

межвузовский сборник научных трудов

**МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

РГРТУ, 2020

УДК 004.4

Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина – Рязань: Издательство ИП Коняхин А.В. (Book Jet), декабрь 2020. – 92 с.

ISBN 978-5-907400-08-5

Представлены статьи, посвященные различным аспектам разработки программных средств ЭВМ, вычислительных систем и сетей, вопросам математического моделирования, методам обработки информации.

Предназначен для научно-педагогических работников вузов, инженеров, аспирантов и студентов старших курсов.

Авторская позиция и стилистические особенности публикаций полностью сохранены.

Редакционная коллегия

д-р техн. наук, проф., засл. работник высшей школы РФ А.Н. Пылькин (отв. редактор, Рязанский государственный радиотехнический университет); д-р техн. наук, проф., зав. каф. Г.В. Овечкин (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. Е.Е. Ковшов (Московский государственный технологический университет «Станкин»); д-р техн. наук, проф. К.А. Майков (МГТУ им. Н.Э. Баумана); д-р техн. наук, проф. В.В. Белов (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. В.В. Золотарев (Институт космических исследований РАН, г. Москва); д-р техн. наук, проф. И.Ю. Каширин (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. В.Н. Малыш («Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» г. Липецк); И.А. Благодарова (Рязанский государственный радиотехнический университет).

Рецензент: к.т.н., доцент, зав. каф. «Информатики, информационных технологий и защиты информации» Липецкого государственного педагогического института Скуднев Д.М.

ISBN 978-5-907400-08-5

© Коллектив авторов, 2020

© ИП Коняхин А.В. (Book Jet), 2020

ВВЕДЕНИЕ

Межвузовский сборник научных трудов содержит результаты научных исследований и разработок по следующим направлениям:

- программное обеспечение вычислительных систем, новые информационные технологии;
- прикладная математика, теория информации, искусственный интеллект;
- ЭВМ в системах обработки, управления и обучения;
- автоматизированное проектирование аппаратных средств вычислительных систем;
- информационные технологии в экономических и социальных системах.

Сборник сформирован на основе статей, в которых рассматриваются различные аспекты разработки программных средств ЭВМ, вычислительных систем и сетей, включая вопросы автоматизации проектирования, теории обработки информации и математического моделирования, и предназначен для студентов, аспирантов и преподавателей технических вузов и научных работников.

Материалы для сборника предоставлены сотрудниками и студентами:

- Рязанского государственного радиотехнического университета им. В.Ф. Уткина г. Рязань;
- Рязанского государственного медицинского университета им. академика И.П. Павлова, г. Рязань.

УДК 519.812.3

Белов В.В., Чистякова В.И.

СИСТЕМАТИЧЕСКИЙ ПОДХОД К ПРОЕКТИРОВАНИЮ
ИНФОРМАЦИОННЫХ СИСТЕМ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

Излагается подход к проектированию, названный систематическим подходом, рекомендуемый в качестве основного подхода для реализации курсового проектирования в рамках дисциплин, связанных с разработкой информационного и программного обеспечения систем.

Рассматриваемая задача

Основу технологии проектирования информационных систем [1] составляет методология, которая определяет сущность и основные отличительные технологические особенности. Методология проектирования предполагает наличие некоторой концепции – совокупности принципов проектирования, которые могут быть названы походом к проектированию. В настоящее время известно более двух десятков подходов к проектированию ИТ-систем (best practices), традиционно называемых моделями проектирования. Семь из них можно квалифицировать как основные. Следует знать, что, по сути дела, все подходы могут быть поделены на две группы по фундаментальному признаку, представляющему собой ответ на вопрос: «Допустимо или нет использование в качестве приложения «недоделанных» – с неполной функциональностью, информационностью и надёжностью проектируемых систем?»

Если ответ отрицателен, то это нервная группа «жестких» методологий, наиболее известными представителями которых являются:

1) каскадная (Waterfall Model), или «водопад» – это, скорее, теоретический взгляд – основа реальных каскадных подходов;

2) поэтапная с промежуточным контролем – это тот же каскад, но уже практический, – дополненный возможностью (практически всегда неизбежных) возвратов на предыдущие этапы и фазы жизненного цикла системы;

3) V-модель – тоже реальный каскад, отличающийся наличием строгой системы контроля качества работ каждого из этапов – концептуальной пары «Верификация – Валидация» – доказательства верности и разумности реализации требований к системе.

Достаточно очевидно следующее замечание: любая «жесткая» методология – это каскадная модель с теми или иными семантическими дополнениями и вербальными обёртками.

Если ответ на вопрос о применении «неделок» положителен, то это вторая группа – «гибких» методологий, имеющая немало представителей. Среди которых наиболее известны RAD, XP и множество фреймворков,

реализующих «систему ценностей» Agile, определяющих систему ролей, событий и артефактов: Scrum, Kanban, less, SaFE, MSS и прочее (всего около 19 представителей).

В рамках курсового проектирования реализация гибкой методологии в большинстве реальных ситуаций представляется не только нецелесообразной, но и крайне проблематичной в реализации по причине отсутствия у студентов надлежащих знаний и навыков, а также отсутствия заинтересованных заказчиков проекта. В то же время анализ учебно-методической литературы [1-4] показывает, что в настоящее время существует заметная потребность в создании совокупности дополнений к каскадной модели жизненного цикла ИТ-систем, образующих систематический и унифицированный подход, позволяющий сформировать у студентов продуктивные знания, умения и навыки реализации процессов в рамках работ по созданию программно-информационной системы. Созданию одного из таких дополнений посвящается настоящая статья.

Положения предлагаемого подхода к проектированию ИТ-систем

Предлагаемый систематический подход основан на следующих положениях:

1) каскадное представление фаз жизненного цикла информационной системы (ИС);

2) принцип ФИС/АП (акроним от слов Функциональность – Информационность – Структура/Архитектура – Поведение);

3) принцип иерархичности/декомпозиции;

4) представление принимаемых проектных решений в виде системы моделей:

а) модель предметной области;

б) модель требований;

в) модель проектирования:

----- Предварительное проектирование -----

*

в1) модель логической архитектуры (архитектура данных и исполнителей процессов);

----- Детальное проектирование -----

в2) модель поведения:

в21) сценарии использования ИС;

в22) алгоритмы решения задач;

в3) модель физической архитектуры (модель реализации):

в31) модель исходного кода;

в32) модель исполняемого кода;

в33) модель артефактов, поставляемых Заказчику;

в34) модель базы данных;

в35) модель размещения артефактов по устройствам ИС.

Модели группы «в1» создаются на этапе предварительного проектирования. Модели групп «в2» и «в3» создаются на этапе детального проектирования.

Фазы и этапы жизненного цикла информационной системы

В процессе создания (конструирования, разработки) и использования информационной системы можно выделить следующие состояния (фазы) и этапы.

1. Анализ.

1.1. Обследование предметной области, завершаемое созданием модели предметной области.

1.2. Техничко-экономическое обоснование (ТЭО) проекта.

1.3. Изложение требований в форме технического задания (ТЗ) и модели требований.

2. Проектирование.

2.1. Предварительное (архитектурное) проектирование – создание модели архитектуры ИС.

2.2. Окончательное (алгоритмическое) проектирование – создание модели поведения ИС, модели данных ИС и модели реализации ИС.

2.3. Разработка пользовательского интерфейса (GUI).

3. Реализация (программирование, кодирование).

4. Тестирование.

5. Внедрение.

6. Сопровождение.

7. Извлечение из эксплуатации.

Указанные фазы и этапы образуют жизненный цикл информационной системы (ЖЦ ИС) в версии методологии, называемой классической, или каскадной, или водопадной. Эти же фазы и этапы составляют содержание итеративных методологий

Приведенный список содержит указание этапов только для двух первых фаз, представляющих наибольший интерес в рамках изучаемой дисциплины.

Принцип ФИС/АП

Имеются следующие основные (фундаментальные) аспекты рассмотрения ИС как в процессе её изучения, так и в процессе её создания.

1. Функциональность.

2. Информационность.

3. Структура/Архитектура.

4. Поведение.

Функциональность – это совокупность всех работ (задач, функций), выполняемых без использования либо с использованием ИС.

Информационность – это совокупность всех видов информации, потребляемых и создаваемых в предметной области и в ИС в процессе её функционирования, включая исходные данные, результаты вычислений,

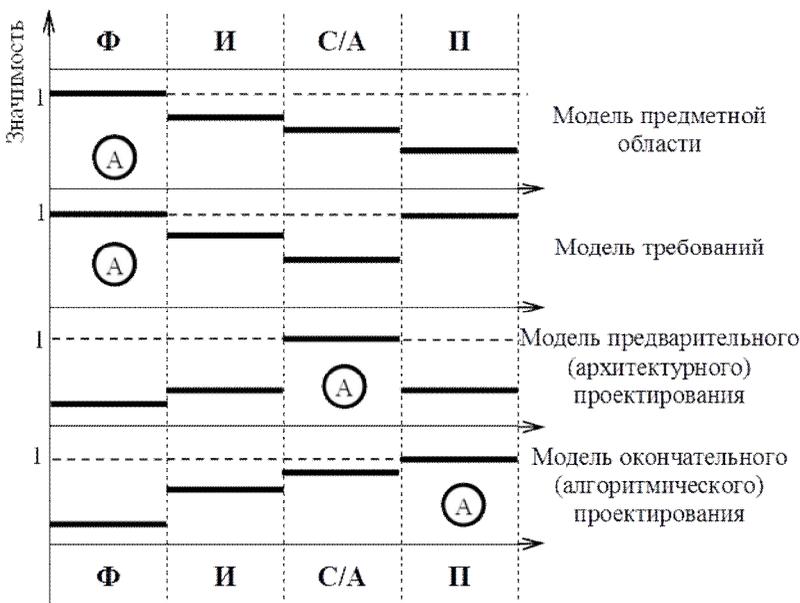
запоминаемые промежуточные значения, знания в форме регламентирующих документов, алгоритмов решения задач, бизнес-правил и прочее.

Структура/Архитектура – это совокупность всех исполнителей функциональности и носителей информации, связей между ними и мест их размещения.

Поведение – это совокупность алгоритмов и сценариев, реализуемых в предметной области без использования либо с использованием ИС. Алгоритмы описывают способы решения содержательно сложных задач.

Принцип ФИС/АП заключается в утверждении, что указанные фундаментальные аспекты рассмотрения ИС имеют место в каждой из фаз, на каждом этапе ЖЦ ИС при создании всех моделей, но с различной степенью значимости (важности, интенсивности).

На рисунке ниже показаны степени значимости фундаментальных аспектов рассмотрения ИС при создании основных моделей ЖЦ.



