DOI: 10.15593/2499-9873/2019.4.05
УДК 004.9:519.237

Ю.И. Володина, М.Я. Зекирьяев

Березниковский филиал Пермского национального исследовательского политехнического университета, Березники, Россия

МЕТОДЫ И СРЕДСТВА ТЕКСТОВОГО АНАЛИЗА В СИСТЕМЕ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЕЙ

Изложены основные проблемы естественно-языкового анализа текста в заявках пользователей в отделе технической поддержки на предприятии. Определены и поставлены цели и задачи, проведено обоснование актуальности данного исследования. Проведен анализ существующих программных продуктов, выделены их преимущества и недостатки. Предложен комплексный метод семантического анализа естественно-языкового текста и формирования естественно-языковых баз знаний с использованием морфологического анализатора русского языка. Разработано программное средство, реализующее такие возможности, как импорт данных из существующей системы, поиск ключевых слов по полям таблицы для проведения обучения, выделения ключевых слов, слов-отрицаний и стоп-слов, построение дерева связей между ключевыми словами и заявками для выявления проблемных заявок, добавление проблем и направлений с привязкой их к ключевым словам для их дальнейшего анализа путем сравнения количества заявок за разные периоды времени и возможностью отправки результатов ответственным по направлениям, а также получение данных по выполненным заявкам путем формирования заявок в разрезах по исполнителям, проблемам и направлениям.

Ключевые слова: data mining, text mining, mystem, suffix tree clustering, текстовый анализ, поддержка пользователей.

Yu.I. Volodina, M.Y. Zekiryaev

Berezniki Branch of Perm National Research Polytechnic University, Berezniki, Russian Federation

TEXT ANALYSIS METHODS AND TOOLS IN THE USER SUPPORT SYSTEM

This article describes the main problems of natural language text analysis of user requests in the technical support department at the enterprise. The aims and objectives are defined and set, the relevance of this research is substantiated. The analysis of existing software products is conducted, its advantages and disadvantages are identified. The complex method of semantic analysis of the natural language text and formation of natural language knowledge bases with the use of a morphological Russian language analyzer is proposed.

A software that implements such features as importing data from the existing system; keyword searching by table fields for learning; keywords, negative words and stop-words selecting; building a links-tree between keywords and requests for identifying problematic requests; adding problems and areas of concern with keywords gridded for its further analysis by comparing the number of requests for different periods of time and the opportunity of sending the results to those responsible for the areas, and obtaining data on completed requests through the formation of requests in terms of implementers, problems and areas is developed.

Keywords: data mining, text mining, mystem, suffix tree clustering, text analyses, helpdesk.

Развитие информационных технологий и внедрение новых информационных систем и услуг в компании и ее дочерних обществах влечет за собой необходимое построение эффективной поддержки пользователей, обслуживаемых организаций.

Автоматизация работы отделов технической поддержки происходит путем внедрения автоматизированного средства HelpDesk. HelpDesk – это информационная система технической поддержки, которая позволяет решать проблемы пользователей с компьютерами и программным обеспечением, в котором они работают. Данная система является важнейшей частью ITIL (IT Infrastructure Library), она позволяет выявить проблемные места в ИТ-структуре и оценить работу отдела поддержки пользователей. Одной из основных функций таких отделов является поддержка информационных систем в рабочем состоянии.

В связи с внедрением и обновлением поддерживаемых информационных систем растет число заявок (обращений) пользователей, связанных с настройкой, устранением ошибок, вопросами об использовании информационных систем и т.д. В таком случае эффективная работа ИТ-отдела становится более значимой для развития бизнеса. Ошибки или сбои в работе систем могут привести к финансовым потерям, задержке выплаты зарплаты работникам, несвоевременной сдаче отчетности в государственные органы, задержке оплаты поставщикам и т.д. Поэтому важной задачей является повышение качества обслуживания специалистами поддержки и выполнения ими своих функциональных обязанностей за счет снижения количества заявок пользователей путем выявления проблем и ошибок в информационных системах.

