan, 20–26 February 2011*, pp. 171–189.
8. Mukhamediev R., et al. (2020) Assessment of the dynamics of publication activity in the field of natural language processing and deep learning. Proceedings of the *4th International Conference on Digital Transformation and Global Society, St. Petersburg, Russia, 19–21 June 2019*. Springer, 2020 (in press).
9. Tarasov D.S. (2015) Deep recurrent neural networks for multiple language aspect-based sentiment analysis. *Computational Linguistics and Intellectual Technologies: Proceedings of Annual International Conference "Dialogue–2015"*, no 14 (21), vol. 2, pp. 65–74.
10. Garcia-Moya L., Anaya-Sanchez H., Berlanga-Llavori R. (2013) Retrieving product features and opinions from customer reviews. *IEEE Intelligent Systems*, vol. 28, no 3, pp. 19–27. DOI: 10.1109/MIS.2013.37.
11. Mavljutov R.R., Ostapuk N.A. (2013) Using basic syntactic relations for sentiment analysis. Proceedings of the *International Conference "Dialogue 2013", Bekasovo, Russia, 29 May – 2 June 2013*, pp. 101–110.
12. Prabowo R., Thelwall M. (2009) Sentiment analysis: A combined approach. *Journal of Informetrics*, vol. 3, no 2, pp. 143–157. DOI: 10.1016/j.joi.2009.01.003.

13. Dai W., Xue G.-R., Yang Q., Yu Y. (2007) Transferring naive Bayes classifiers for text classification. Proceedings of the *22nd National Conference on Artificial intelligence (AAAI 07). Vancouver, British Columbia, Canada, 26–27 July 2007*, vol. 1, pp. 540–545.

14. Cortes C., Vapnik V. (1995) Support-vector networks. *Machine Learning*, vol. 20, no 3, pp. 273–297. DOI: 10.1023/A:1022627411411.

15. Friedman J.H. (2001) Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, vol. 29, no 5, pp. 1189–1232.

16. Zhang G.P. (2000) Neural networks for classification: A survey. *IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews)*, vol. 30, no 4, pp. 451–462.

17. Schmidhuber J. (2015) Deep learning in neural networks: An overview. *Neural Networks*, no 61, pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003.

18. Devlin J., Chang M.-W., Lee K., Toutanova K. (2018) BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

19. Vladimirova T.N., Vinogradova M.V., Vlasov A.I., Shatsky A.A. (2019) Assessment of news items objectivity in mass media of countries with intelligence systems: The Brexit case. *Media Watch*, vol. 10, no 3, pp. 471–483. DOI: 10.15655/mw/2019/v10i3/49680.

20. Romanov A.S., Vasilieva M.I., Kurtukova A.V., Meshcheryakov R.V. (2018) Sentiment analysis of text using machine learning techniques. Proceedings of the *2nd International Conference "R. Piotrowski's Readings in Language Engineering and Applied Linguistics (Saint-Petersburg, 2017)*, pp. 86–95 (in Russian).

21. Barakhnin V.B., Mukhamedyev R.I., Mussabaev R.R., Kozhemyakina O.Yu., Issayeva A., Kuchin Ya.I., Murzakhmetov S.B., Yakunin K.O. (2019) Methods to identify the destructive information. *Journal of Physics: Conference Series*, vol. 1405, no 1. DOI: 10.1088/1742-6596/1405/1/012004.

22. Barakhnin V.B., Kozhemyakina O.Y., Zabaykin A.V. (2014) The algorithms of complex analysis of Russian poetic texts for the purpose of automation of the process of creation of metric reference books and concordances. *CEUR Workshop Proceedings*, vol. 1536, pp. 138–143.

About the authors

Vladimir B. Barakhnin
Dr. Sci. (Tech.), Associate Professor;
Leader Researcher, Institute of Computational Technologies, Siberian Branch of the Russian Academy of Sciences, 6, Academician M.A. Lavrentiev Avenue, Novosibirsk 630090, Russia;
Professor, Faculty of Information Technologies, Novosibirsk State University, 1, Pirogova Street, Novosibirsk 630090, Russia;
E-mail: bar@ict.nsc.ru
ORCID: 0000-0003-3299-0507

Olga Yu. Kozhemyakina
Cand. Sci. (Philol.);
Senior Researcher, Institute of Computational Technologies, Siberian Branch of the Russian Academy of Sciences, 6, Academician M.A. Lavrentiev Avenue, Novosibirsk 630090, Russia;
E-mail: olgakozhemyakina@mail.ru
ORCID: 0000-0003-3619-1120

Ravil I. Mukhamediev
Dr. Sci. (Eng.);
Professor, Satbayev University, 22a, Satbayev Street, Almaty 050013, Kazakhstan;
Leader Researcher, Institute of Information and Computational Technologies, 125, Pushkin Street, Almaty 050010, Kazakhstan;
Professor, ISMA University, 1, Lomonosova Street, Riga LV-1019, Latvia;
E-mail: ravil.muhamedyev@gmail.com
ORCID: 0000-0002-3727-043X

Yulia S. Borzilova
Doctoral Student, Institute of Computational Technologies, Siberian Branch of the Russian Academy of Sciences, 6, Academician M.A. Lavrentiev Avenue, Novosibirsk 630090, Russia;
E-mail: i.borzilova@alumni.nsu.ru
ORCID: 0000-0002-8265-9356

Kirill O. Yakunin
Doctoral Student, Satbayev University, 22a, Satbayev Street, Almaty 050013, Kazakhstan;
Developer Engineer, Institute of Information and Computational Technologies, 125, Pushkin Street, Almaty 050010, Kazakhstan;
E-mail: yakunin.k@mail.ru
ORCID: 0000-0002-7378-9212