Ⓐ – признак абсолютной значимости аспекта

Как видно из рисунка, при создании модели предметной области и модели требований наивысший (абсолютный) уровень значимости имеет аспект функциональности, т.е. описание работ (задач, процессов), выполняемых в модели предметной области без использования, а в модели требований – с использованием ИС.

Существенное различие значимостей аспекта поведения в моделях предметной области и требований обусловлено следующим:

1) поведение в предметной области исчерпывается правилами (сценариями) «ручного» выполнения актуальных работ, описание которых может быть весьма поверхностным,

2) в модели требований описывается выполнение актуальных работ с помощью ИС, и поэтому это описание должно быть абсолютно полным и точным.

Абсолютная значимость аспекта структуры/архитектуры в модели предварительного проектирования обусловлена тем, что главная работа предварительного проектирования состоит в том, что окончательно определяются все элементы логической структуры ИС – классы, а при необходимости модули, структуры и иные типы.

Высокая значимость аспекта структуры/архитектуры в модели окончательного проектирования обусловлена тем, что на этапе окончательного проектирования определяются элементы физической структуры ИС – важнейшие артефакты: файлы исходных, объектных и исполняемых кодов, артефакты, поставляемые Заказчику и устройства, в которых размещаются поставляемые артефакты.

Заключение

Основной составляющей предлагаемого систематического подхода к проектированию ИТ-систем является принцип ФИС/АП, дающий ответы на основные вопросы, возникающие при выполнении курсового проекта: 1) какие проектные решения нужно принимать в процессе проектирования системы? (нужно создавать указанные модели проектирования); 2) как нужно составлять модели проектирования? (нужно описывать фундаментальные аспекты рассмотрения систем в контексте текущего этапа проектирования).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Белов В.В., Чистякова В.И. Проектирование информационных систем: учебник – М.: КУРС, 2018. – 400 с. ISBN 978-5-906923-53-0 (КУРС).
2. Орлов С.А. Программная инженерия. Технологии разработки программного обеспечения: Учебник для вузов. – 5-е изд. обновл. и доп. Стандарт третьего поколения. – СПб: Питер, 2016. – 640 с.
3. Избачков Ю., Петров В. и др. Информационные системы. – СПб: Питер, 2010. – 544 с.
4. Ипатова Э.Р., Ипатов Ю.В. Методологии и технологии системного проектирования информационных систем. – М.: Флинта, 2008. – 256 с.

УДК 004.9

Дмитриева Т.А., Рунцо А.А.

ОПИСАНИЕ ТРЕБОВАНИЙ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОЦЕССА ПОЛУЧЕНИЯ ГОТОВОЙ ПРОДУКЦИИ ПОТРЕБИТЕЛЯМИ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

Приведены требования и описание информационной системы для повышения эффективности процесса получения готовой продукции потребителями ЗАО «МПК «КРЗ».

Разрабатываемая информационная система предназначена для автоматизации процесса получения готовой продукции потребителями промышленных предприятий, в том числе ЗАО «МПК «КРЗ». Система на платформе «1С: Предприятие» позволит оформить потребителю заказ на получение готовой продукции у предприятия и получить в определенные сроки свой товар, получая СМС оповещение о готовности заказа, что дает возможность избежать длительных очередей на территории предприятия из автомобилей заказчиков.

Система будет осуществлять процесс заказа продукции от покупателей, оформление соответствующих документов, связанных с заказом, производством и выдачей продукции конечному потребителю, путем СМС рассылки информации о готовности продукции и времени прибытия на территорию предприятия за готовым заказом.

Процесс реализации продукции с помощью разрабатываемой системы представлен следующим образом. Покупатель вводит в поисковую строку своего персонального компьютера официальный сайт предприятия. После этого находит и открывает каталог выпускаемой продукции с прайс-листом, в котором указывается описание продукции, тип продукции, технические характеристики, логистические характеристики, способ применения, сертификация, документация, цена и метраж.

Далее потребитель производит оформление заказа, вводит необходимую информацию о себе. После этого, осуществляется процесс обработки заказа сотрудником отдела продаж предприятия. Сотрудник отдела продаж заполняет соответствующие справочники и документы («Заказ на производство», «Реализация товаров: Накладная», «Счет-фактура, выданный на реализацию») поступившими от заказчика данными. Следующим этапом, данный сотрудник предприятия передает данные о заказе продукции в производственно-технический отдел для его производства. В соответствующих с типом производства в цехах осуществляется производственный процесс по заказу.

После этого сотрудник отдела продаж заполняет документ «Отчет производства за смену» на основании произведенной на предприятии

продукции. После производства товаров по заказу готовая продукция передается на склад предприятия в транспортно-складской комплекс.

Сотрудник отдела продаж составляет документ «Реестр покупателей, приглашенных на отгрузку продукции» и составляет смс рассылку покупателю о готовности продукции и времени прибытия на собственном транспорте за готовой продукцией на территорию предприятия. После этого покупатель получает смс о готовности своего заказа и приезжает на территорию предприятия, без ожидания забирает товары и покидает завод.

Работа информационной системы происходит следующим образом. После запуска программы открывается форма авторизации. Пользователю (сотруднику отдела продаж предприятия) необходимо ввести логин и пароль для входа в систему. При неправильном вводе логина или пароля будет введено сообщение об ошибке. В случае успешной авторизации открывается основное окно программы.

Далее пользователь вносит соответствующие данные от заказчика продукции в представленные в программе справочники по мере необходимости их заполнения: документ, изготовитель, контрагент, менеджер, номенклатура, номенклатурная группа, номер счета, операция, организация, подразделения затрат, процент НДС, склад, состояние заказа, счет затрат, счет учета.

После этого сотрудник предприятия заполняет документы.

1. Заказ на производство.
2. Отчет производства за смену.
3. Реализация товаров: Накладная.
4. Счет-фактура, выданный на реализацию.
5. Реестр покупателей, приглашенных на отгрузку продукции.

После заполнения соответствующих справочников и документов информацией о поступивших заказах на производство продукции предприятия, пользователь составляет отчеты на основе поступивших данных.

После производства продукции и поступления ее на склад транспортно-складского комплекса, сотрудник отдела продаж предприятия заходит во вкладку «Прочее», выбирает «Сервис», нажимает «СМС рассылка». Вводит в представленные поля данные и нажимает на кнопку «Отправить смс».

Разрабатываемая система является актуальной и практически значимой для промышленного предприятия, так как позволяет уменьшить большие очереди среди автомобилей заказчиков, что приводит к улучшению конкурентных позиций предприятия. Все это приводит к повышению числа клиентов предприятия, исходя из того, что заказчикам сотрудники завода предлагают улучшенные условия для процесса заказа продукции и ее получения.

При помощи данной информационной системы клиенты предприятия перестанут уходить к организациям-конкурентам, что приведет к увеличению прибыли завода и повышение конкурентоспособности по фактору «обслуживание потребителей».

Для разработки информационной системы был использован программный продукт компании 1С «1С: Предприятие» версии 8.3, предназначенный для автоматизации деятельности на предприятии [1]. В системе «1С: Предприятие 8.3» были созданы константы: «Название организации», «ИНН», «Генеральный директор», «Дата основания» и «Юридический адрес», которые содержат неизменяемую информацию о предприятии. Константы: «Название организации», «Юридический адрес», «Генеральный директор» имеют тип – строка, «ИНН» – число, «Дата основания» – дата. Справочники: «Документ», «Изготовитель», «Контрагент», «Менеджер», «Номенклатура», «Номенклатурная группа», «Номер счета», «Операция», «Организация», «Подразделения затрат», «Процент НДС», «Склад», «Счет затрат», «Счет учета» «Состояние заказа», «Операция». Перечисление «Составлен» с данными: на бумажном носителе, в электронном виде. Также были созданы документы, перечисленные выше.

Для различных расчетов в модулях был прописан программный код. Рассмотрим один из фрагментов более подробно. Для того чтобы пользователь смог отправить СМС покупателю о готовности заказа и времени приезда на территорию предприятия за готовой продукцией была создана обработка «СМС рассылка» со следующими реквизитами: логин (тип – строка, длина – 10), пароль (тип – число, длина – 10), номер адресата (тип – число, длина – 12), текст (тип – строка, длина – 100). В форме обработки «СМС рассылка» была создана команда «Отправить СМС». Для работы кнопки «Отправить СМС» и получения сообщения заказчиками продукции был прописан следующий код.

```
ИмяФайлаОтвета = ПолучитьИмяВременногоФайла («txt»);
Соединение = Новый HTTPСоединение («api.avisosms.ru»);
СтрокаПодключения = «sms/get/?username=» + СокрЛП (Логин) +
«&password=» + СокрЛП (Пароль) + «&destination_address=» + СокрЛП (НомерАдресата) +
«&source_address=SENDER&message=» + СокрЛП (Текст);
Соединение.Получить (СтрокаПодключения, ИмяФайлаОтвета);
ФайлОтвета = Новый Файл (ИмяФайлаОтвета);
Если ФайлОтвета.Существует () Тогда
    ТекстОтвета = Новый ТекстовыйДокумент ();
    ТекстОтвета.Прочитать (ИмяФайлаОтвета);
    Если ТекстОтвета.КоличествоСтрок () > 0
        Тогда
            ОтветСервера = ТекстОтвета.ПолучитьТекст ();
            Если
                Найти (ОтветСервера, «OK_Operation_Completed») > 0
                    Тогда
                        Сообщить («Сообщение успешно отправлено
                            на номер» + СокрЛП (НомерАдресата));
            Иначе
                Сообщить («Проблемы с отправкой СМС на
                    номер » + СокрЛП (НомерАдресата)
                    + «. Ответ сервера: » + ОтветСервера);
            КонецЕсли;
```

КонецЕсли;
 УдалитьФайлы (ИмяФайлаОтвета) ;
 КонецЕсли;

Алгоритм работы данного программного кода представлен на рисунке.