На сегодняшний день одним из основных способов передачи информации является естественный язык. Он обладает множеством свойств, которые осложняют понимание его вычислительной техникой. Например, к ним можно отнести избыточность, социальность и непрозрачность естественного языка. Необходимость в изучении и понимании увеличивающегося с каждым днем объема неструктурированных текстовых данных делает задачу обработки и анализа таких данных актуальной. За последние несколько лет в мире было проведено множество различных исследований, связанных с классификацией текстовых данных пользователей. Существует несколько различных подходов, в соответствии с которыми тексты могут быть разделены на следующие классы:

1) субъективные и объективные [1];

2) плохие и отличные (thumbs down и thumbs up) [2];

3) положительные и отрицательные [3];

4) положительные, отрицательные и нейтральные [2, 3];

5) фальшивые и подлинные [4].

Задачу классификации текстовых данных относят к основным задачам при автоматическом анализе естественно-языкового текста. Для оценки качества работы методов классификации применяют стандартные метрики анализа текстов, такие как точность, полнота, достоверность.

Методы автоматической классификации текстовых данных разделяют на три группы:

1) автоматическое обучение без учителя [5];

2) обучение с учителем [3];

3) методы, позволяющие проводить лингвистический анализ текста, работающие по синтаксическим правилам и шаблонам.

Основной целью обработки текстовых данных на сегодняшний день является классический анализ тональности, в котором используется словарь оценочных слов и статистические меры.

В настоящее время имеется незначительное количество научных работ, которые посвящены формированию словаря оценочных слов на русском языке. Автор работы [7] предлагает метод, позволяющий посчитать вес оценочных слов с использованием пяти статистических мер на массиве коротких сообщений. В работе [8] автор использует метод извлечения русскоязычного предметно-ориентированного словаря оценочных слов. Работа [9] заключается в анализе применения тематических моделей с целью получения однословных терминов.

На сегодняшний день нет доступных словарей оценочных слов на русском языке, основанных на аспектах первых двух групп. Также неизвестны исследования с использованием измененных тематических моделей для выполнения задачи анализа заявок на русском языке.

Первым этапом обработки текстовых данных для создания групп критериев, к которым далее будут применены различные методы и модели, является кластеризация. Требуется обратить внимание на то, что задача кластеризации текстовых документов очень схожа с классификацией документов. Классификация – это группировка документов по классам с заранее определенными параметрами, которые были получены при обучении. Количество таких классов ограничено.

Задача кластеризации документов – это группировка списка документов на кластеры (подмножества). Предварительно параметры таких классов неизвестны, а количество кластеров может быть неограниченным или заранее фиксированным. Также все методы разделяют на числовые и нечисловые. Числовые методы характеризируются использованием числовых данных о текстовых документах. Нечисловые – это методы, в которых для работы используются слова и фразы. Задача кластеризации текстовой информации является одним из основных вопросов Data Mining, очень часто она имеет практический характер и применяется в различных областях, например прогнозирование каких-либо событий, анализ текстовых данных и изображений.

Задачу кластеризации текста очень трудно формализовать, в то время как у классификации существуют объективные и точные методы оценки качества. Оценить качество кластеризации в численных показателях трудно, так как они учитываются в зависимости от мнения эксперта. Таким образом, одной из основных проблем кластеризации текстовой информации является оценка качества результата, полученного при обработке, так как на сегодняшний день не выявлено ни одного метода оценки, который был бы применим во всех случаях кластеризации.

В табл. 1 произведено сравнение основных алгоритмов кластеризации. В результате можно сделать вывод, что лучшими из предложенных методов являются Suffix Tree Clustering и Concept Indexing (работающий по методу рекурсивной бисекции), имеющие неплохую точность результатов и скорость вычисления. Однако метод Suffix Tree Clustering лучше за счет более наглядного представления кластеров в результатах выполнения кластеризации.

Проанализировав заявки пользователей, выделили следующие типы фраз:

1. Заявка на устранение ошибки с явным упоминанием. Содержит в себе явное указание на ошибку в системе, например «не открывается документ, выходит ошибка», «ошибка при печати отчета», «не запускается программа» и т.д.

