Математическая биология и биоинформатика
2022. Т. 17. № 2. С. 423–440. doi: 10.17537/2022.17.423

==========ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ ==========
==========ТЕХНОЛОГИИ В БИОЛОГИИ И МЕДИЦИНЕ==========

Извлечение фактографической информации о пандемии из открытых источников сети Интернет

Акулинина Е.Ю.[*1], Карманов А.Л.[†1], Теплых Н.А.[‡1], Власов В.В.[§1], Балута В.И.[2], Варыханов С.С.[††2], Карандеев А.А.[‡‡2], Осипов В.П.[§§2], Рыков Ю.Г.[***2], Четверушкин Б.Н.[†††2]**

[1]*ФГУП «РФЯЦ ВНИИТФ им. академ. Е.И. Забабахина, Снежинск, Россия*
[2]*Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия*

Аннотация. Создание базирующихся на мультиагентных подходах моделей распространения инфекционных заболеваний основывается на использовании большого объема разнородных исходных данных, как правило, отсутствующих в непосредственном доступе, в связи с чем одной из ключевых проблем конструирования таких моделей является разработка инструментов получения данных из различных источников. В настоящей статье представлены подходы, позволяющие извлекать из текстовых сообщений, опубликованных в сети Интернет, значения параметров функционирования моделируемого общества и статистические данные о процессе развития пандемии. Предложены метод и программная реализация для целенаправленного поиска открытых источников информации в сети интернет и обработки неструктурированных данных. Собранные таким образом данные используются для настройки математической модели при исследовании различных сценариев развития эпидемии в конкретных регионах. Акцент в предлагаемом подходе обработки данных сделан на двух основных технологиях: применение регулярных выражений и анализ с использованием методов машинного обучения. Использование метода регулярных выражений позволяет обеспечить высокую скорость обработки текстов, но его применимость ограничивается сильной зависимостью от контекста. В свою очередь, машинное обучение позволяет адаптироваться под информационный контекст сообщения, однако при этом наблюдаются относительно большие затраты времени на анализ. Для повышения точности анализа и нивелирования недостатков каждого из этих подходов предлагаются способы совмещения названных технологий. В статье излагаются полученные результаты оптимизации алгоритмов получения необходимых данных. Реализация предлагаемых решений выполнена на языках Python и C++ с использованием библиотек по обработке русскоязычной текстовой информации. Также представлено решение на основе современной программной платформы для автоматизации процесса мониторинга и обработки выбранных информационных каналов.

[*]e.yu.akulinina@vniitf.ru
[†]a.l.karmanov@vniitf.ru
[‡]teplykhna@vniitf.ru
[§]v.v.vlasov@vniitf.ru
[**]vbaluta@keldysh.ru
[††]masmx64@gmail.com
[‡‡]KarAlex755@gmail.com
[§§]osipov@keldysh.ru
[***] yu.rykov@keldysh.ru
[†††]office@keldysh.ru

Ключевые слова: *анализ текстовых данных, регулярные выражения, синтаксические деревья, платформа сбора данных.*

ВВЕДЕНИЕ

Традиционно используемые в эпидемиологии различные версии компартментальных моделей базируются на предположении взаимодействия «всех со всеми» и оперируют интегральными параметрами. Потенциально такой подход не позволяет получать детализированные результаты и проводить сравнительный анализ значимости различных факторов в распространении эпидемии, оценивать ожидаемую эффективность мер вмешательства и исследовать множество других аспектов в процессе развития эпидемии. В связи с чем внимание исследователей обращено на создание агентных моделей [1], которые ориентированы на персонализацию взаимодействий и лишены указанных недостатков. К сожалению, детализированный подход нуждается и в существенной детализации исходных данных для достоверного моделирования, что само по себе представляет проблему.

В отечественной практике отсутствует значимый срез актуальных данных об отдельных аспектах социальных процессов в обществе, по крайней мере, в удобном концентрированном виде для непосредственного применения в агент-ориентированных моделях. Значительная часть из имеющихся в наличии данных, которые могли бы быть использованы, как правило, является труднодоступной либо по причине закрытости первичных источников, либо ввиду высокой стоимости их получения [2]. В то же время можно ожидать, что значительное количество требующихся данных рассеяно во множестве альтернативных источников, находящихся в открытом доступе сети интернет [3]. Именно их рассеянность и отсутствие предварительных сведений об их наличии создают огромные трудности для анализа всей имеющейся информации. В ряду сложностей можно отметить и тот факт, что наиболее распространенной формой представления информации в сети Интернет является текстовое описание, когда данные не имеют унифицированной структуры. При этом поле текстовых источников расширяется и за счет обработки аудио-файлов путем преобразования аудиозаписей речи в текстовый формат.

Наиболее перспективными с точки зрения поиска и мониторинга динамики актуальных данных о социальных процессах из числа общедоступных открытых источников в сети Интернет являются RSS–ленты новостных изданий, социальные сети, сайты администраций муниципалитетов и государственных учреждений. RSS (англ. Rich Site Summary – обогащенная сводка сайта) – это семейство XML–форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и другие. Информация из различных источников, представленная в формате RSS, может быть собрана, обработана и представлена пользователю в удобном для него виде специальными программами-агрегаторами или онлайн-сервисами, такими как NewsAlloy, FeedBucket и другими аналогичными по функциональности [4]. Информация на таких ресурсах обычно хранится в неструктурированном виде, что существенно усложняет процесс автоматического извлечения данных. Под термином «неструктурированная информация» или «неструктурированные данные» (англ. unstructured information or unstructured data), как правило, понимается любая информация, которая не имеет предопределённой формальной модели данных или не укладывается в таблицу реляционной базы данных [5].

Дополнительно для поиска источников информации предлагается использовать общедоступные поисковые системы. В рамках данного проекта используются платформы от Google и Yandex, поскольку они хорошо адаптированы к русскому языку и индексируют огромное количество интернет-ресурсов.

Автоматическая обработка и анализ разнородной информации, представленной на естественном языке, является одним из самых востребованных на сегодняшний день

направлений исследований. В области анализа текста сформировалось направление, обозначаемое термином Information Extraction (извлечение информации). Отличительной особенностью систем данного класса является выборочный анализ только релевантных фрагментов текста [6].

Получение из текстов фактографических данных связано с анализом текста для выявления в нем информации определенного вида и представления ее структурированным образом, удобным для последующей обработки [7].

Сложность обработки текстовой информации состоит в том, что в зависимости от предметной области возникают особенности ее структурирования, которые упрощают анализ в каждом отдельном случае, но в общем случае этот процесс не поддается унификации. Например, обработка табличных данных позволяет использовать типовую структуру таблиц для автоматизации анализа [8]. Анализ событий с указанием времени их возникновения также имеет ряд особенностей, которые могут быть использованы при обработке текстовой информации, содержащей такие данные [9]. Для повышения качества распознавания в задаче извлечения геоданных из текстов используются заранее заданные категории, например, названия организаций, геообъектов, различные топонимы [10].