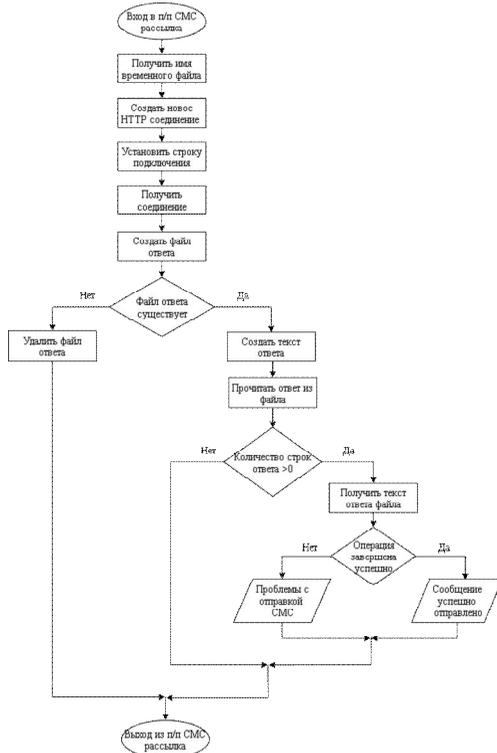


Схема алгоритма программного кода по СМС рассылке

Для того чтобы хранились суммы по заказам покупателей был создан регистр сведений «Сумма купленной продукции по заказам», с его помощью пользователь сможет быстро найти необходимую сумму по заказчикам и подставить ее в поле «Сумма» документа «Счет-фактура, выданный на реализацию». Регистр сведений «Сумма купленной продукции по заказам» с периодичностью – в пределах дня, режим записи – независимый, основной отбор по периоду. Измерения: контрагент (тип – Справочник Ссылка.Контрагент), ресурсы: сумма (тип – число, длина – 10, точность – 2, неотрицательное).

Помимо этого было создано несколько отчетов, позволяющих проанализировать данные в различных разрезах, например, был создан отчет «Сумма по договору поставки», который предназначен для обработки информации и предоставлению пользователю в виде табличного документа,

который может быть распечатан или сохранен как внешний файл. При формировании данного отчета пользователь сможет увидеть вид продукции, сумму, контрагента и договор поставки по произведенным заказам.

Разработанная информационная система полностью работоспособна и отвечает все предъявляемым к ней требованиям.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Заика А.А. Разработка прикладных решений для платформы 1С. Предприятие 8 в режиме «Управляемое приложение» [Электронный ресурс] / А.А. Заика. – 2-е изд. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 238 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/73721.html>.

УДК 005.519.8

Шурыгина О.В., Цуканова Н.И.

РАЗРАБОТКА НОУТБУКА В СРЕДЕ GOOGLE COLAB ПО ТЕМЕ «КЛАССИФИКАЦИЯ ТЕКСТОВЫХ СООБЩЕНИЙ С ПОМОЩЬЮ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ»

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматривается ноутбук, описывающий методику классификации текстовых сообщений с помощью глубоких нейронных сетей, разработанных на языке Python

В связи с пандемией коронавируса все образование, в том числе высшее, было переведено на дистанционное обучение. Это потребовало внедрения новых технологий в обучение. Широкое применение нашли облачные технологии, в частности Google Colaboratory, виртуальная среда, предназначенная для обучения глубоким нейронным сетям, а также новые методические инструменты – ноутбуки.

Начнем с определения: что такое «**notebook**» (блокнот)? Блокнот это единый документ, который объединяет код на некотором языке программирования, текстовую, графическую и видео информацию, предназначенную для изложения и пояснения определенной темы. Этот единый документ может быть выполнен в браузере, благодаря чему он становится доступным интерактивным динамичным учебником и может быть использован как на лекциях, так и при самостоятельном изучении материала, а также при быстрой интерактивной разработке программных продуктов. Все что нужно для создания и использования таких учебных ноутбуков, так это программное обеспечение, способное воспроизводить ноутбуки. В настоящее время наиболее известными средами для разработки и воспроизведения ноутбуков являются *Jupyter Notebook* и *Google Colab*.

Используя Google Colab, можно:

– улучшить свои навыки программирования на языке Python;

– развивать глубокое обучение нейросетей с использованием популярных библиотек, таких как Keras, TensorFlow PyTorch, и OpenCV.

Наиболее важной особенностью, которая отличает Colab от других бесплатных облачных сервисов, является то, что Colab предоставляет графический процессор GPU и это абсолютно бесплатная услуга. Благодаря графическому процессору обучение глубоких нейронных сетей становится возможным по времени.

В настоящее время большое внимание уделяется вопросам разработки нейронных сетей для решения практических задач. Одной из таких задач является интеллектуальная обработка текстовой информации [2], которая, в частности, включает такие задачи, как классификация текста, определение его тональности, кластеризация текста [1] и другие.

В настоящей статье рассматривается задача определения источника статьи по ее заголовку и выполнена она в виде ноутбука, содержащего текстовые ячейки и ячейки кода. В тексте выделены и пояснены основные этапы обработки текста.

Ноутбук как учебник по применению нейронных сетей к решению практических задач может создаваться в соответствии со следующим планом:

- постановка задачи;
- описание, понимание и исследование набора данных, предназначенного для обучения нейронной сети;
- предобработка набора данных (особенно текста);
- формирование нейронной сети, определение ее архитектуры и гиперпараметров;
- обучение нейронной сети;
- оценка качества нейронной сети с графической иллюстрацией результатов оценки;
- применение обученной нейронной сети к новым данным.

В каждом из перечисленных пунктов сначала следует поясняющий текст, затем код, реализующий те процессы, которые описаны в тексте.

При работе с текстом очень важен этап его предобработки. Так как нейронная сеть работает только с числовой информацией, то текст надо преобразовать в числа. В данной работе предобработка текста включала следующие шаги: разделение текста на токены (слова), рассматривались различные виды кодирования токенов (кодирование частотой использования, one hot кодирование, кодирование плотными векторами (Embedding)), усечение текста до определенных размеров. Каждый из этих шагов описан и разъясняется в текстовых полях.

В работе рассмотрены две конфигурации нейронной сети: рекуррентная нейронная сеть, основанная на ячейках LSTM, и одномерная сверточная нейронная сеть [2]. Оценка качества каждой из этих архитектур показала, что большую точность обеспечивает сеть на ячейках LSTM. На новых данных также были получены неплохие результаты.

Одним из самых важных и сложных этапов решения задач классификации текстов является формирование обучающей выборки. В интернете можно много найти текстовой информации, однако собрать нужные примеры в выборку и пометить каждый пример (указать класс, к которому он принадлежит) оказывается совсем не простой задачей. В данной работе рассматриваются два способа формирования выборки: в первом англоязычная информация извлекается с помощью запроса к базе данных *Hacker News* на веб-сервисе *bigQuery*, во втором рассматриваются русскоязычные статьи с сайта *CyberLeninka*.

Для формирования русско-язычного набора данных написана программа на языке *Java*, с помощью которой был получен набор (более 700000 примеров) с заголовками статей и указанием источника статьи (всего 5 источников). Этот набор данных был сохранен в файле *Articles.csv*. В качестве источников статей были выбраны рубрики:

- *EconomicsAndBusiness*
- *ClinicalMedicine*
- *EducationalSciences*
- *LanguagesAndLiterature*
- *HistoryAndArchaeology*

Для каждого источника считывалась информация с 7000 страниц (по 20 статей на странице).

Программа обрабатывает 1 страницу в секунду, так что создание файла занимает несколько часов. После создания файла необходимо открыть его текстовым редактором и в начало дописать заголовок файла *source, title*.

Для более удобного использования программы, был написан графический интерфейс.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гиголаев А.В., Цуканова Н.И. Анализ семантической близости слов с помощью карт Кохонена. Вестник РГРТУ. 2018. №2. – С.85-91.
2. Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. – Спб.: Питер, 2019. – 368с.

УДК 658.011.56

Крошилина А.А.,
Благодаров Е.А., Благодарова Т.А.

ПРИМЕНЕНИЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

В статье приводится обоснование применения мобильных приложений в образовательном процессе. Обозначен круг задач, которые необходимо решить для разработки мобильного приложения. Проведен анализ существующих мобильных приложений и сформированы требования к системам обучения и контроля в образовательном процессе.

Интенсивные процессы информатизации общества привели к массовому оснащению образовательных учреждений средствами информационных и коммуникационных технологий. Данные средства превратились в новые педагогические инструменты, позволяющие помочь в решении проблемы снижения качества обучения, являющейся одной из основных проблем современного образования. Как показывает практика, оснащение образовательных учреждений техническими средствами обучения без обновления содержания, методов и форм обучения не дает ощутимых результатов [1].

Анализ мировых тенденций демонстрирует жизненную остроту применения в образовательной деятельности мобильных приложений для решения различных педагогических задач, организации удаленного доступа к общесетевым и специализированным ресурсам и сервисам учебных заведений. Своевременность применения мобильных технологий в образовательной среде обусловлена следующими предпосылками: высокий уровень и динамика распространения мобильных устройств (не редкость, когда один пользователь является владельцем двух и более устройств), устойчивый интерес к их применению, возможностью превратить в медиаконтент и сопутствующее содержание в инфраструктуру образовательного и научно-исследовательского пространства [2].

Важную роль в повышении качества обучения играет систематический и оперативный контроль знаний обучающихся, который является обязательным этапом учебного процесса. Контроль выполняет функции управления процессом усвоения знаний, мотивации и формирования познавательного интереса, что возможно при соблюдении требований объективности, открытости, системности и оперативности [3]. Эффективность контроля знаний зависит от регулярности и систематичности его проведения. Оперативный контроль является основным инструментом управления познавательной деятельностью обучающихся, он позволяет формировать у обучающихся важное понимание необходимости сопро-

вождения любого обучения проверкой, которую необходимо проводить регулярно.

Таким образом, можно сформировать набор задач, которые необходимо решить для разработки мобильного приложения в образовательном процессе:

- сбор и обработка информации об организации контроля знаний в мобильных приложениях;
- разработка анкет, анкетирование и анализ полученной информации;
- исследование современных методик организации контроля знаний по изучаемым дисциплинам;
- обзор существующих мобильных приложений, выявление их достоинств и недостатков;
- изучение возможностей мобильной платформы для разработки в рамках организации контроля знаний по учебным дисциплинам;
- проектирование набора функционала мобильного приложения в рамках разрабатываемой структуры;
- разработка тестирующих заданий для мобильного приложения;
- отладка мобильного приложения;
- продвижение мобильного приложения в сервисе Google Play.

Разрабатываемое приложение должно отвечать следующим требованиям: давать возможность осуществлять контроль полученных знаний; предоставлять методики решения задач по соответствующим разделам; обладать эргономичным интерфейсом.

В настоящее время существует ряд мобильных приложений, среди которых наиболее популярны следующие: Vernier Video Physics, «Формулы по физике для ЕГЭ», «ЕГЭ физика», «Мобильная физика». Анализ указанных приложений позволил выделить их достоинства и недостатки.

Таким образом, разрабатываемое мобильное приложение должно отвечать следующим требованиям: кроссплатформенность; содержать теоретический курс предмета; содержать банк заданий по изученному материалу; осуществлять промежуточный контроль знаний; хранить историю промежуточного контроля для анализа.

Основными достоинствами подобных средств являются доступность, простота в использовании, мобильность и технологичность. Мобильные приложения в образовательном процессе можно рекомендовать к использованию для организации и проведения оперативной проверки знаний обучающихся.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бобровская Л.Н., Данильчук Е.В., Куликова Н.Ю. Методические особенности использования интерактивных средств обучения для реше-

ния дидактических задач учителя на уроках информатики. // Информатика и образование. – 2013. – № 2 (241) – С. 76-78.