2. Заявка на устранение ошибки с косвенным упоминанием. Данный тип не содержит в себе явного упоминания ошибок, но содержит вспомогательные слова, которые подразумевают саму ошибку. Например, «не открывается документ», «не печатается акт», «не могу войти в программу» и т.д.

Таблица 1

Сравнение основных алгоритмов кластеризации

Наименование алгоритма	Вид	Ограничение	Пересекаемость кластеров	Инкрементность	Использование числовых характеристик	Требуется предварительное обучение	Скорость работы
Single Link, Complete Link, Group Average	Числовой, «снизу вверх», кластеризирующий	По количеству документов в кластере	–	+	Матрица близости	–	*SL* – $O(n^2)$ *CL* – $O(n^3)$ *GA* – $O(n^2)$
Suffix Tree Clustering	Нечисловой, кластеризирующий	–	+	+	–	–	$O(k^2N)$, где k – число кластеров, N – число документов
Scatter/Gather	Числовой, «снизу вверх», кластеризирующий	По количеству кластеров	–	–	Матрица близости	–	*Buckshot* – $O(kN)$ *Fractionation* – $O(mN)$, где $m = O(k)$, где k – число кластеров
Latent Semantic Indexing	Числовой, кластеризирующий	По количеству кластеров	–	+	Матрица *TF–IDF*	–	N^2xk, где k – факторы
Concept Indexing обучаемый	Числовой, классифицирующий	По количеству кластеров	–	–	Матрица близости или матрица *TF–IDF*	+	Неизвестна
Concept Indexing необучаемый	Числовой, «сверху вниз», кластеризирующий	По количеству кластеров	–	+	Матрица близости	–	$O(N \cdot \log k)$, где k – число кластеров
K-means	Числовой, кластеризирующий	По количеству кластеров и центроидов	–	–	Матрица *TF–IDF*	–	$O(n)$
SOM	Числовой, классифицирующий	По количеству кластеров	+	+	Матрица близости или матрица *TF–IDF*	+	Неизвестна

3. Заявки на устранение пользовательских ошибок или исправление данных. В этот тип входят заявки, сообщающие об ошибках в пользовательских данных вследствие человеческого фактора, пользователь не знает, как исправить ошибку, или не имеет доступа к исправ-

лению. Например, «на остатках не проводится», «не проводится», «слетело сальдо», «не идут итоги», «посмотрите регистры» и т.д.

4. Заявки на предоставление конкретному пользователю доступа к системе или определенному функционалу. Данный тип проблем может решаться только администраторами системы, так как у остальных пользователей нет доступа к этой функции. Например, «нет доступа к вкладке», «предоставить доступ», «пропали права» и т.д.

Обозначим текстовую конструкцию $phrase_{ij} = \left(r\left(s_{ij}\right), s_{ij}\right)$, где $r\left(s_{ij}\right) \in [0,1]$ и обозначает численное значение связи между s_{ij} и проблемными заявками или высказываниями.

В организации существует некоторое множество автоматизированных систем $S = \{S_1, S_2, \ldots, S_n\}$, в которых работают пользователи. Каждая коллекция текстовых данных состоит из заявок пользователей, состоящих из ошибок, найденных в системе, из определенной предметной области (автотранспорт, бухгалтерия, заработная плата и т.д.).

Для каждой системы $S_i \in S$ задается множество заявок пользователей $R = \{r_1, r_2, \ldots, r_m\}$, где $r_i = \left\{s_{i1}, s_{i2}, \ldots, s_{i|r_i|}\right\}$ и s_{ij} являются текстами заявок пользователей. Некоторые заявки являются разовыми или выступают предложениями на доработку системы. Каждая система $S_i \in S$ состоит из некоторого множества классов (категорий) $C = \{c_1, \ldots, c_h, \ldots, c_n\}$. В таком случае для каждого класса требуется сформировать множество признаков $F(C) = \cup F(C_r)$ при $F(C_r) = \left\langle t_1^r, \ldots, t_k^r, \ldots, t_z^r \right\rangle$. Набор признаков $F(C_r)$ является словарем, который состоит из характеристик соответствующего класса заявки – лексем.