На сегодня уже создано большое количество методов обработки текстовых данных, которые решают целый ряд базовых задач анализа текста, представленного на естественном языке [11]. Выделяют три основных направления в задаче извлечения информации: статические методы, методы на основе правил и методы на основе машинного обучения [12, 13].

Статические методы – наиболее простые, которые справляются с задачей извлечения информации, когда исходный текст представлен в структурированном виде. Это узкоспециализированные методы, которые не позволяют обрабатывать тексты, не имеющие жестко заданной структуры [12]. Однако такие методы позволяют обеспечить высокую скорость поточной обработки текстовой информации. Например, регулярные выражения позволяют обеспечить выбор необходимой информации путем поиска в строке подстроки, которая подходит под описание, заданное с помощью специального языка [14]. Регулярные выражения строятся на основе композиции метасимволов. Каждый метасимвол имеет специальное назначение: набор символов, последовательность этих символов, количество повторений и так далее.

Альтернативный подход представляют методы, основанные на правилах, которые позволяют обрабатывать тексты, не имеющие жесткой структуры. Описание текстовых ситуаций выполняется при помощи правил на специальном языке. Обычно правило состоит из двух частей: образец, которому должна соответствовать речевая ситуация, и действие, которое правило должно совершить с данной ситуацией. Такой подход подразумевает разработку правил лингвистами и специалистами в конкретной предметной области. Правила в результате должны быть достаточно общие, чтобы находить схожие речевые ситуации, но и достаточно специализированные, чтобы исключать ложные срабатывания [12].

Подход, основанный на явном представлении знаний, предполагает привлечение различных лингвистических ресурсов и написание правил, которые отражают структуру предметной области и определяют релевантность фрагментов текста относительно искомой информации. Развитием этого подхода стало представление знаний предметной области в виде формальной модели (онтологии) [6]. Концепция семантической сети предлагает вместе с текстовой информацией хранить структурированные знания, описанные с помощью специальных языков [15].

Подход, основанный на машинном обучении, предполагает автоматическое построение образцов с использованием большого корпуса текстов. Чаще всего используются итерационные алгоритмы типа bootstrapping (дословно - шнурование): на вход системе подается небольшой набор образцов; на основе этих образцов в тексте

находят слова, описывающие значимые объекты и их свойства; затем в текстах ищутся контексты, в которых встречаются данные слова, и используют эти контексты в качестве образцов на следующих итерациях [6].

В настоящей работе рассматриваются сходные методы извлечения данных, частично базирующиеся на разных библиотеках программных продуктов.

Первый предполагает извлечение статистических данных о текущем состоянии пандемической ситуации из новостей, заключающийся в применении комбинации статического подхода на основе регулярных выражений и подхода, основанного на машинном обучении. В этом методе для анализа текстовой информации применен набор библиотек Natasha [16], написанных на языке Python, который широко используется в решении частных задач обработки естественного русского языка, включая сегментацию текста на токены и предложения, морфологический и синтаксический анализ, лемматизацию [17].

Аналогом упомянутой библиотеки Natasha является также представленная в базе языка Python в виде отдельного инструментария для разбора, в том числе, предложений русского языка библиотека MaltParser [18]. В основе лежит нейросеть, построенная на основе библиотек популярного фреймворка Tensorflow, которая предназначена для выстраивания и работы с деревьями зависимостей. Данная библиотека позволяет выстраивать модели взаимодействия слов по размеченному корпусу и строить деревья для новых данных, основываясь на уже созданных моделях. Такая архитектура подразумевает реализацию нескольких алгоритмов построения синтаксических деревьев.

Библиотека MaltParser хорошо показала себя в сравнении с другими нейронными сетями аналогичного типа, а также, по утверждениям авторов, достигла передовых показателей в корректном распознавании синтаксических связей в североевропейских языковых группах. К последней версии библиотеки также приложены готовые модели для английского, французского, шведского и испанского языков. Помимо этого, существуют модели для русского языка, натренированные на первом корпусе русских текстов с синтаксической разметкой SynTagRus [19].

ВИДЫ ИНФОРМАЦИИ

В процессе моделирования для получения прогнозов требуются исторические статистические данные о характере протекания эпидемии в субъекте моделирования (например, число новых заражений, число госпитализированных, число смертей, число выздоровевших). Для получения высокоточных прогнозов уровень детализации данных является ключевым. На данный момент в федеральных источниках информации на регулярной основе публикуются сведения только о развитии эпидемии в регионах в целом. Региональные представительства публикуют более детальные сведения, но форматы их представления не стандартизированы и могут меняться во времени.

Всю публикуемую в открытых источниках информацию можно условно разделить на три класса: структурированная, слабоструктурированная и неструктурированная. Структурированная информация представлена в формате, удобном для машинной обработки, например, в табличном виде. Слабоструктурированная информация имеет повторяющиеся паттерны, которые могут частично или полностью меняться во времени. Неструктурированная информация чаще всего может быть представлена в виде текстов на естественном языке.

Можно выделить несколько основных способов сбора данных из сети Интернет:

(1) с помощью прикладного программного интерфейса, предоставляемого непосредственно поставщиком данных;

(2) с помощью семантического разбора веб-страниц;

Математическая биология и биоинформатика. 2022. Т. 17. № 2. doi: 10.17537/2022.17.423

(3) с помощью средств эмуляции действий пользователя в браузере при поиске информации.

В случае отсутствия API для доступа к данным на внешних интернет-ресурсах задача сбора существенно усложняется и требует применения специальных методов анализа текстовых данных (табл. 1).

Таблица 1. Набор специальных символов, используемых в предлагаемом решении

Символ	Описание
.	любой символ, за исключением символа новой строки
^	начало строки
$	конец строки, или символ перед символом новой строки
*	любое количество повторений регулярного выражения перед этим символом, 0 или более раз
+	1 или более повторений регулярного выражения перед этим символом
?	или 0, или 1 повторение регулярного выражения перед этим символом
{m, n}	число повторений регулярного выражения перед этим символом от m до n раз
\	спецсимволы или символ «\»
[]	набор или диапазон символов
\A	поиск совпадений только в начале строки
\Z	поиск совпадений, начиная только с конца строки
\d	любое десятичное число, равносильно [0–9]
\D	любой символ, не являющийся числом, равносильно [^ \d]
\s	любой символ, являющийся разделителем, равносильно [\t \n \r \f \v]
\S	любой символ, не являющийся разделителем слов, равносильно [^ \s]
\w	поиск совпадений с любым буквенным или числовым символом, равносильно [a–zA–Z0–9]

МЕТОДИКА ПОИСКА ИСТОЧНИКОВ НЕСТРУКТУРИРОВАННОЙ ИНФОРМАЦИИ И ЕЁ АНАЛИЗ

Для решения задачи поиска информационных источников и выделения значений искомых параметров из рассеянных в сети Интернет текстов разработан и реализован в программном коде алгоритм Net_Offroad. Поиск параметров в неструктурированных источниках делится на две стадии. Первая стадия – это автоматический поиск потенциальных источников информации с помощью поисковых систем Google и Yandex. Второй – это запрос веб-страниц из найденных источников и их последующий анализ.