2. Сергеев А.Н. Профессиональная подготовка будущих учителей в контексте обучения в сетевых сообществах Интернета. // Известия Волгоградского государственного педагогического университета. Серия «педагогические науки»: научный журнал. – 2010. – №1.

3. Карпенко А.П., Домников А.С., Белоус В.В. Тестовый метод контроля качества обучения и критерии качества образовательных тестов. Обзор. // Наука и образование. – 2011. – №4. [Электронный ресурс]. URL: <http://technomag.edu.ru/doc/184741.html> (дата обращения: 10.03.2020).

УДК 658.5

Белов В.В., Моудио Моудио Фернанд Елисе

СООТНОШЕНИЕ МЕЖДУ ФУНКЦИОНАЛЬНЫМ И ПРОЦЕССНЫМ ПОДХОДАМИ К УПРАВЛЕНИЮ БИЗНЕС-ПРОЦЕССАМИ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

Показано, что противопоставление функционального и процессного подходов к управлению бизнес-процессами, по сути, является противопоставлением внутренних и внешних бизнес-процессов.

В многочисленных обсуждениях различий между функциональным и процессным подходами к управлению бизнес-процессами приводятся различного рода признаки различий, которые в массе своей являются несущественными, непринципиальными, а иногда и вовсе надуманными.

Самым оригинальным, фактически общепринятым выводом рассуждений о функциональном и процессном управлении явилось заключение об отсутствии противоречий и взаимодополняемости функционального и процессного управления бизнес-процессами. Практически каждую публикацию по теме функциональности/процессности завершает вывод, совпадающий по смыслу с одним из следующих примеров.

1. «Таким образом, можно сделать вывод о том, что противопоставление процессного и функционального подходов принципиально неверно, поскольку функции и процессы, являясь равнозначными понятиями управленческой деятельности, не могут существовать в отрыве друг от друга. Противоречий между двумя подходами не существует – они дополняют друг друга и их следует интегрировать» [1].

2. «В то же время оба подхода не являются противопоставлением друг другу. При применении сразу двух подходов предприятие будет иметь систему взаимосвязанных процессов, объединяющих схожие функции в рамках различных бизнес-процессов. Для достижения наилучших результатов предприятию целесообразней использовать эти подходы параллельно» [2].

3. «Противопоставление процессного и функционального подходов принципиально неверно. В практической работе менеджера, почти всегда,

используется суперпозиция этих подходов к управлению. Функции также, как и процессы, являются равнозначными понятиями управленческой деятельности и не могут существовать в отрыве друг от друга. Например, инструментом для реализации функционального подхода к управлению является система бюджетирования (делегирование полномочий и ответственности через систему центров финансовой ответственности). В свою очередь, управление бюджетированием требует применения процессного подхода к управлению» [3].

Подобного рода выводы принципиально некорректны, хотя и сделаны с использованием обращений к понятиям функции и процесса.

Являются ли равнозначными понятиями управленческой деятельности функции и процессы?

Уточним понятия функции и процесса для упрощения понимания ассоциированных с ними понятий и исключения коллизий в выводах.

Функция (F) – это понятие, семантика (смысл) которого представляет собой пересечение семантик понятий *Работа*, *Задача*, *Операция*, *Деятельность*, *Деятельность*:

$$\text{Sem}(F) = \text{Sem}(\text{Работа}) \cap \text{Sem}(\text{Задача}) \cap \text{Sem}(\text{Операция}) \cap \text{Sem}(\text{Деятельность}) \cap \text{Sem}(\text{Деятельность}) \quad (1)$$

Указанное утверждение означает, что смысл понятия Функция состоит из того, что является общим в смыслах, перечисленных в правой части (1) понятий. По сути дела, правая часть (1) – это то, что обычно называют перечислением синонимов, но отличается указанием на тот факт, что перечисляются не синонимы, а близкие по смыслу понятия, имеющие не только общие, но различные составляющие в своих семантиках.

Назовем инфоструктурой, или кратко Info, сведения, характеризующие понятие, – его назначение, параметры, признаки и т.п. Обозначим инфоструктуру понятия Концепт так: $\text{Info}(\text{Концепт})$. Важен факт: инфоструктура – это информационное начало, предназначенное для выражения (представления) семантики. Примером инфоструктуры является иерархическая система параметров некоторого технического агрегата, например, автомобиля (приводимая ниже система параметров не полна и носит чисто иллюстративный характер):

ПараметрыАвто = Технические \cap Экономические \cap Эргономические;
 Технические = Время разгона до 100 км/час \cap Расход топлива на 100 км;
 Экономические = Цена покупки \cap Цена владения;
 Эргономические = Цвет кузова \cap Форма кузова \cap Материал отделки салона.

Концевые элементы инфоструктуры (листья дерева иерархии) представляют собой названия конкретных параметров. Обозначим их совокупность InfoS. Верхние элементы инфоструктуры – это названия групп параметров. Обозначим их совокупность InfoG.

В программных системах компьютерной обработки информации элементам InfoS для краткости записи, как правило, ставятся в однозначное соответствие короткие цепочки символов – идентификаторы (обозначения, имена), совокупность которых обозначим InfoN. Эти идентификаторы в программах символизируют участки памяти, в которых размещаются конкретные значения элементов InfoS. Совокупность этих значений обозначим InfoV. Таким образом, существует цепочка однозначных отображений: $\text{InfoS} \rightarrow \text{InfoN} \rightarrow \text{InfoV}$.

В языках программирования, СУБД и системах моделирования элементы InfoN представляют собой переменных типа структура (запись) или класс.

Инфоструктура понятия *Функция* состоит из инфоструктур *Цели* и *Процесса достижения цели*:

$$\text{Info}(F) = \text{Info}(\text{Цель}) \cup \text{Info}(\text{Процесс достижения цели}) .$$

$$\text{Причем } \text{Sem}(\text{Цель}) = \text{Sem}(\text{Требуемый результат}) ;$$

$$\text{Sem}(\text{Процесс достижения цели}) = \text{Sem}(\text{Алгоритм реализации функции}) \cup \text{Sem}(\text{Сценарий реализации функции}) .$$

В привычной кортежной нотации, отражающей информационное наполнение сущностей предметной области на уровне InfoN, можно записать кратко так:

$$F = \langle T, P \rangle ,$$

где F, T, P – соответственно инфоструктуры понятий *Функция*, *Цель* и *Процесс достижения цели*, или кратко – *Процесс*.

Алгоритм и сценарий – понятия семантически схожие: и то, и другое представляют собой последовательности реализации некоторых функций. Отличие состоит только в том, что сценарий необязательно должен быть детерминированным, массовым и результатным. Это означает следующее:

$$\text{Sem}(P) = \text{Sem}(F_1 \rightarrow F_2 \rightarrow \dots \rightarrow F_n) ,$$

где n – число шагов процесса

Одновременно с этим: $\text{Info}(P) = \text{Info}(F_1) \rightarrow \text{Info}(F_2) \rightarrow \dots \rightarrow \text{Info}(F_n)$, или кратко: $P = F_1 \rightarrow F_2 \rightarrow \dots \rightarrow F_n$.

$$\text{А это означает: } F = \langle T, F_1 \rightarrow F_2 \rightarrow \dots \rightarrow F_n \rangle .$$

В общем случае функции могут декомпозироваться – представляться в виде совокупности алгоритмически более простых функций (подфункций):

$$F_i = \langle T_i, F_{i,1} \rightarrow F_{i,2} \rightarrow \dots \rightarrow F_{i,n_i} \rangle , \quad i = \overline{1, n} .$$

Итак, функция – это, во-первых, некоторая цель – формулировка ответа на вопрос: «Что нужно получить?» – и, во-вторых, это процесс – формулировка ответа на вопрос: «Как это можно получить?» При этом ответ на второй вопрос имеет форму последовательности более простых (по сравнению с рассматриваемой функцией) функций.

Изложенное означает, в утверждении: «... Функции и процессы, являясь равнозначными понятиями управленческой деятельности, не могут существовать в отрыве друг от друга» [1] безусловно верной является только вторая часть – функции и процессы, действительно, не могут существовать в отрыве друг от друга, поскольку процессы – это способ реализации функций, причем рекуррентным образом – в виде последовательности функций. Образно говоря, процессы – это форма существования функций. Утверждение же об их равнозначности становится верным, только, если под равнозначностью понимать равную важность (значимость) этих понятий, но не их смысловую эквивалентность, как это определяют словари [4].

Верно ли противопоставление процессного и функционального подходов?

Сформировавшееся и распространенное в настоящее время утверждение: «Противопоставление процессного и функционального подходов принципиально неверно», «Противоречий между двумя подходами [функциональным и процессным] не существует – они дополняют друг друга и их следует интегрировать» является двойственным, неоднозначным – его верность/неверность зависит от толкования сути сопоставляемых подходов.

Если исходить из указанного выше соотношения между функциями и процессами, то противопоставление функционального и процессного подходов не то, чтобы неверно, оно просто лишено смысла. Следовательно, суть проблемы не в формальном соотношении между функциями и процессами, а в чем-то ином.

Исторически противопоставление и противоречие между функциональным и процессным подходами явилось «притчей во языцех», возникшей как продукция многочисленных публикаций, посвященных вопросам внедрения методологии структурного анализа и проектирования» (SADT – Structural Analysis and Design Technique), предложенной Дугласом Т. Рос-сом в 1969 г., и далее методологии и системы стандартов IDEF (Icam DEFinition, или Integrated DEFinition). IDEF создана в рамках ICAM (Integrated Computer-Aided Manufacturing) – программы ВВС США, нацеленной на компьютеризацию промышленности. Активное применение IDEF обусловило создание базовых идей популярнейшего в настоящее время понятия – BPR (Business Process Reengineering). Именно цепочка SADT → IDEF → BPR породила «конфликт» между функциональным и процессным подходами.

Классики теории управления бизнес-процессами, – Анри Файоль (1841-1925) – автор концепции процессного подхода, Джеймс Муни (1884-1957) – разработчик универсальных принципов организации, Алан Рейли (1869-1947) – исследователь принципа делегирования ответственности, – ничего не говорили о функциональном подходе, объектом их рассмотрения были процессы, естественным образом, состоящие из функций. Даже сегодня последователи указанных авторов говорят только о трех подходах к управлению: 1) процессный; 2) системный; 3) ситуационный.