Каждая из обрабатываемых заявок имеет набор признаков, при помощи которых заявку можно сравнить с классом $F(r_i) = \left\langle t_1^i, \ldots, t_l^i, \ldots, t_y^i \right\rangle$. Множество всех признаков классов должно совпадать с множеством признаков заявок $F(C) = F(R) = \cup F(r_i)$.

Таким образом, отношение заявки r_i к классу c_r вычисляется при помощи $F(r_i) \cap F(c_h)$. Задачей существующих методов класси-

фикации является поиск набора признаков и построение набора правил, с помощью которых будет приниматься решение о принадлежности заявки к определенной группе или классу, т.е. ставится задача классификации путем построения функции F', которая близка к функции цели F', на вход которой передаются значения $\langle r_i, c_h \rangle$, а на выходе получают информацию о том, принадлежит или нет заявка r_i классу c_h.

Анализ заявок методами Text Mining выполняется по следующим шагам:

1. Поиск информации в тексте. На данном этапе выполняется определение заявок для дальнейшего анализа и обработки.

2. Предварительная обработка заявок. На этом этапе алгоритма тексты заявок преобразуются в удобную для работы форму. В результате данного этапа на выходе имеется текст без лишних слов и спецсимволов, которые могут повлиять на результаты анализа.

3. Получение необходимой информации. Из обрабатываемого текста происходит формирование наборов ключевых слов, анализ которых потребуется провести.

4. Использование методов Text Mining. Основной этап анализа заявок, в котором собираются новые знания и скрытые закономерности в данных.

5. Анализ и вывод полученных результатов. Вывод результатов работы алгоритма в удобной для пользователя форме.

На этапе предварительной обработки заявок выполняется токенизация текста, т.е. разбиение документа на слова, предложения или абзацы. Токен – результат выполнения токенизации. После выполнения разбиения чаще всего проводят фильтрацию слов, которые никаким образом не влияют на смысл текстов (междометия, предлоги, частицы, союзы и др.). Такие слова называют стоп-словами. Их список составляется перед обработкой текста в зависимости от языка. После очистки текста от стоп-слов выполняется нормализация слов, так называемая лемматизация и стемминг, где каждое слово приводится в начальную форму для более легкой обработки данных. Все слова текста ставятся в именительный падеж и единственное число. К недостаткам данного приема можно отнести возможное нарушение смысла предложений или фраз, поэтому необходимо обязательно учитывать язык текста, как и при токенизации.

На основании описания задачи, нам необходимо извлечь преобладающие заявки (фразы заявок), которые указывают на проблемные ситуации в информационных системах $S_i \in S$ и их целевых объектах (направлениях) T_i, используя перечень пользовательских заявок *R*.

Используя особенности задач анализа мнений [10], можно разделить исходную задачу на следующие подзадачи:

1. Определение ключевых слов (фраз) и стоп-слов для заявок, которые указывают на проблемные ситуации в информационной системе.

2. Отбор заявок по ключевым фразам и словам, которые указывают на проблемные ситуации в информационной системе, из текстов заявок пользователей.

3. Отбор тематически сгруппированных направлений информационных систем для получения преобладающих проблем в перечне заявок пользователей по определенным предметным областям.

4. Вывод полученных результатов для возможности определения пользователем основных проблем в системах по конкретным предметным областям.

5. Сбор статистических данных о заявках за разные периоды времени для возможности отображения количества заявок до и после принятия решения по устранению проблемы.

Процесс нормализации текста разделяют на две шага:

1. Стемминг – получение корня слова (неизменной части слова).

2. Синтезирование нормальной формы слова – получение суффиксов нормальной формы.

Особое внимание необходимо уделить первому шагу, так как синтез нормализованного слова очень сильно зависит от алгоритма получения стеммы, также необходимо отметить, что большая часть алгоритмов синтезирования синтезирует все возможные варианты лемм и дает множество результатов.

Существует три популярных стеммера, которые основаны на различных принципах и могут обрабатывать не существующие в русском языке слова: стеммер Портера SnowBall, Mystem, Stemka.