Для каждого конкретного параметра формируются запросы для поиска с помощью поисковых систем Google и Yandex, а также список ключевых слов для поиска нужных значений в найденных результатах.

В первую очередь, к поисковой системе осуществляется обращение через API или через стандартную страницу поиска информации пользователем (называемое обычно парсингом поисковой выдачи) с указанием желаемого количества страниц выдачи, или, другими словами, задания глубины поиска. Предопределено, что вне зависимости от применяемого метода (через API или парсинг) на одной странице выводится заданное число результатов. Полученный ответ анализируется и добавляется в базу данных ссылок на найденные страницы. Сами страницы на этой стадии не загружаются. Путем сравнения с ранее записанными в базу адресами проводится выбраковка дублирующихся результатов и выделение новых уникальных ссылок. Загрузка самих страниц для анализа происходит на второй стадии.

Скачанные страницы подвергаются синтаксическому анализу с помощью синтаксического анализатора MaltParser. На его вход подаётся текст за исключением разметки страницы. На выходе получается последовательность предложений с

указанием синтаксического дерева и некоторой другой информации для каждого из них.

Данные, полученные в результате синтаксического анализа, сканируются на наличие ключевых слов и числовых значений, а также взаимных отношений между ними.

Программный код для выполнения описанной задачи состоит из нескольких модулей. Структура их взаимодействия в рамках системы представлена на рисунке 1.

Алгоритм работы системы реализует следующие шаги:

1. Модуль поисковых запросов считывает заданные через файлы конфигурации списки искомых параметров, включая привязку к территориальным зонам, в автоматическом режиме формирует поисковые запросы и отправляет их поисковым системам Google и Yandex. Полученные результаты поиска анализируются и заносятся в базу данных.

Рис. 1. Основные элементы системы поиска информации в сети Интернет.

2. Модуль загрузки страниц подключается к базе данных и выгружает из неё адреса страниц, которые ещё не были загружены, затем производит их загрузку в базу данных. В случае контроля за динамически изменяющимися параметрами содержащие их страницы могут быть загружены повторно по истечении заданного промежутка времени. Это необходимо для ресурсов, которые постоянно обновляют информацию. В случае ошибки загрузки, страницы будут помечены как недоступные и загружены позднее.

3. Модуль анализа текста реализует задачу поиска конкретных числовых значений в тексте на странице. Поскольку информация в тексте является неструктурированной, то с помощью нейронной сети MaltParser для предложений из текста строятся синтаксические деревья, позволяющие установить степень приближенности найденных числовых значений искомым параметрам.

4. После построения синтаксического дерева для каждого предложения необходимо определить, содержит ли данное предложение искомое значение

параметра. Для этого используется поиск по ключевым словам. Во время поиска учитываются различные возможные склонения слов. Конкретные сочетания ключевых слов зависят от выбранного параметра.

Схематическое представление пути, который проходят содержащиеся в информационных источниках данные от формирования поискового запроса до выделения значений искомого параметра, отображено на рисунке 2.

Результаты работы алгоритма приведены на рисунке 3. В публикуемом примере продемонстрированы отдельные результаты последовательного поиска данных по параметру «количество умерших от заражения коронавирусом» в «столице» в «течение суток».

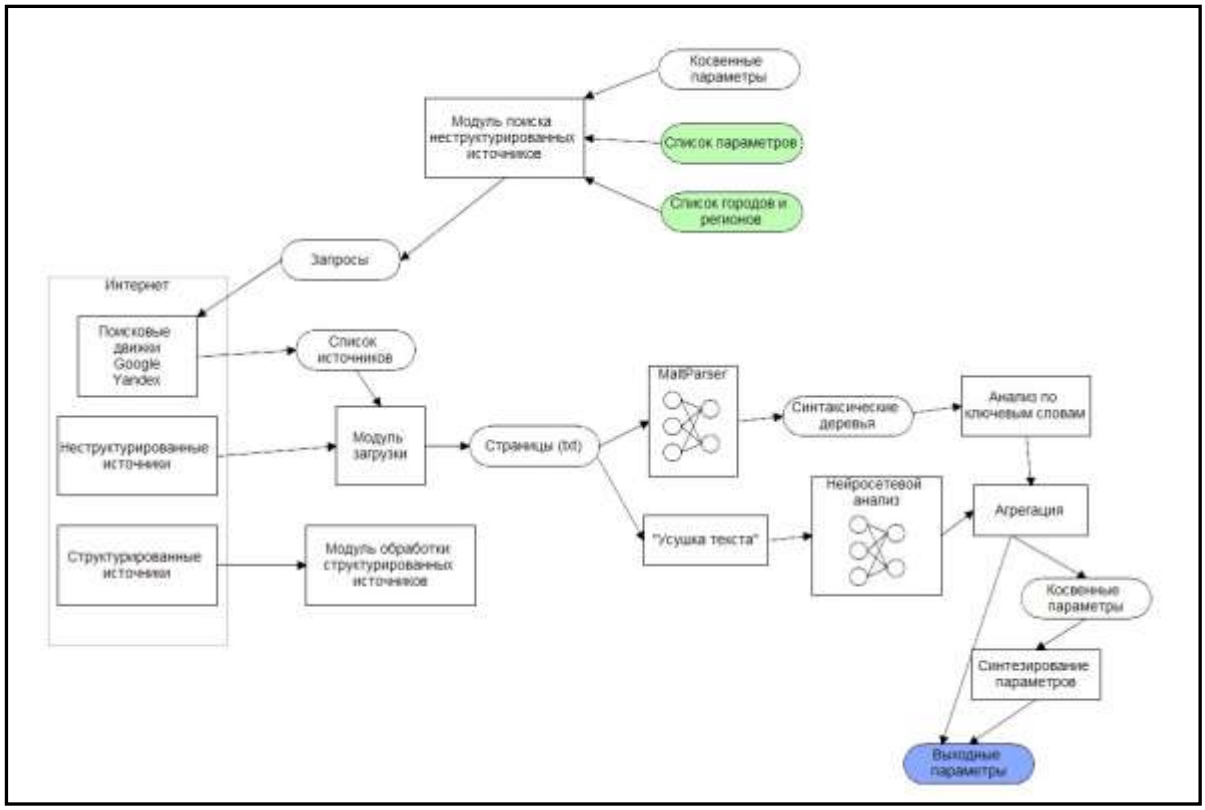

Рис. 2. Диаграмма обработки потока данных.