«Функциональный подход» как термин создали разработчики и популяризаторы методологий SADT → IDEF → BPR. Образно говоря, из многочисленных публикаций семидесятих годов прошлого века следовало: «То, что было ДО сего дня – это функциональный подход, а для повышения эффективности бизнеса (в том числе и производства) необходимо применять НО-ВЫЙ – процессный подход». Начиная с этого момента из различий между функциональным и процессным подходами, удалено соотношение между понятиями функции и процесса. В основу различий положена *организация управления* бизнес-процессом. При этом квинтэссенцией различий в организации управления являются: 1) принципы контроля качества результатов выполнения функций процесса; 2) процедура передачи результатов от одной функции другой – последующей функции процесса.

В функциональном подходе контроль качества выполнения текущей функции и процедура передачи полученных результатов последующей функции для продолжения бизнес-процесса осуществляется по внутрифункциональным критериям – без прогноза качества результатов выполнения последней функции процесса, выдающей готовую продукцию, поставляемую потребителю, т.е. не оценивается ожидаемое качество выполнения всего процесса в целом. Например, в пошивочном ателье в процессе реализации функции «Раскрой» в силу субъективных или объективных причин (поломка измерительного инструмента) могла быть допущена ошибка – рукава пиджака оказались разной длины. Во время передачи продукта первой функции в последующую функцию «Пошив рукавов пиджака» данная ошибка могла оказаться незамеченной, поскольку контроль длины рукавов является обязательной операцией функции «Раскрой», и поэтому считается, безусловно, выполненной, и причем правильно. Функция «Пошив рукавов пиджака» может быть выполнена очень качественно (хорошей строчкой, прочными нитками), но если внутри этой функции контролировать только качество строчки и прочность шитва, то брак на выходе процесса изготовления костюма гарантирован. Добавление длины рукавов в список контролируемых параметров на этапе «Пошив рукавов пиджака» (повтор параметра предыдущей функции) – это мероприятие, нацеленное на обеспечение качества результата выполнения финальной функции, это признак перехода к процессному подходу, главной особенностью которого является применение при контроле качества внутренних процессов, – реализующих отдельные функции, – системы критериев, обеспечивающих высокое качество сквозного (end-to-end) процесса, на входе которого находится сырье, материалы и прочее, а на выходе – готовая продукция.

Таким образом, можно сделать вывод: функциональный и процессный подходы кардинально отличаются системой показателей качества реализации внутренних (end-to-process, process-to-process, process-to-end) процессов, реализующих последовательные функции сквозного (end-to-end) процесса, и способом осуществления процедур сдачи/приема резуль-

татов выполнения внутренних процессов типа process-to-process. Образно говоря, на знамени «функционального» подхода написано: «Даешь высокое качество внутренних процессов!», а на знамени «процессного» подхода: «Даешь высокое качество сквозных процессов!»

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. http://www.rusnauka.com/13_NPN_2010/Economics/65418.doc.htm
2. <https://scienceforum.ru/2014/article/2014007689>
3. <https://conf.mirea.ru/CD2015/pdf/p1/34.pdf>
4. https://syn_tesaurus.academic.ru

УДК 004.9

Карпушкина Д.А.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

Информационная система предназначена для более эффективного процесса получения готовой продукции заказчиками промышленных предприятий

Хотя информационные системы являются обычным программным продуктом, они имеют ряд существенных отличий от стандартных прикладных программ и систем. В зависимости от предметной области информационные системы могут весьма значительно различаться по своим функциям, архитектуре, реализации. Однако можно выделить ряд свойств, которые являются общими.

1. Информационные системы предназначены для сбора, хранения и обработки информации, поэтому в основе любой из них лежит среда хранения и доступа к данным.

2. Информационные системы ориентированы на конечного пользователя, не обладающего высокой квалификацией в области вычислительной техники.

Таким образом, при разработке информационной системы приходится решать две основные задачи – разработка базы данных для хранения информации и разработка графического интерфейса пользователя клиентских приложений.

Система управления базой данных (СУБД) является неотъемлемой частью любой информационной системы. Тип используемой СУБД обычно определяется масштабом информационной системы.

Первым шагом в проектировании информационной системы является формальное описание предметной области, построение полных и функциональных непротиворечивых и информационных моделей. Это логически сложная, трудоёмкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов. Области применения и реализации информационных систем

Бухгалтерский учёт – классическая и наиболее часто реализуемая на сегодняшний день область применения информационных технологий.

Управление финансовыми потоками – внедрение информационных технологий обусловлено критичностью этой области управления предприятия к ошибкам. Неправильно построив систему расчётов с поставщиками и потребителями, можно спровоцировать кризис наличности даже при налаженной сети закупки, сбыта и хорошем маркетинге.

Управление складом, ассортиментом, закупками – заморозить оборотные средства в чрезмерном складском запасе – самый простой способ сделать любое предприятие потенциальным инвалидом. Можно посмотреть перспективный товар, вовремя не вложить в него деньги.

Управление производственным процессом. Автоматизированное решение подобной задачи даёт возможность правильно планировать, учитывать затраты, проводить техническую подготовку производства, оперативно управлять процессом выпуска продукции в соответствии с производственной программой.

Управление маркетингом. Подразумевается сбор и анализ данных о фирмах конкурентах, их продукции и ценовой политике, а также моделирование параметров внешнего окружения для определения оптимального уровня цен, прогнозирования прибыли и планирования рекламных кампаний.

Документооборот. Хорошо отлаженная система учётного документооборота отражает реально происходящую на предприятии текущую производственную деятельность и даёт управленцам возможность воздействовать на неё.

Оперативное управление предприятием. Информационная система, решающая задачи оперативного управления предприятием, строится на основе базы данных, в которой фиксируется вся возможная информация о предприятии.

Предоставление информации о фирме. Активное развитие Интернета привело к необходимости создания корпоративных серверов для предоставления различного рода информации о предприятии. Практически каждое уважающее себя предприятие сейчас имеет свой веб-сервер. Кроме того, использование веб-технологий открывает широкие перспективы для электронной коммерции и обслуживания покупателей через Интернет.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Избачков Ю., Петров В. Информационные системы. – М.: Издательство «Юнион Принт», 2017. – 16 с.
2. Колесник А.П. Компьютерные системы в управлении финансами [Электронный ресурс]. – Режим доступа: https://studbooks.net/2270134/informatika/analiz_suschestvuyuschih_resheniy_oblasti_realizatsii_gotovoy_produktsii. Дата обращения 22.05.2020 г.

УДК 004

Новинская Д.В, Проказникова Е.Н.
ОСОБЕННОСТИ АЛГОРИТМОВ
ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В статье рассматривается понятие алгоритма интеллектуального анализа данных и описываются некоторые особенности этих алгоритмов.

Интеллектуальный анализ данных (data mining) – это исследование данных, использующее методы искусственного интеллекта и ориентированное на придание системе свойств искусственного интеллекта для принятия решений в различных сферах человеческой деятельности. Наиболее известная реализация технологий Data Mining – это поисковые системы в Интернете. [1]

Современные системы добычи данных используют основанные на методах искусственного интеллекта средства представления и интерпретации, что и позволяет обнаруживать не очевидную, но весьма ценную информацию. [2]

Алгоритм интеллектуального анализа данных – это набор вычислений и методов, представляющий собой механизм, который создаёт модель интеллектуального анализа данных. Чтобы создать модель, алгоритм анализирует предоставленные данные, осуществляя поиск закономерностей, затем использует результаты этого анализа для выбора оптимальных параметров создания модели интеллектуального анализа данных, после чего эти параметры применяются ко всему набору данных, чтобы выявить подходящие для эксплуатации последовательности и получить подробную статистику.

Создаваемая алгоритмом модель интеллектуального анализа данных, может иметь разнообразные формы:

- математическую модель, составляющую прогнозы продаж;
- дерево принятия решений, которое предсказывает результат и отображает влияние на него различных критериев;
- набор кластеров, описывающих связи вариантов в наборе данных;
- набор правил, показывающих группирование продуктов в транзакции, а также вероятности одновременной покупки продуктов.

Несколько наиболее эффективных алгоритмов data mining:

C4.5 – строит классификатор в форме дерева решений. Для этого ему необходимо передать набор уже классифицированных данных. Большим достоинством этого метода является простая интерпретация.

Метод k -средних – это популярная техника кластерного анализа для исследования набора данных. Он создаёт группы с наиболее однородными членами из набора объектов. Его достоинства: высокая скорость выполнения и простота.

Метод опорных векторов – классифицирует данные по двум классам, используя гиперплоскость. Выполняет те же операции, что и С4.5, но не использует деревья решений. Одной из его слабых сторон является плохая интерпретируемость.

Алгоритм Argiogi ищет ассоциативные правила. Применяется данный алгоритм по отношению к базам данных, которые имеют огромное число транзакций. Алгоритм Argiogi легко реализуется и имеет множество модификаций.

EM-алгоритм – алгоритм максимизации ожиданий для обнаружения знаний. Этот алгоритм прост в исполнении и может делать предположения, относительно отсутствующих данных.

PageRank – это алгоритм ссылочного ранжирования, разработанный для определения относительной важности объекта, связанного с сетью объектов. Главное достоинство этого метода – надёжность.

Проанализировав научные работы других авторов, можно прийти к выводу, что не существует уникального алгоритма, для решения задач. Каждый алгоритм имеет свои плюсы и минусы, которые следует учитывать, при реализации конкретной цели.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мусаев А. Интеллектуальный анализ данных: учебное пособие. А.А. Мусаев – СПб.: СПбГТИ(ТУ), 2018. – 56 с.
2. Михеев А. Алгоритмы интеллектуального анализа данных в информационной системе поддержки удалённого эксперимента – 2013. – № 2 – С. 86.

УДК 004.9

Дмитриева Т.А., Рябичев А.А.

НЕОБХОДИМОСТЬ РАЗРАБОТКИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ ЭЛЕКТРОННЫХ ТЕХНИЧЕСКИХ РУКОВОДСТВ ПО ОБСЛУЖИВАНИЮ И РЕМОНТУ ИЗДЕЛИЙ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматривается интерфейс разработанного программного обеспечения для образовательной платформы, включающей интерактивные электронные технические руководства (ИЭТР) по обслуживанию и ремонту изделий, обучение и тестирование пользователя, для использования на АО «Рязанский радиозавод».

В статье [1] была рассмотрена проблема обучения и тестирования пользователей технических руководств по обслуживанию и ремонту изделий, сформулированы требования, выдвигаемые к разрабатываемому программному обеспечению для решения выявленной проблемы, определена практическая ценность данного прикладного программного обеспечения для АО «Рязанский радиозавод».

Рассмотрим интерфейс разработанной системы. После запуска программы на экране отображается страница входа в приложение. Оператор вводит логин и пароль. После успешной авторизации пользователя открывается страница в соответствии с его правами. Это может быть страница обычного пользователя модератора, обучающего, администратора. Рассмотрим страницу пользователя (рис. 1).