Алгоритм стеммер Портера был разработан Мартином Портером в 1979 г. для обработки слов на английском языке [11, 12]. Изначально стеммер Портера был написан на не используемом в наше время языке программирования Basic Combined Programming Language (BCPL). По-

сле был создан проект SnowBall, где были дописаны стеммеры для большинства языков, за основу которых был взят оригинальный алгоритм. Данный алгоритм включает в себя 5 шагов. Каждый шаг алгоритма отрезает слово- или формообразующий суффикс, после чего оставшаяся часть слова проверяется на соответствие установленным правилам. К примеру, у русских слов должно быть не меньше одной гласной буквы. Если полученное на первом шаге слово не удовлетворяет всем правилам, то алгоритм выбирает другой суффикс для обработки, иначе происходит переход к шагу 2. Согласно информации на официальном сайте [13], алгоритм отсекает суффиксы в следующем порядке:

1) максимально возможный формообразующий суффикс;
2) буква «и»;
3) словообразующий суффикс;
4) превосходные формы суффикса;
5) одна из двух букв «н», мягкий знак.

Преимущества алгоритма:

– не использует словари и базы основ слов;
– быстродействие.

Недостатки алгоритма:

– часто обрезает слова больше, чем требуется;
– человеческий фактор при установке правил.

В отличии от стеммера Портера, алгоритм Stemka возвращает набор данных из нескольких стемм, которые удовлетворяют таблицам перехода. В связи с этим при сравнении с другими алгоритмами обычно рассматривается два режима работы: агрессивный и консервативный. Стоит отметить, что данный алгоритм использует то же условие, что и стеммер Портера, – наличие как минимум одной гласной в стемме.

Первым шагом алгоритма MyStem является определение возможных границ между суффиксами и стеммой, которое выполняется при помощи дерева суффиксов. После этого для каждой стеммы происходит бинарный поиск по дереву с основами слов. Если стемма не найдена, то ищутся самые близкие к ней основы. Если слово найдено в словаре, работа алгоритма заканчивается; если слово не нашлось, алгоритм разбивает слово дальше. Если полученный вариант основы слова так и не совпал ни с одной словарной основой, то создается гипотетическая модель изменений данного слова. Эта гипотеза сохраняется,

а если ранее уже была построена аналогичная, то ее вес увеличивается. Если гипотеза не была найдена в словаре, тогда длина требуемых окончаний основ слов уменьшается на одну единицу, и поиск повторяется снова до появления новых гипотез. Когда длина слова достигает 2, поиск по словарю останавливается и происходит сортировка всех гипотез по их весам. Если вес гипотез менее самого большого веса в 5 и более раз, то эта гипотеза отсеивается. В результате работы MyStem собирается набор гипотез или одна гипотеза для несуществующего слова.

В статье [14] И. Сегалович проводит сравнение перечисленных стеммеров. Критерием оценки качества является количество пар «форма слова – основа слова» (PPMV), которые получаются в результате работы алгоритмов. Проверка всех алгоритмов происходила на национальном корпусе русского языка [15]. В результате, хотя Mystem и находит очень много излишних пар, в отличие от других алгоритмов, но добавленные Mystem пары наиболее семантически близки. Также у Mystem самое минимальное количество потерянных пар и лучшая семантическая близость.

Таким образом, для разработки данного программного обеспечения будет использована комбинация алгоритмов Mystem и Suffix Tree Clustering, которые показали наилучшие результаты на этапе сравнения (см. табл. 1). Для обработки заявок и получения ключевых слов будет использован алгоритм Mystem, который позволяет произвести токенизацию текста, т.е. разбивку документа на слова, предложения или абзацы. После чего – нормализовать каждое слово, привести его в единственное число именительного падежа. Алгоритм Suffix Tree Clustering используется для получения дерева заявок по ранее полученным ключевым словам и дальнейшего присвоения им проблем.

Разработанное программное средство имеет следующие возможности:

1. Импорт данных из выгруженного из программы HelpDesk excel-файла в таблицу базы данных.