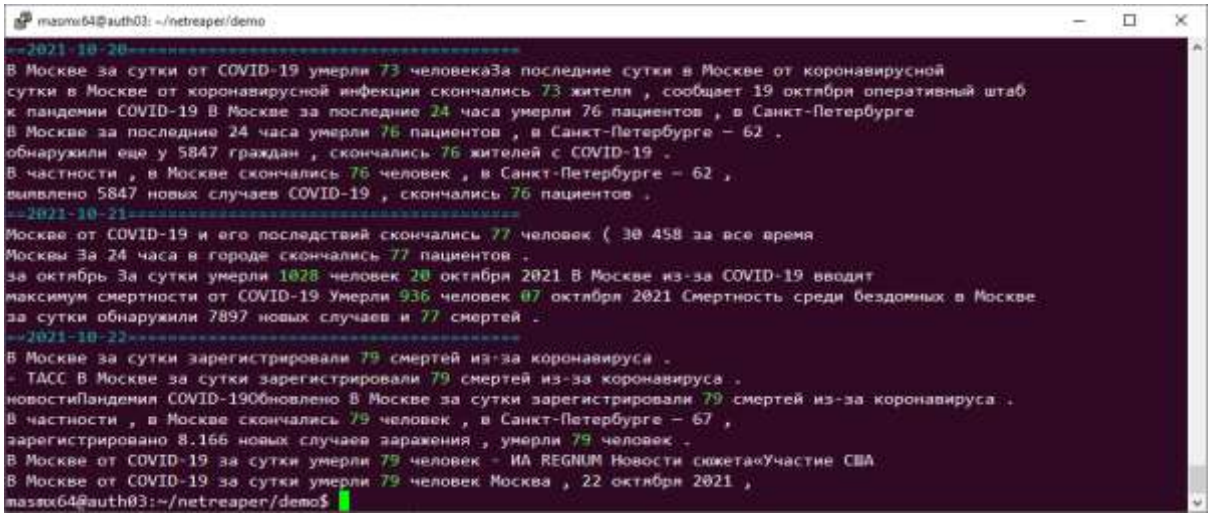

Рис. 3. Вывод результатов работы алгоритма Net_Offroad.

РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ И СИНТАКСИЧЕСКИЕ ДЕРЕВЬЯ

Обработка структурированного текста заключается в извлечении информации на основе известных повторяющихся паттернов. Описание паттернов можно выполнять с помощью регулярных выражений. Регулярные выражения – это шаблон, составленный из спецсимволов, описывающий правила соответствия анализируемых подстрок необходимому виду.

Механизмы поиска совпадений на основе регулярных выражений реализованы в разных языках программирования (C++, C#, Python) и имеют незначительные отличия в основной логике применения. Для примера, в таблице 1 представлен набор спецсимволов, использующийся для построения регулярных выражений на языке Python.

Пример 1. Рассмотрим пример обработки текстовой информации с помощью метода, основанного на регулярных выражениях:

Анализируемый текст:

«В Екатеринбурге 66 заболевших, в Челябинске 74 заболевших, а в Снежинске 15 человек с диагнозом COVID–19»

Регулярное выражение:

\w + \s?\d+

Результат обработки:

[’Екатеринбурге 66’, ’Челябинске 74’, ’Снежинске 15’]

Так же регулярные выражения могут использоваться и для предобработки исходных данных с целью минимизации избыточности и неоднозначности представления текстов. Например, удаление гласных из исходного текста сохраняет смысловую нагрузку, но сокращает объем данных.

Пример 2. В этом примере показан результат удаления всех гласных из текстового сообщения:

Анализируемый текст:

«В Москве число зараженных COVID–19 за сутки впервые за долгое время сократилось до 2355 человек».

Регулярное выражение:

ˆ[АЕИОУЫЭЮЯаеиоуыэюя]+|[АЕИОУЭЮЯаеиоуыэюя]+$|[ˆАЕИОУЭЮЯаеиоуыэюя]

Результат обработки:

«В Мскв чсл зржннх COVID–19 з стк впрв з длг врм скртлсь д 2355 члвк»

Новостные статьи, как правило, имеют неструктурированный вид, в связи с чем возникает проблема лингвистического анализа естественного языка.

Ниже перечислены основные проблемы при анализе текстов и способы их решения:

(1) Флективность – словоизменения (изменения суффиксов и окончаний). Решается с помощью стемминга (получение основы слова путем отбрасывания окончания и приставок) и лемматизации (получение начальной формы слова, с помощью использования знаний о точной части речи слова).

(2) Омонимия – использование одного слова в различных значениях. Различают смысловую («замок двери» и «замок рыцаря») и частиречную («прибрать комнату» и «прибрать к рукам») омонимию. Частиречная неоднозначность снимается с помощью POS–анализа (Part of Speech tagging). Смысловая омонимия разрешается путем оценки сходства по количеству общих смыслов (Word sense disambiguation).

(3) Изменение порядка слов в предложениях (отсутствие в русском языке строгих правил относительно порядка следования слов в предложении). Данная проблема решается с помощью синтаксического разбора и построения синтаксического дерева.

Синтаксическое дерево – это упорядоченное корневое дерево (иерархическая структура с «корневым» значением и поддеревьями, представленное как совокупность

Математическая биология и биоинформатика. 2022. Т. 17. № 2. doi: 10.17537/2022.17.423

соединенных между собой цепочек по принципу подчинения), которое представляет собой отображение синтаксической структуры предложения. На рисунке 4 представлен пример такой иерархической синтаксической структуры следующего предложения: «В Алтайском крае подтверждено еще 3 случая заболевания коронавирусной инфекцией».

Рис. 4. Пример синтаксического дерева.

ПРЕДЛАГАЕМЫЙ ПОДХОД И РЕЗУЛЬТАТЫ ОБРАБОТКИ

Современным подходом решения вышеперечисленных задач является применение алгоритмов обработки текстовых данных [6, 11], часть из которых реализована в проекте Natasha для языка программирования Python. Данная библиотека предназначена для обработки естественного русского языка, и позволяет выполнять:

(1) сегментацию на токены и предложения;

(2) морфологический и синтаксический анализ;

(3) лемматизацию и поиск именованных сущностей (NER, Named-entity recognition);

(4) поиск топонимов.

Модели синтаксического, морфологического и NER-анализа обучены на большом датасете Nerus, содержащем 700 000 размеченных новостных статей. Качество для новостных статей сравнимо или превосходит существующие решения, при большей скорости обработки и меньшем объеме модели.

Другим семейством инструментов обработки текста являются алгоритмы с нечеткой логикой, которые реализуют поиск конкретных числовых значений в тексте на странице из сети Интернет. Информация в тексте является неструктурированной, сам естественный язык слишком сложен. Если выражаться в терминах машинных грамматик, то естественные языки не подходят даже под класс контекстно-свободных грамматик.