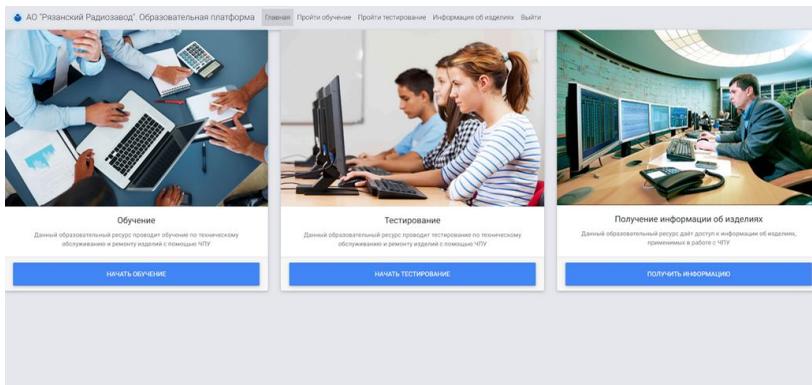


Рис. 1. Страница пользователя

После перехода на страницу обычного пользователя будут доступны вкладки с обучением и тестированием пользователя, а также с информацией о изделиях. Рассмотрим страницу обучения пользователя с открытой вкладкой «Арматура» (рис. 2).

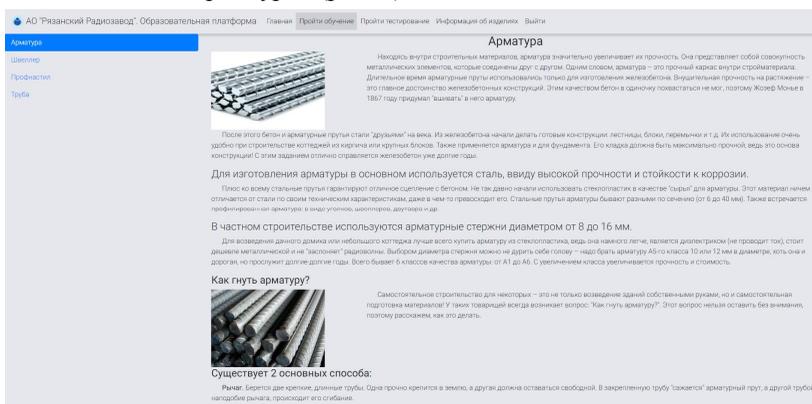


Рис. 2. Страница обучения пользователя с открытой вкладкой «Арматура»

Оператор имеет возможность ознакомиться с информацией о выбранном изделии в виде ИЭТР. После перехода по вкладкам открывается ин-

фор-мация по изделиям. После прохождения обучения предоставляется воз-можность тестирования.

Рассмотрим страницу тестирования пользователя (рис. 3).

АО "Разанский радиозавод" Образовательная платформа Главная Пройти обучение Пройти тестирование Информация об изделиях Выйти

В каком году первый раз произошло "слияние" бетона и арматуры?

в 1825 году

в 1887 году

в 1900 году

в 1925 году

Какие существуют способы сгибания арматуры?

Использование рычага

Использование активной заморозки

Использование нагревательных элементов с последующим участке молотка

Использование молотка

Ширина профиля швеллера может в колебаться в диапазоне?

от 32 до 90 мм

от 32 до 115 мм

от 45 до 90 мм

от 45 до 115 мм

Какие самые востребованные размеры швеллеров?

Швеллеры с номерами от 4 до 16

Швеллеры с номерами от 8 до 20

Швеллеры с номерами от 9 до 16

Швеллеры с номерами от 12 до 20

Рис. 3. Страница тестирования пользователя

Пользователь может пройти тестирование путем выбора вариантов ответа и нажатия кнопки «Закончить тестирование и получить результаты». После выполнения тестирования откроется страница результатов тестирования. В зависимости от правильности прохождения теста откроется та страница результатов теста, которая характеризует данные, полученные при тестировании.

Рассмотрим данные страницы результатов тестирования (рис. 4).

АО "Разанский радиозавод" Образовательная платформа Главная Пройти обучение Пройти тестирование Информация об изделиях Выйти

Результаты теста

Тест пройден!

Количество правильных ответов 13 из 13

Количество неправильных ответов 0

ВЕРНУТЬСЯ НА НАЧАЛЬНУЮ СТРАНИЦУ

Рис. 4. Страница результатов тестирования

Пользователь имеет возможность вернуться на начальную страницу или продолжить обучение снова, в случае не пройденного теста.

Если пользователь вошёл в систему как обучающийся, то перед ним будут списки тестов и ответов. Он может добавить или удалить тест, а также изменить данные в электронном каталоге (рис. 5).

В разработанном программном обеспечении предусмотрены возможности для добавления новых тестов, для изменения данных тестов, для удаления тестов. Также можно добавить или удалить ответ и изменить данные в электронном каталоге.

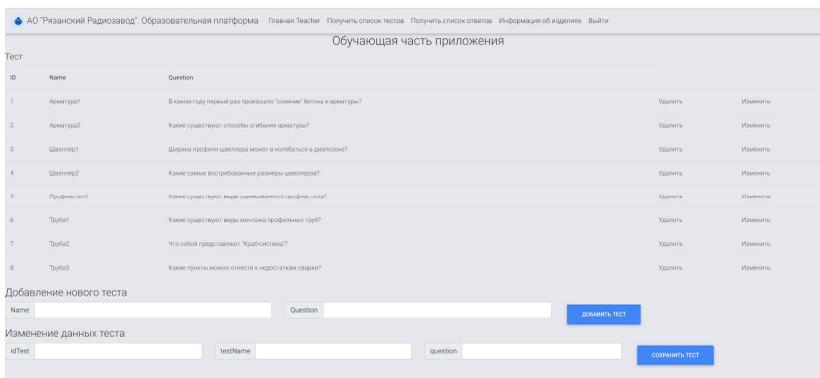


Рис. 5. Список тестов на странице обучающего

Если пользователь вошёл в систему как модератор, то перед ним будут списки пользователей. Он имеет возможность блокировать и разблокировать пользователя. Рассмотрим данные страницы модератора (рис. 6).

Для того чтобы заблокировать или разблокировать пользователя оператору нужно нажать кнопку «Блокировка/Разблокировка». В поле «Status» автоматически добавиться новый статус пользователя, затем нужно нажать на кнопку «Сохранить изменения».

Если оператор вошёл в систему как администратор, то перед ним будут списки пользователей. Он может добавить или удалить пользователя, а также изменить данные об этой сущности в электронном каталоге (рис. 7).

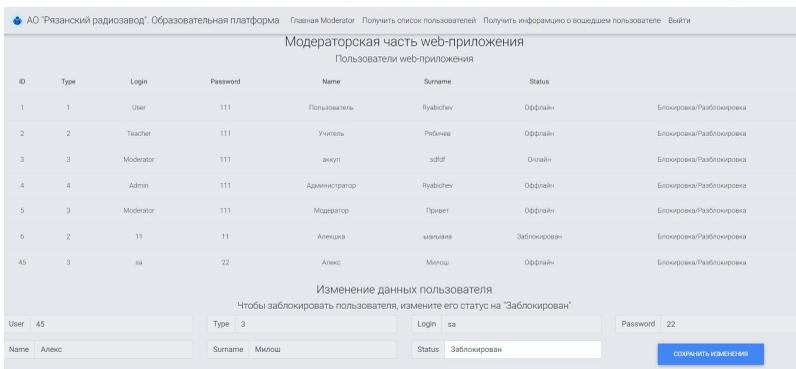


Рис. 6. Блокировка пользователя на странице модератора

После сохранения измененных данных в списке пользователей появится актуальная информация. На каждой странице есть возможность выхода в главное меню с нажатием кнопки «Выйти».

Разработанное решение протестировано при помощи типичных сценариев использования приложения. В процессе тестирования ошибок в ра-

боте программного продукта не выявлено. Тестирование показало, что программа полностью соответствует заявленным требованиям к составу выполняемых функций.

ID	Тип	Логин	Пароль	Имя	Фамилия	Статус		
1	1	User	111	Пользователь	Рябичев	Офлайн	Удалить	Изменить
2	2	Teacher	111	Учитель	Рябичев	Офлайн	Удалить	Изменить
3	3	Moderator	111	модер	Рябичев	Офлайн	Удалить	Изменить
4	4	Admin	111	Администратор	Рябичев	Онлайн		
5	3	Moderator	111	Модератор	Привет	Офлайн	Удалить	Изменить
6	2	11	11	Александра	Иванова	Заблокирован	Удалить	Изменить
25	3	111	111	mod	m111	Офлайн	Удалить	Изменить

Добавление нового пользователя

Тип Логин Пароль Имя Фамилия Статус

Рис. 7. Страница администратора

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дмитриева Т.А., Рябичев А.А. Необходимость разработки прикладного программного обеспечения для создания интерактивных электронных технических руководств по обслуживанию и ремонту изделий / Межевзовский сборник научных трудов Информационные технологии. – Рязань: Рязанский государственный радиотехнический университет имени В.Ф. Уткина, 2019. – 271 с.

УДК 004

Тюрина Е.М., Проказникова Е.Н.

ОЦЕНКА ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ ПОЖАРНОЙ ЧАСТИ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

Рассматривается ряд вопросов, связанных с оценкой эффективности информационных систем пожарной части.

Автоматизация деятельности пожарной части позволит: повысить эффективность доступа к информации, улучшить контроль процесса учета пожаров и исключить возможность появления ошибок в подготовке документации.

Ручное выполнение должностных обязанностей начальника пожарной части требует большого количества времени. В связи с этим возникла потребность в создании информационной системы учета пожаров, позволяющей повысить достоверность и оперативность обработки информации и снизить трудоемкость процессов обработки данных.

Выделяют 5 видов информационных систем:

– обеспечивающие критически важные процессы компании;

- выполняющие безальтернативные требования внешнего регулятора;
- осуществляющие поддержку;
- нацеленные на рост и развитие организации;
- закладывающие инновационный фундамент развития.

В настоящее время существует немало информационных систем по автоматизации учета пожаров (например, АРОГАНИТ АРМ ДДС 01, АСУ «Система-01» и др.). У этих систем есть ряд преимуществ, среди которых: прием заявок из системы 112, учет пожаров и другие. Однако следует отметить ряд существенных недостатков: рассмотренные системы являются дорогостоящими, содержат функциональные возможности, которые не востребованы в реальной деятельности сотрудников и требуют дополнительного обучения.

Существуют различные методы оценки эффективности информационных систем, которые зависят от конкретного типа предприятия и комплекса необходимых задач:

- метод инвестиционного анализа;
- качественный метод;
- вероятностный метод;
- метод финансового анализа.