2. Поиск ключевых слов по полям таблицы для проведения дальнейшего обучения программы.

3. Проведение обучения с участием пользователя для определения ключевых слов и стоп-слов.

4. Нормализация слов, поиск слов-отрицаний.

5. Построение дерева связей между ключевыми словами и заявками для выявления проблемных заявок.

6. Добавление проблем и направлений с привязкой их к ключевым словам для дальнейшего анализа.

7. Анализ имеющихся проблем путем сравнения количества заявок за разные периоды времени.

8. Анализ текущих проблем за выбранный период времени с возможностью отправки полученных результатов на электронную почту ответственным за направления путем формирования excel-файла.

9. Получение данных по выполненным заявкам путем формирования заявок в разрезах по исполнителям, проблемам и направлениям.

В систему были загружены заявки из MS Excel, после чего система была обучена для выделения стоп-слов, отрицаний и ключевых слов для последующего использования в поиске проблемных заявок. В процессе обучения также происходит очистка от запрещенных символов (например, «!», «@», «#», «$», «%», «^», «&» и т.д.) и выполняется «кирилификация» текста, так как предполагается, что все заявки пользователей являются русскоязычными, происходит замена символов, написание которых схоже на русском и английском языках, например таких символов, как «a», «c», «e» и т.д.

Далее была проведена нормализация слов предложений с использованием морфологического анализатора русского языка MyStem. После выполнения нормализации в предложении были найдены слова-отрицания. Если отрицание найдено и оно не в конце строки, тогда выбирается следующее за ним слово, в ином случае – предыдущее.

После обучения была выполнена привязка ключевых слов к проблемам и направлениям, в результате получено дерево ключевых слов и связанных с этими словами заявок, а также произведена связка ключевых слов с проблемой.

В итоге были получены графики по проблемам и направлениям, так, например, на рисунке представлен график сравнения заявок за период с 01.01.2018 г. по 31.05.2018 г. и с 01.01.2019 г. по 31.05.2019 г.

При использовании разработанного программного обеспечения, выполнена проверка заявок за первые 5 мес. 2018 и 2019 г., и видно, что после анализа проблем к концу 2018 г. ответственными по направлениям были предприняты меры по устранению проблем в системах.

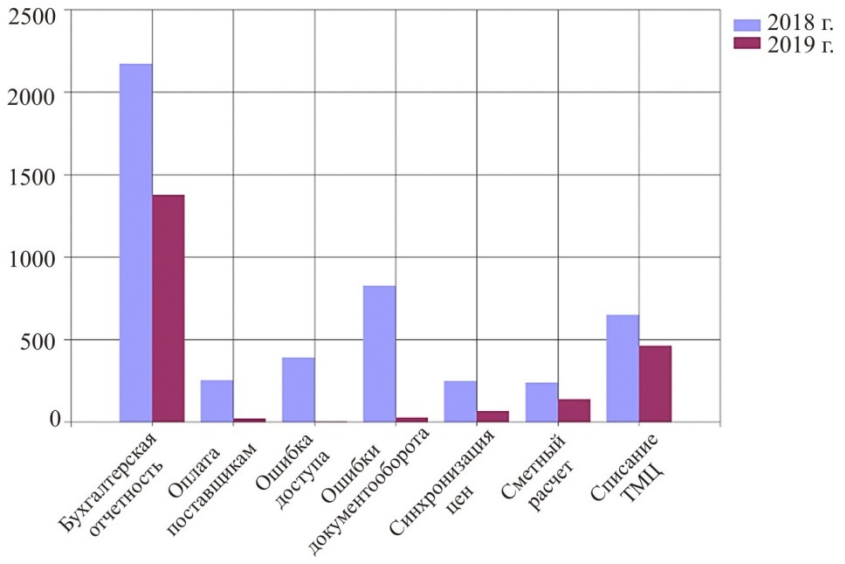

Рис. Результат сравнения заявок

Таблица 2

Сравнение заявок за 5 мес. 2018 и 2019 г.