Это приводит к тому, что построить синтаксическое дерево и тем более «понять» смысл сказанного можно только алгоритмами с нечёткой логикой, в нашем случае была использована нейронная сеть MaltParser. Она настроена работать с русским языком и выдаёт удовлетворительные результаты.

После построения синтаксического дерева для каждого предложения определяется наличие в нем искомого значения параметра. Для этого используется проверка по ключевым словам. Во время такой проверки учитываются различные возможные склонения слов. Конкретные сочетания ключевых слов зависят от искомого параметра. Причем каждое из ключевых слов может иметь свой вес, чтобы предложения, которые больше соответствуют запросу по наличию ключевых слов, получали и больший приоритет.

Алгоритмом предусмотрены процедуры учета и проверки слов, стоящих непосредственно после выявленных значений, поскольку именно в этом месте текста чаще всего указываются единицы измерений, позволяющие с большой долей вероятности установить или исключить принадлежность величины к искомому параметру. Проводится разбор и сложных единиц, когда некоторые значения указываются, например, в формате с дополнительными уточнениями типа «100 млн. человек». К сожалению, различных вариантов подобных случаев, затрудняющих непосредственную интерпретацию значений, достаточно велико.

Поскольку алгоритм имеет нечёткую логику, то при автоматической обработке возможно извлечение ошибочных величин. Зачастую проблема решается путем отсечения явно ошибочных значений, когда они выходят из пределов разумного диапазона, которые для большинства параметров можно довольно точно оценить вручную.

Проведем анализ фразы: «В конгломерате Москвы и Московской области по данным Росстата проживает 20 363 549 человек (московская агломерация), это около 13.93 % населения России».

В результате работы MaltParser, мы увидим следующее представление древовидной структуры (рис. 5).

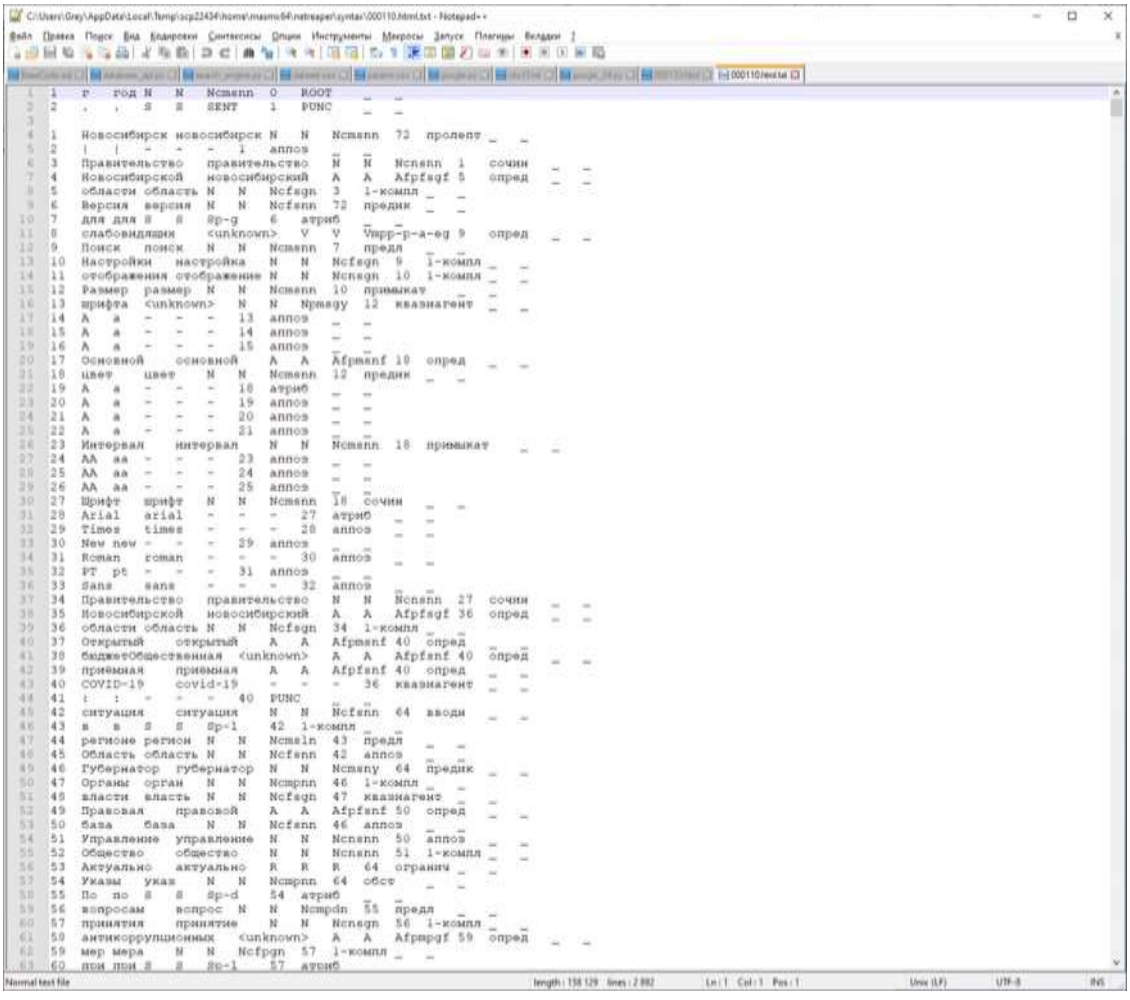

Рис. 5. Пример представления синтаксического дерева.

Первый столбец содержит номер узла в дереве, которое хранит это слово. Второй столбец – это само слово из предложения. В третий столбец помещается нормализованная форма этого слова. Столбец под номером 7 хранит номер родительского узла сформированного дерева. Остальные столбцы несут в себе

Математическая биология и биоинформатика. 2022. Т. 17. № 2. doi: 10.17537/2022.17.423

вспомогательную информацию, которая используется при других видах анализа (о части речи и т.д.). В решаемой здесь задаче эти данные не важны.