Финансовый анализ предусматривает метод расчета общей (совокупной) стоимости владения информационными системами. Методология совокупной стоимости владения подходит для подсчета текущих стоимостных параметров, с ее помощью можно проанализировать затраты на эксплуатацию информационной системы.

Таким образом, переход от бумажного документооборота к электронному позволяет сократить время, требуемое на подготовку конкретных маркетинговых и производственных задач, исключить возможность появления ошибок в подготовке документации, повысить контроль над движением документов, существенно упростить и повысить скорость доступа к информации, а впоследствии повысить эффективность управления учреждением.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мусаев А. Интеллектуальный анализ данных: учебное пособие. А.А. Мусаев – СПб.: СПбГТИ(ТУ), 2018. – 56 с.
2. Михеев А. Алгоритмы интеллектуального анализа данных в информационной системе поддержки удалённого эксперимента – 2013. – № 2 – С. 86.

УДК 004

Бухтина А.О., Горбатова Ю.А.

ВЫБОР АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ
ОСНОВНЫХ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Проведен анализ оптимального выбора информаци-
онной системы для основных бизнес-процессов

Автоматизированные информационные системы в настоящее время внедряются во все области деятельности человека. Автоматизация помогает организации своевременно получить нужную информацию для дальнейшего оперативного принятия решений. При автоматизации бизнес-процессов организация сможет увеличить количество операций, выполняемых сотрудниками. Автоматизация бизнес-процессов имеет огромный класс задач, которые не ограничиваются только движением и обработкой документа, а также включают различные операции, выполняемые сотрудниками, и пошаговую обработку всех данных [1].

Автоматизированная информационная система бизнес-процессов предприятия должна ускорять информационный обмен между разными отделами организации, сокращать время для согласования документов.

Внедрение современных информационных систем позволяет повысить эффективность деятельности предприятия за счет скорости получения необходимой информации о состоянии его деятельности [2].

Основные бизнес-процессы, которым требуется автоматизация для повышения качества управления организации:

- управление персоналом;
- документооборот;
- оценка контрагентов;
- планирование деятельности организации.

Для выбора автоматизированной информационной системы рассмотрим «1С: Предприятие 8.3» [3]. Данная система состоит из ядра и разработанных для нее прикладных решений, которые называются конфигурациями. Такая архитектура системы 1С стала очень популярной, так как обеспечивает открытость прикладных решений.

Для управления персоналом была разработана конфигурация «1С: Зарплата и управление персоналом», которая предназначена для управления человеческими ресурсами предприятия любой численности. В данной конфигурации автоматизируются такие процессы как: расчет и выплата заработной платы, прием, перевод и увольнение сотрудников, кадровый и расчетный документооборот и т.д.

Для документооборота разработчики 1С создали конфигурацию «1С: Документооборот». Данная конфигурация автоматизирует процессы согласования и принятий документов, учет и планирование рабочего време-

ни, управление проектами. Одна из наиболее удобных функций данной кон-фигурации – возможный доступ с разных устройств, кроме того программа присылает уведомления о просрочке или о новых задачах пользователю на электронную почту.

Оценка контрагентов возможна в программе 1С с помощью сервисов «1С: Контрагент» и «1СПАРК Риски». Данные сервисы позволяют автоматически заполнять реквизиты контрагентов, оценивать их платежеспособность, выявлять «фирмы-однодневки».

Автоматизацию основных бизнес-процессов организации предпочтительнее реализовывать с помощью программы 1С по следующим причинам:

- высокая производительность программы в сравнении с аналогами;
- возможность настроить программу под нужды конкретного предприятия;
- возможность перехода программы от более старой версии к новой без необходимости создания новой базы данных;
- относительно низкая стоимость программы.

Таким образом, система 1С является наиболее оптимальным вариантом для предприятий, внедряющих автоматизацию в свои бизнес-процессы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Репин В.В. Бизнес-процессы. Моделирование, внедрение, управление. М.: Манн, Иванов и Фербер, 2014. 512 с.
2. Гвоздева В.А., Лаврентьева И.Ю. Основы построения автоматизированных информационных систем. М.: Форум, Инфра-М, 2016. – 320 с.
3. 1С [Электронный ресурс]. – Режим доступа: <https://v8.1c.ru/> Дата обращения 15.05.2020 г.

УДК 519.812.3

Белов В.В., Филоненко И.Н.

ПОКАЗАТЕЛЬ СПЕЦИАЛЬНОЙ ЛЕКСИЧЕСКОЙ СХОЖЕСТИ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Излагается метод вычисления показателя схожести информационных единиц по совокупности содержащихся в них специальных терминов, который может быть использован как косвенная оценка семантической общности сопоставляемых информационных источников.

Рассматриваемая задача

В педагогической среде часто обсуждаются вопросы, связанные с упущениями в подготовке абитуриентов к предстоящему обучению в вузе. Отмечается, что из того факта, что студенты, поступившие в вуз, имеют разную степень готовности к обучению, естественно образом, вытекает

следующее положение: студентов с разной степенью готовности к обучению необходимо учить по-разному, т.е. организация учебного процесса в общем случае должна предусматривать несколько версий процесса обучения (учебных траекторий). Отказ от указанного положения, имеющий место во всех вузах без исключения приводит к тому, что студенты с недостаточной степенью готовности к обучению завершают обучение с недостаточной степенью освоения надлежащих знаний, навыков и умений. О том, что разные студенты в разной степени осваивают, предусмотренные процессом обучения компетенции, говорят разные оценки в приложении к диплому. При этом выпускники с троечными дипломами в подавляющем большинстве не осваивают практически ничего из того, что предусматривалось учебными планами.

Позитивным управляющим воздействием на учебный процесс, позволяющим научить студентов «приемлемому минимуму», является организация раздельного обучения. Очевидно, что «приемлемый минимум», значительно предпочтительнее «практически ничего». Конечно же, строгая персонализация учебных траекторий (тьютеринг) требует значительных затрат, не каждому доступных и в зарубежных учебных практиках. По этой причине целесообразно осуществлять формирование учебных траекторий для групп обучаемых с примерно одинаковой степенью готовности к дальнейшему обучению. С той или иной формой разделения обучаемых по способностям и/или текущим уровнем готовности согласны многие специалисты-педагоги [1-4]. Наиболее систематически идея дифференциации обучения изложена в [4].

При осуществлении раздельного обучения для «слабых» групп возникает потребность в реализации специального дополнительного процесса, который по сути дела, может быть назван процессом «доучивания». Этот дополнительный процесс может осуществляться как в форме дополнительной нагрузки на учащегося, так и за счёт текущих учебных мероприятий в вузе. Недостатком первого варианта являются противоречия с санитарно-гигиеническими нормами на учебную нагрузку, защищаемыми юридическими нормативными актами. В принципе, на эти противоречия можно либо не обращать внимания («Ну, не доучился вовремя, пусть помучается сейчас», но это чревато последствиями), либо «закрывать» заявлениями слабых студентов с просьбой о дополнительных занятиях.

Теоретическим недостатком второго варианта является новая недоученность, – доучиваемые по пререквизитам студенты недоучат новые дисциплины, изучаемые уже в вузе. Однако, этот недостаток является чисто иллюзорным, – недоученные по пререквизитам студенты недоучат новые дисциплины безусловно, а при доучивании они, во-первых, получают шанс всё-таки одолеть новый материал, а, во-вторых, они становятся более образованными людьми, что, в конечном счёте, является решающим достоинством.

Образно говоря, вопрос «Доучивать или нет?» содержательно эквивалентен вопросу «Учить или нет?». Попытка учить специальным дисциплинам студентов, не освоивших пререквизиты – это практически без-условная потеря времени и средств.

Построение «почти индивидуальных» учебных траекторий предполагает усложнение процесса составления учебных программ: они теперь должны включать не только специальный дисциплинарный материал, но и дополнений из предметов, не доученных в школе. Причем для каждой статусной группы объём этого материала должен быть различным. Естественно, этот процесс явно нуждается в программно-информационной поддержке. Общая задача дозирования дополнительного учебного материала требует предварительного выяснения вопроса: «Какие пререквизиты и в какой степени связаны с дисциплиной, учебная программа которой составляется?» В качестве средства выяснения этого вопроса в настоящей статье предлагается метод оценки степени схожести информационных единиц по совокупности содержащихся в них специальных терминов, названной оценкой специальной лексической схожести.

Предлагаемая оценка специальной лексической схожести

Лексической схожестью информационных единиц назовём количественное выражение доли слов с одинаковой морфологической основой в этих единицах.

Специальной лексической схожестью информационных единиц назовём количественное выражение доли слов специальной лексики с одинаковой морфологической основой в этих единицах.

Специальная лексика – это совокупность слов, связанных с профессиональной деятельностью людей. К специальной лексике относятся термины и профессионализмы.

Получить специальную фракцию из некоторого текста можно путём удаления из этого текста слов общеупотребительной лексики

Общеупотребительная лексика – это совокупность слов, используемых, понимаемых и употребляемых – людьми в быту, художественной литературе и прочих языковых сферах независимо от места жительства, профессии, образа жизни. Общеупотребительную лексику образуют многие существительные, прилагательные, наречия, глаголы, местоимения, большинство служебных слов. Одним из первых словарей общеупотребительных слов явился словарь «Полный толковый словарь всех общеупотребительных иностранных слов» Н. Дубровского, изданный в 1895 году, содержащий 530 страниц слов, вошедших в русский язык, с указанием корней, в качестве настольной справочной книги «для всех и каждого при чтении книг, журналов и газет». Не смотря на наличие словарей общеупотребительной лексики, представляется целесообразным создание «собственного» такого словаря, для каждой учебной дисциплины. «Собственный» словарь может быть сформирован относительно малыми затра-

тами путём удаления из известного словаря «не столь уж» общеупотребительных слов. Например, категоризируемые в большинстве словарей как общеупотребительные названия цветов могут быть признаны как специальные для химии, биологии и географии.

Назовём показателем ξ специальной лексической схожести двух информационных единиц A и B следующее отношение:

$$\xi = \frac{|(A \setminus C) \cap (B \setminus C)|}{|A \setminus C|},$$

где C – множество общеупотребительных слов; $|M|$ – символ мощности множества M .

Заметим следующее: $A \setminus C$ – это множество специальных терминов в единице A ; $B \setminus C$ – множество специальных терминов в единице B ; $|(A \setminus C) \cap (B \setminus C)|$ – множество специальных терминов, содержащихся одновременно и в A , и в B ; ξ – это доля общих для A и B специальных терминов от совокупности специальных терминов информационной единицы A .