Направление	Количество заявок в 2018 г.	Количество заявок в 2019 г.	Разница
Администрирование	389	212	177
Бухгалтерский учет	2101	1470	631
Документооборот	826	78	748
Производство	310	56	254
Склады и закупки	1150	181	969
Всего	4776	1997	2779

В табл. 2 мы видим, что количество заявок по проблемным направлениям на конец мая 2018 г. составляло 4776. После устранения проблем по большинству направлений в 2019 г. количество заявок снизилось до 1997, а именно на 2779 заявок, или 58 %.

Таким образом, мы видим, что разработанное программное обеспечение позволило определить проблемные места в информационных системах и существенно снизить количество соответствующих заявок.

Список литературы

1. Balahur A., Mihalcea R., Montoyo A. Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications // Computer Speech & Language. – 2014. – Vol. 28, no. 1. – P. 1–6.

2. Pang B., Lee L., Vaithyanathan S. Thumbs up? sentiment classification using machine learning techniques // Proceedings of the ACL-02 conference on Empirical methods in natural language processing / Association for Computational Linguistics. – 2002. – Vol. 10. – P. 79–86.

3. Kiritchenko S., Zhu X., Mohammad S.M. Sentiment analysis of short informal texts // Journal of Artificial Intelligence Research. – 2014. – № 50. – P. 723–762. – DOI: 10.1613/jair.4272

4. Finding deceptive opinion spam by any stretch of the imagination / M. Ott [et al.] // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies / Association for Computational Linguistics. – 2011. – Vol. 1. – P. 309–319.

5. Катасев А.С., Катасева Д.В., Кирпичников А.П. Нейросетевая технология классификации электронных почтовых сообщений // Вестник Технологического университета. – 2015. – Т. 18, № 5. – С. 180–183.

6. Kohonen T. Self-Organization of Very Large Document Collections: State of the Art. Perspectives in Neural Computing // Proceedings of the 8th International Conference on Artificial Neural Networks, Skövde, Sweden, 2–4 September 1998 – London: Springer-Verlag London, 1998. – P. 65–74. – DOI: 10.1007/978-1-4471-1599-1_6

7. Рубцова Ю.М. Метод построения и анализа корпуса коротких текстов для задачи классификации отзывов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: тр. XV Всерос. науч. конф. RCDL. – Ярославль: Изд-во Ярославского государственного университета им. П.Г. Демидова, 2013. – С. 269–275.

8. Chetviorkin I., Braslavskiy P., Loukachevich N. Research of lexical approach and machine learning methods for sentiment analysis // Computational Linguistics and Intellectual Technologies. – 2012. – Vol. 2. – P. 1–14.

9. Nokel M., Loukachevitch N. Application of Topic Models to the Task of SingleWord Term Extraction // CEUR Workshop Proceedings. – 2013. – Vol. 1108. – P. 52–60.

10. Pang B., Lee L. Opinion mining and sentiment analysis // Foundations and Trends in Information Retrieval. – 2008. – Vol. 2, № 1–2. – P. 1–135.

11. Затонский А.В., Варламова С.А., Беккер В.Ф. Построение и анализ системы управления качеством образования вуза // Автоматизация и современные технологии. – 2009. – № 5. – С. 35–42.

12. Porter M. Official home page for distribution of the Porter Stemming Algorithm, written and maintained by its author, Martin Porter. – URL: https://tartarus.org/martin/PorterStemmer (accessed 14 October 2019).

13. Russian stemming algorithm SnowBall. – URL: http://snowball.tartarus.org/algorithms/russian/stemmer.html (accessed 14 October 2019).

14. Сегалович И.А. Fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. – URL: https://pdfs.semanticscholar.org/983b/7014df3b7d4e82e32ba4f45f71f3879f8c96.pdf (дата обращения: 14.10.2019).

15. Национальный корпус русского языка [Электронный ресурс]. – URL: http://www.ruscorpora.ru/new (дата обращения: 14.10.2019).

References

1. Balahur A., Mihalcea R., Montoyo A. Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications. *Computer Speech & Language*, 2014, vol. 28, no. 1, pp. 1-6.