Пример 3. В качестве примера рассмотрим текст информационного сообщения, полученного с официальной страницы «ВКонтакте» оперштаба Алтайского края от 9 апреля 2020 года:

> *#коронавирус В Алтайском крае подтверждено еще 3 случая заболевания коронавирусной инфекцией. Итого на сегодня в регионе 21 заболевший, среди них два ребенка. Все случаи подтверждены в государственном научном центре вирусологии и биотехнологии «Вектор». Среди заболевших: 1 случай – завозной, 4 случая выявлены у больных пневмонией, 2 случая – у заболевших ОРВИ, 14 случаев – по контакту с заболевшими. Все – жители г. Барнаула. Специалистами алтаского Роспотребнадзора уточняется круг контактных лиц, осуществляется контроль за их лабораторным обследованием и проведением заключительной дезинфекции в местах проживания, организован весь комплекс противоэпидемических мероприятий. На 09.04.2020 в Алтайском крае на коронавирусную инфекцию обследовано 8007 человек. На данный момент под медицинским наблюдением находятся 1159 человек (в том числе, 7 иностранных граждан), снято с меднаблюдения 4378 человек.*

В результате NER-анализа с помощью библиотеки Natasha (рис. 6) успешно получены все топонимы: названия организаций, городов и регионов.

```
------ NER-анализ ------
Начальная форма     Нормализованный вид    tag

Алтайском крае      Алтайский край         LOC
Вектор              Вектор                 ORG
Барнаула            Барнаул                LOC
Роспотребнадзора    Роспотребнадзор        ORG
Алтайском крае      Алтайский край         LOC
------ NER-анализ (конец) ------
```

Рис. 6. Результат NER-анализа информационного сообщения из примера 3.

На следующем рисунке (рис. 7) представлен результат морфологического анализа для первого предложения из примера 3.

```
------ Морфологический анализ ------
В             ADP
Алтайском     ADJ | Case=Loc | Degree=Pos | Gender=Masc | Number=Sing
крае          NOUN | Animacy=Inan | Case=Loc | Gender=Masc | Number=Sing
подтверждено  VERB | Aspect=Perf | Gender=Neut | Number=Sing | Tense=Past |
              Variant=Short | VerbForm=Part | Voice=Pass
еще           ADB | Degree=Pos
3             NUM
случая        NOUN | Animacy=Inan | Case=Gen | Gender=Masc | Number=Sing
заболевания   NOUN | Animacy=Inan | Case=Gen | Gender=Masc | Number=Sing
коронавирусной ADJ | Case=Ins | Degree=Pos | Gender=Fem | Number=Sing
инфекцией     NOUN | Animacy=Inan | Case=Ins | Gender=Fem | Number=Sing
.             PUNCT
------ Морфологический анализ (конец) ------
```

Рис. 7. Результат морфологического анализа.

Разработанный алгоритм обработки новостных сообщений включает в себя поиск структурированной информации (списков) в каждом предложении новостной статьи. При наличии структурированной информации предложение обрабатывается с помощью регулярных выражений. В случае неструктурированной информации, как в примере 3, осуществляется лингвистический анализ, основанный на построении синтаксических деревьев. Программная реализация осуществлена на языке Python 3.8 с использованием библиотек Natasha, pandas и re. Схема алгоритма представлена на рисунке 8.

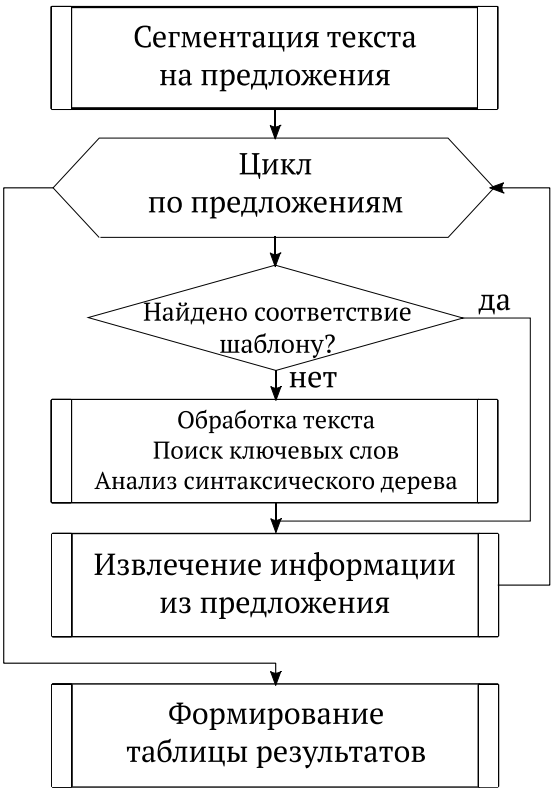

Рис. 8. Блок-схема работы предложенного алгоритма.

Работа алгоритма заключается в поиске ключевых слов в каждом предложении текста, и применении в последующем анализа синтаксического дерева, если ключевое слово найдено. Результат работы алгоритма для предложенного примера 3 представлен на рисунке 9.

	loc	general	fordays
0	Алтайский край	0	3
1	Алтайский край	21	0

Предложения с ключевыми словами неструктурированного текста:

0 : В Алтайском крае подтверждено еще 3 случая заболевания коронавирусной инфекцией.

1 : Итого на сегодня в регионе 21 заболевший, среди них два ребёнка.

Рис. 9. Результат обработки текстового сообщения из примера 3.

Из всего текста для анализа было выбрано по ключевым словам два предложения. Результатом работы алгоритма является таблица с тремя столбцами: место, общее число заболевших и число заболевших за сутки.

Пример 4. Рассмотрим пример с числительными в текстовом сообщении, которые необходимо исключить из результатов анализа:

#алтайвыручай

Четвертый инфицированный коронавирусом житель Алтайского края доставлен в городскую больницу N 5 Барнаула.

#алтайскийкрай #coronavirus #коронавирус

Заболевшая женщина накануне родила ребенка в Родильном доме КГБУЗ «Городская больница N 11» города Барнаула. Мама переведена в «Городскую больницу N 5 г.Барнаула», за ребенком осуществляется безпрерывное наблюдение медперсоналом роддома. Положительный результат анализа женщины подтвержден государственным научным центром вирусологии и биотехнологии «Вектор». По информации Минздрава Алтайского края, состояние всех четверых больных удовлетворительное. Проводятся противоэпидемические мероприятия, определяется круг контактных лиц.

На рисунке 10 представлен результат извлечения статистических данных из рассматриваемого текстового сообщения. В результате обработки удалось исключить числительное, представленное в текстовом виде, и номер больницы, который не относится к искомым значениям статистической информации.

Так же с помощью предложенного алгоритма можно получать данные из структурированного текста, в том числе представленные в смешанном виде.

loc	general	fordays
0 Алтайский край	4	0

Предложения с ключевыми словами неструктурированного текста:
0 : Четвёртый инфицированный коронавирусом житель Алтайского края доставлен в городскую больницу N 5 Барнаула..

Рис. 10. Результат работы алгоритма на данных из примера 4.

Пример 5. Рассмотрим пример текстового сообщения оперштаба Московской области от 21 июля 2021 года, в котором присутствуют данные, как в структурированном, так и неструктурированном виде:

2 161 случай заболевания коронавирусом выявлен в Подмосковье за сутки.
Всего на данный момент в Московской области выявлено 367 197 случаев заболевания коронавирусной инфекцией. Скончались 6 949 человек. Выздоровели и выписались 295 499 человек.
Выявленные случаи заболевания по городским округам: ...*
Зарайск 2551 (+26)
Звездный городок 53 (+0)
Лосино-Петровский 1172 (+3)
Мытищи 16405 (+80)
В подмосковных медицинских организациях наблюдаются и проходят лечение пациенты из Москвы и иных регионов России, а также из-за рубежа. Рекомендуем Вам при появлении любых симптомов ОРВИ вызывать врача на дом. В Подмосковье работает «горячая» линия по вопросам коронавируса по единому номеру «122» и сайт covid.mz.mosreg.ru.
** Статистика по городским округам составляется на основе данных по проведенным в Московской области тестам на коронавирус, предоставляемых Роспотребнадзором по МО.*
В статистике учитываются:
1. Граждане, сдававшие анализ на COVID–2019 в Московской области
2. Указавшие в графе «фактическое место проживания» один из городских округов Подмосковья.

Для сообщения из примера 5 основной текст (неструктурированный) обрабатывался с помощью синтаксического анализа (см. рис. 11, строки с индексами 0 и 1), а список городов (структурированный текст) – с помощью регулярных выражений (см. рис. 7, строки с индексами 0–9).

		general	fordays
0	Подмосковье	0	2161
1	Московская область	367197	0
2	Зарайск	2551	26
3	Лосино-Петровский	1172	3
4	Мытищи	16405	80
5	Наро-Фоминск	4447	44
6	Серебряные Пруды	1255	0
7	Серпухов	5455	14
8	Солнечногорск	7712	79
9	Ступино	5085	52

Предложения с ключевыми словами неструктурированного текста:

0 : 2 161 случай заболевания коронавирусом выявлен в Подмосковье за сутки..

1 : Всего на данный момент в Московской области выявлено 367 197 случаев заболевания коронавирусной инфекцией.

Рис. 11. Результат работы алгоритма на данных из примера 5.

ПЛАТФОРМА ДЛЯ АВТОМАТИЗАЦИИ СБОРА ДАННЫХ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ

Анализ всех региональных источников информации в интерактивном режиме на регулярной основе является трудновыполнимой задачей, поэтому важным элементом решения для сбора данных является автоматизация. Для построения архитектуры системы сбора данных из открытых источников выбрана программная платформа Apache Airflow [20]. Airflow – программное обеспечение для разработки, планирования и мониторинга ETL-процессов [20–22]. Основная особенность Airflow в том, что для описания процессов используется язык Python, который обеспечил его популярность и статус стандарта де-факто в инфраструктурах сбора данных.

Основной сущностью в Airflow является направленный ациклический граф работ (DAG). DAG позволяет описать последовательность задач для выполнения их в строго определенной последовательности по заданному расписанию. В процессе проектирования графа определяется набор вершин (операторов), реализующих алгоритмы по обработке данных. В Airflow существуют операторы для реализации практически любой задачи: вызов Python кода, выполнение SQL, запуск подпроцессов и так далее. Также возможно добавление пользовательских операторов – за счет расширения базового класса BaseOperator. Когда в проекте возникает часто используемый код, построенный на стандартных операторах, рекомендуется его преобразовывать в собственный оператор.

К основным компонентам платформы относятся:

(1) Airflow-worker – отдельный процесс, в котором выполняются задачи.

(2) Airflow-scheduler – компонент, отвечающий за запуск задач в требуемом порядке по заданному расписанию.

(3) Airflow webserver – отвечает за пользовательский интерфейс, где предоставляется возможность настраивать DAG и их расписание, отслеживать статус их выполнения.

Процесс получения новостных ресурсов и представленные алгоритмы обработки текстовых данных реализованы как отдельные Airflow-задачи, что позволяет обеспечить непрерывный мониторинг выбранных открытых источников информации. Поскольку языком программирования для описания задач в Airflow является Python это позволяет использовать современные наработки из области анализа данных.

Для хранения данных используется базы данных, допускающие обращение с неструктурированной информацией, например, PostgreSQL.

PostgreSQL – это система реляционных баз данных с открытым исходным кодом. PostgreSQL поддерживает запросы как SQL (реляционные), так и JSON (нереляционные). PostgreSQL – это очень стабильная база данных, поддерживаемая более чем 20-летним опытом разработки сообществом разработчиков ПО с открытым исходным кодом. PostgreSQL используется в качестве основной базы данных для многих веб-приложений, а также мобильных и аналитических приложений.

PostgreSQL обладает наборами функций: средство для управления параллелизмом (MVCC), восстановление на определенный момент времени, детальный контроль доступа, табличные пространства, асинхронную репликацию, вложенные транзакции, оперативное резервное копирование, усовершенствованный планировщик запросов и ведение журнала с опережающей записью. Он поддерживает международные наборы символов, многобайтовые кодировки символов, Unicode и поддерживает сортировку, чувствительность к регистру и форматирование с учетом локализации. PostgreSQL хорошо масштабируется, как по количеству данных, которыми он может управлять, так и по количеству одновременно обслуживаемых пользователей.

Логирование PostgreSQL с опережающей записью делает его очень отказоустойчивой базой данных. PostgreSQL совместим с ACID и полностью поддерживает внешние ключи, объединения, представления, триггеры и хранимые процедуры на многих разных языках.Он включает большинство типов данных SQL: 2008, включая INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL и TIMESTAMP. Он также поддерживает хранение больших двоичных объектов, включая изображения, звуки или видео.

Исходный код PostgreSQL доступен по лицензии с открытым исходным кодом, что дает свободу использовать, изменять и внедрять его бесплатно по своему усмотрению. PostgreSQL не требует лицензионных затрат, что исключает риск чрезмерного развертывания. Специальное сообщество участников и энтузиастов PostgreSQL регулярно находит ошибки и исправления, способствуя общей безопасности системы баз данных.

ЗАКЛЮЧЕНИЕ

Рассмотрен подход на основе совмещения статического метода и машинного обучения для извлечения статистических данных о процессе протекания пандемии COVID–19 из открытых новостных источников. Исследованы и проанализированы инструменты по сбору и анализу информации из неструктурированных источников. Данный подход позволяет извлекать фактографические данные из табличного формата, но без явного оформления таблицы в тексте обрабатываемого сообщения. В целях демонстрации работы подхода приведены примеры обработки нескольких новостей.

Предлагаемый подход позволил организовать автоматическую обработку текстовой информации для выбранных новостных каналов без явного указания структуры. С помощью такого решения удобно извлекать данные с региональных порталов, содержащей статистическую информацию о процессе развития эпидемии.

Для организации автоматизированного мониторинга новостных ресурсов с целью извлечения данных о пандемии реализовано техническое решение на основе программной платформы Apache Airflow. Данное решение позволило в автоматическом

режиме организовать сбор данных основных региональных источников с декабря 2019 года.