2. Pang B., Lee L., Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. 2002, vol. 10, pp. 79-86.

3. Kiritchenko S., Zhu X., Mohammad S.M. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 2014, no. 50, pp. 723-762, DOI: 10.1613/jair.4272.

4. Ott M., Choi Ye., Cardie C., Hancock J.T. Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, vol. 1, pp. 309-319.

5. Katasev A.S., Kataseva D.V., Kirpichnikov A.P. Neirosetevaia tekhnologiia klassifikatsii elektronnykh pochtovykh soobshchenii [Neuronet technology of the e-mail massages classification]. *Vestnik tehnologicheskogo universiteta*, 2015, vol. 18, no. 5, pp. 180-183.

6. Kohonen T. Self-Organization of Very Large Document Collections: State of the Art. Perspectives in Neural Computing. *Proceedings of the 8th International Conference on Artificial Neural Networks.* London, Springer-Verlag London, 1998, pp. 65-74, DOI: 10.1007/978-1-4471-1599-1_6

7. Rubtsova Iu.M. Metod postroeniia i analiza korpusa korotkikh tekstov dlia zadachi klassifikatsii otzyvov [A method for development and analysis of short text corpus for the review classification task] / *Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollektsii: Trudy XV Vserossijskoj nauchnoj konferencii RCDL.* Yaroslavl, Izdatel'stvo Iaroslavskogo gosudarstvennogo universiteta im. P.G. Demidova. – 2013. – pp. 269-275.

8. Chetviorkin, I. Braslavskiy P., Loukachevich N. Research of lexical approach and machine learning methods for sentiment analysis. *Computational Linguistics and Intellectual Technologies*, 2012, vol. 2. pp. 1-14.

9. Nokel M., Loukachevitch N. Application of Topic Models to the Task of Single-Word Term Extraction. *CEUR Workshop Proceedings*, 2013, vol. 1108, pp. 52-60.

10. Pang B., Lee L. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2008, vol. 2, iss. 1-2, pp. 1-135.

11. Zatonskii A.V., Varlamova S.A., Bekker V.F. Postroenie i analiz sistemy upravleniia kachestvom obrazovaniia vuza [Management system construction and analysis of the higher education quality]. *Avtomatizatsiia i sovremennye tekhnologii*, 2009, no 5, pp. 35-42.

12. Porter M. Official home page for distribution of the Porter Stemming Algorithm, written and maintained by its author, available at: https://tartarus.org/martin/PorterStemmer (accessed 14 October 2019)

13. Russian stemming algorithm SnowBall. available at: http://snowball.tartarus.org/algorithms/russian/stemmer.html (accessed 14 October 2019)

14. Segalovich I.A. Fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. available at: https://pdfs.semanticscholar.org/983b/7014df3b7d4e82e32ba4f45f71f3879f8c96.pdf (accessed 14 October 2019)

15. Natsional'nyi korpus russkogo iazyka. available at: http://www.ruscorpora.ru/new (accessed 14 October 2019)

Получено 14.10.2019

Сведения об авторах

Володина Юлия Игоревна (Березники, Россия) – кандидат технических наук, доцент, кафедра «Автоматизация технологических процессов», Березниковский филиал Пермского национального исследовательского политехнического университета (618404, Березники, ул. Тельмана, 7, e-mail: julia_volodina@mail.ru).

Зекирьяев Марсель Якубович (Березники, Россия) – магистрант, кафедра «Автоматизация технологических процессов», Березниковский филиал Пермского национального исследовательского политехнического университета (618404, Березники, ул. Тельмана, 7, e-mail: atp@bf.pstu.ru).

About the authors

Yulia I. Volodina (Berezniki, Russian Federation) – Ph.D. in Engineering, Associate Professor, Department of Automation of Technological Processes, Berezniki branch of Perm National Research Polytechnic University (618404, Berezniki, Telmana st., 7, e-mail: julia_volodina@mail.ru).

Marsel Ya. Zekirjaev (Berezniki, Russian Federation) – Master Student, Berezniki branch of Perm National Research Polytechnic University (618404, Berezniki, Telmana st., 7, e-mail: atp@bf.pstu.ru).