В дальнейшем предполагается применить данный подход к получению данных о введенных во время пандемии в различных административно-территориальных единицах ограничениях на основе анализа текстовой информации, представленной на естественном языке.

Работа выполнена при поддержке Минобрнауки России в рамках Соглашения № 075-11-2020-011 от 19.10.2020 (ИГК 0000000007520RHT0002).

СПИСОК ЛИТЕРАТУРЫ

1. Макаров В.Л., Бахтизин А.Р., Сушко Е.Д., Агеева А.Ф. Моделирование эпидемии COVID–19 – преимущества агент-ориентированного подхода. *Экономические и социальные перемены: факты, тенденции, прогноз*. 2020. Т. 13. № 4. С. 58–73.
2. *Мэрия Москвы потратила 516 млн. рублей на покупку данных о передвижениях горожан*. URL: https://www.rbc.ru/politics/04/03/2019/5c7cd5fe9a794760d9cfb900 (дата обращения: 13.05.2022).
3. *Онлайн статистика агентства LiveStats*. URL: http://www.internetlivestats.com (дата обращения: 20.04.2022).
4. Суханов А.А., Маратканов А.С. Анализ основных источников социальных данных в российском сегменте сети Интернет. *International Scientific Review*. 2017. № 1. С. 20–22.
5. Шигаров А.О. Проект интеллектуальной системы извлечения табличной информации из неструктурированных текстов. *Вестник Бурятского государственного университета*. 2013. № 9. С. 110–118.
6. Пивоварова Л.М. Фактографический анализ текста в системе поддержки принятия решений. *Вестник СПбГУ. Сер. 9. Филология. Востоковедение. Журналистика*. 2010. № 4. С. 190–197.
7. Власова Н.А. К проблеме разметки текстов на русском языке для задачи извлечения фактографической информации. *Программные системы: теория и приложения*. 2014. Т. 5. № 4(22). С. 67–82.
8. Хмельнов А.Е., Шигаров А.О. Метод извлечения таблиц из неформатированного текста. *Вычислительные технологии*. 2008. Т. 13. № 1. С. 93–101.
9. Виноградов А.Н., Воздвиженский И.Н., Кормалев Д.А., Куршев Е.П. Моделирование временного аспекта описания ситуации в задаче извлечения информации из текстов. *Программные системы: теория и приложения*. 2014. Т. 5. № 4(22). С. 215–229.
10. Вицентий А.В., Диковицкий В.В., Шишаев М.Г. Технология извлечения и визуализации пространственных данных, полученных при анализе текстов. *Труды Кольского научного центра РАН*. 2020. Т. 11. № 8(11). С. 115–119.
11. Kao A., Poteet S. *Natural language processing and text mining*. London: Springer-Verlag, 2007. 272 p.
12. Крутиков Н.О., Подаков Н.Г., Жилякова В.А. Разработка системы извлечения информации из текстов на русском языке в области криминалистики. *Проблемы информатики*. 2016. № 3. С. 70–84.
13. Комарова А.В., Менщиков А.А., Полев А.В., Гатчин Ю.А. Метод автоматизированного извлечения адресов из неструктурированных текстов. *International Journal of Open Information technologies*. 2017. Т. 5. № 11. С. 21–27.
14. Фридл Д. *Регулярные выражения*. СПб.: Символ-Плюс, 2008. 608 с. (Перевод с англ. Jeffrey E. F. Friedl. Mastering regular expressions, 2006).

15. Смоляков А.Л. Извлечение знаний из текстовой информации с помощью метода шаблонов. *Вестник СПбГУ. Прикладная математика. Информатика. Процессы управления*. 2008. Вып. 2. С. 44–50.
16. *Проект Natasha. Набор качественных открытых инструментов для обработки естественного русского языка (NLP)*. URL: https://habr.com/ru/post/516098 (дата обращения: 13.05.2022).
17. Гончаров А.С., Гончарова М.А. Роботизация обработки обращений граждан по вопросам социального и бытового обслуживания. *Наука без границ. Технические науки*. 2021. № 6(58). С. 40–52.
18. S. Sharoff, J. Nivre. The proper place of men and machines in language technology Processing Russian without any linguistic knowledge. 2011. http://corpus.leeds.ac.uk/serge/publications/2011-dialog.pdf (дата обращения: 01.12.2022).
19. Иншакова Е. С., Иомдин Л. Л., Митюшин Л. Г., Сизов В. Г., Фролова Т. И., Цинман Л. Л. СинТагРус сегодня. *Труды Института русского языка им. В. В. Виноградова*. 2019. № 21. С. 14–40.
20. 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.
21. Троценко Р.В., Болотов М.В. Процесс извлечения данных из разнотипных источников. *Приволжский научный вестник*. 2014. № 12. С. 52–54.
22. Талгатова З.Т. Анализ и сравнение существующих моделей процессов ETL для хранилищ данных. *Технические науки – от теории к практике*. 2016. № 1(54). С. 85–94.

Рукопись поступила в редакцию 24.11.2022, переработанный вариант поступил 29.11.2022.
Дата опубликования 04.12.2022.

========== INFORMATION AND COMPUTER ==========
========== TECHNOLOGIES IN BIOLOGY AND MEDICINE ==========

Extracting Factual Information about the Pandemic from Open Internet Sources

Akulinina E.Yu.[1], Karmanov A.L.[1], Teplykh N.A.[1],Vlasov V.V.[1], Baluta V.I.[2], Varykhanov S.S.[2], Karandeev A.A.[2], Osipov V.P.[2], Rykov Yu.G.[2], Chetverushkin B.N.[2]

[1]*FSUE «RFNC – VNIITF named after Academ. E.I. Zababakhin», Snezhinsk, Russia*
[2]*Keldysh Institute of Applied Mathematics RAS, Moscow, Russia*

Abstract. A large number of different source data is needed for multi-agent models of the spread of infectious diseases. Most of them are not directly accessible. Therefore, one of the key problems to design such models is the development of tools for obtaining data from various sources. This article presents approaches that allow to extract the values of the parameters of the functioning of the simulated society and statistical data on the development of the pandemic from text messages published in the Internet. The proposed method and software implementation provide intelligent search of open source information in the Internet and process of

unstructured data. The data collected this way used to set up parameters of mathematical model, which provides ability to study various scenarios and predict progress of the epidemic in concrete regions. The emphasis of the proposed approach is placed on two main technologies. The first is the use of regular expressions. The second is analysis using machine learning methods. The use of the regular expression method allows for high-speed text processing, but its applicability is limited by a strong dependence on the context. Machine learning allows to adapt the information context of the message, but at the same time there is a relatively large amount of time spent on analysis. To improve the accuracy of the analysis and to level the shortcomings of each of these approaches, ways of combining these technologies are proposed. The article presents the obtained results of optimization of algorithms for obtaining the necessary data.

Key words: *text data analysis, regular expressions, syntax trees, data collection platform.*