Заключение

Показатель ξ специальной лексической схожести информационных единиц A и B можно рассматривать как косвенную оценку семантической общности этих единиц, на основе которой можно принимать решение о степени значимости единицы A для освоения содержания единицы B .

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лейтес Н.С. Умственные способности и возраст. М.: Педагогика, 1971. – 277 с.
2. Новые педагогические и информационные технологии в системе образования: учеб. пособие для студ. высш. учеб. заведений / Е.С. Полат, М.Ю. Бухаркина, М.В. Моисеева, А. Е. Петров; под ред. Е.С. Полат. – 3-е изд., испр. и доп. – М.: Издательский центр «Академия», 2008. – 272 с.
3. Шадриков В.Д. Способности и интеллект человека. М.: Издательство современного государственного университета, 2004 – 188 с.
4. Платонова С.М. Детская одарённость. Учебно-методическое пособие. – Санкт-Петербург: Издательство Ленинградского государственного университета имени А.С. Пушкина, 2011. – 81 с.

УДК 004.89

Федосова Е.Б.

РАЗРАБОТКА МОРФОЛОГИЧЕСКОГО АНАЛИЗАТОРА
ВОПРОСНО-ОТВЕТНОЙ СИСТЕМЫ
ДЛЯ МЕТРОЛОГИЧЕСКОЙ ЛАБОРАТОРИИ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

В настоящей статье рассмотрена разработка морфологического анализатора для ВОС на языке программирования Python. Разобраны основные шаги определения окончания слова, его основы, получения морфологических характеристик.

Под морфологическим анализом понимается обработка словоформ вне связи с контекстом [1]. Задачей морфологического анализатора является построение совокупности строк морфологической информации (МИ). Для разработки морфологического анализатора (МА) выбрали язык программирования Python.

На вход МА поступает предложение, сформулированное пользователем системы. Поскольку пользователь строит запрос на естественном языке, то входное предложение может содержать неинформативные с точки зрения машинной обработки слова. Например, в запросе «Скажите, пожалуйста, какой диапазон измерения у расходомера номер 456?» слова «скажите» и «пожалуйста» излишни для МА. Т.е. первым этапом МА будет отсечение избыточных слов входного запроса.

Алгоритм работы МА зависит от способа хранения информации в системе. В настоящей работе реализован процедурный способ, т.к. он позволяет сократить расходуемый объем памяти по сравнению с декларативным.

Для работы МА требуются несколько элементов: таблица окончаний, словарь основ, содержащий правильные основы и номера флективных классов, и таблица морфологической информации. Наиболее существенной является таблица окончаний. Она используется для определения необходимой МИ, а именно числа и падежа для существительных и прилагательных [2]. Таблица окончаний представлена в виде класса констант `class TableOfEndings(object)`, каждый элемент которого представляет собой список с элементами-буквами окончания. Например, окончание «ий» представлено в таблице как `End5 = ['u', 'i']`. Такая структура необходима для определения окончаний слов и дальнейшего получения основы слов. Определение номера окончания слова реализовано функцией `def GetNumberOfEnd(word)`, на вход которой подается анализируемое слово. Производится побуквенное сравнение последних букв слова и букв окончания из таблицы окончания:

```
if TableOfEndings.End1[0] == word[lenOfWord - 1]:
```

```

numberOfEnd = 1
elif TableOfEndings.End2[1] == word[lenOfWord - 1] and TableOfEndings.End2[0] == word[lenOfWord - 2] numberOfEnd = 2...

```

Следующим шагом МА является получение основы слов. Для этого требуется удалить из слова полученное ранее окончание. Такое действие реализует функция *defGetBaseOfWord(word, end)*, принимающая на входе само слово и определенное ранее окончание.

Далее необходимо определить к какому флективному классу относится полученная основа. Для этого используется словарь основ, реализованный в виде структуры данных словарь (*dict*), и функция *defGetNumberOfFleckClasses(BaseOfWord)*. Структура словаря выглядит так: *DictFleckClasses = {1: [' ', ' ', ...], 2: [' ', ' ', ...],}*, где 1,2 ... – номера флективных классов, [' ', ' ', ...]... – список основ слов, принадлежащих данному флективному классу.

В общем случае, для приписывания основе слова и окончанию морфологических характеристик используется морфологическая таблица с двумя входами. Строки в таблице являются порядковыми номерами окончания, столбцы – номерами флективных классов; на их пересечении указывается номер соответствующей МИ. При программной разработке системы морфологическую таблицу не реализовали в явном виде; ее заменили функцией, которая на вход принимает номер окончания номер флективного класса, «склеивает» эти два номера, получая при этом номер МИ. Далее, по полученному номеру извлекаем из таблицы МИ саму морфологическую информацию. Таблицу МИ также реализовали с помощью словарей (*dict*): *DictMorphInf = {01: ['Сущ', 'мр', 'ед.ч', 'им'], ...}*, где 01 – номер МИ, полученный после склеивания (0 – номер окончания, 1 – номер флективного класса), а ['Сущ', 'мр', 'ед.ч', 'им'] – морфологические характеристики, присущие основе слова и окончанию.

Полученные для каждого слова морфологические характеристики добавляются в пустой список *MorphInfOfSentence*. Таким образом, на выходе МА получаем совокупность строк морфологической информации, описанных с помощью такой структуры данных, как список. В общем случае, строка МИ выглядит так: [[' ', ' ', ' ', ' ', ...], [' ', ' ', ' ', ' ', ...], ...], где количество вложенных списков равно количеству слов во входном запросе. В каждом из вложенных списков индекс элемента играет важную роль – идентифицирует тот или иной морфологический признак. Так, индекс 0 соответствует части речи (существительное, глагол, предлог и т.д.), индекс 1 – роду, индекс 2 – числу, индекс 3 – падежу и т.д. Если у части речи отсутствует какой-либо признак (например, у глагола нет падежей), то место списка с определенным индексом будет пустым.

Описанные таким образом морфологические характеристики будут удобны для дальнейшей обработки входного предложения, в частности,

для того, чтобы передать морфологическую информацию на блок синтаксического анализа текста.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Попов Э.В. Общение с ЭВМ на естественном языке: монография / Э.В. Попов. – Москва: Наука, 1982. – 360 с.
2. Бажанова А.И., Мартыненко Т.В., Андриевская Н.К. Разработка морфологического анализатора для построения понятийного аппарата предметной области // Информатика и компьютерные технологии. – 2011, с. 326-330

УДК 004.9

Авачева Т.Г., Дорошина Н.В., Кабанов А.Н.

ПОВЫШЕНИЕ ОПЕРАТИВНОСТИ И ТОЧНОСТИ АЛГОРИТМОВ ОПТИМИЗАЦИИ МНОГОМЕРНОЙ, МНОГОКРИТЕРИАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ФОРМИРОВАНИЯ ЭКСПЕРТНЫХ ГРУПП МЕТОДОМ ВЗВЕЩЕННОГО ПАРНОГО СРАВНЕНИЯ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань.
Рязанский государственный медицинский университет им. акад. И.П. Павлова, г. Рязань

Рассматривается задача формирования управляющих экспертных групп в задаче оптимизации многомерной, многокритериальной системы за счет предварительного определения веса каждого критерия.

Постановка задачи

Целью работы является поиск решения для повышения оперативности и точности оптимизации многомерной, многокритериальной системы. Предлагается технология обработки, которая состоит из нескольких этапов:

- 1) построение областей допустимых изменений параметров для каждого критерия; построение общей области допустимых изменений для всех критериев с исключением лишних неравенств;
- 2) определение оптимума номинала для общей области допустимых изменений параметров;
- 3) определение объема для общей области допустимых изменений параметров и каждого критерия;
- 4) формирование экспертных групп методом взвешенного парного сравнения;

Теоретическая часть

1. Метод гиперплоскостей заключается в последовательном включении каждой граничной точки в N -мерном пространстве в выпуклую оболочку и в исключении гиперплоскостей, оказавшихся внутри области. Вычислительная процедура построения области работоспособности по граничным точкам методом гиперплоскостей заключается в выполнении следующих операций. [1].

1. Выбираются произвольным образом первые $(N + 1)$ граничные точки и строятся по ним $(N + 1)$ гиперплоскости. Для каждой построенной

гиперплоскости запоминаются координаты граничных точек, по которым она построена, и координаты ее вершины.

Вершиной данной гиперплоскости условимся называть ту точку из выбранных $(N + 1)$ точек, через которую не проводится гиперплоскость.

2. Определяется для следующей, выбранной произвольно, граничной точки соответствующая ей генеральная гиперплоскость. Генеральной гиперплоскостью данной граничной точки будем называть гиперплоскость, вершина которой и данная граничная точка расположены по разные от нее стороны

3. Выполняется п. 1 для данной граничной точки и точек, через которые была ранее проведена ее генеральная плоскость, найденная в п. 2. Затем стираются значения коэффициентов генеральной гиперплоскости, координаты ее вершины и точек, через которые она проведена.

4. Выбирается следующая по порядку граничная точка, и все повторяется с п. 2. После перебора всех граничных точек процесс построения области работоспособности заканчивается и производится определение знаков « \geq », « \leq » для системы линейных неравенств.

II. Метод оптимума номинала. Нахождение точки для общей области

Для определения точки внутри области примерно на равном расстоянии от границы области применим квадратичный критерий.

Исходная система ограничений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1; \\ \dots\dots\dots; \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m. \end{cases} \quad (1)$$

Левые части обозначим:

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n; \\ \dots\dots\dots; \\ y_m = a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n; \end{cases}$$

Берём y_1 как целевую функцию и на множестве (1) находим максимальное и минимальное значения y_1 . Затем берем y_2 и опять находим максимальное и минимальное значения y_2 и т.д. ($y_{i \min}$, $y_{i \max}$).

$$\bar{y}_i = \frac{y_{i \min} + y_{i \max}}{2} - \text{среднее значение.}$$

Тогда система ограничений примет вид:

$$\begin{cases} \bar{y}_1 = a_{11}\bar{x}_1 + \dots + a_{1n}\bar{x}_n; \\ \dots\dots\dots; \\ \bar{y}_m = a_{m1}\bar{x}_1 + \dots + a_{mn}\bar{x}_n. \end{cases}$$

Это система линейных алгебраических уравнений.

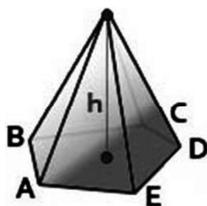
$$\bar{y}_{cp} = \bar{A} \cdot \bar{x}_{cp};$$

$$\bar{x}_{cp} = \left(\bar{A}^T \cdot \bar{A} \right)^{-1} \cdot \bar{A}^T \cdot \bar{y}_{cp}.$$

III. Определение объема для общей области критериев и для каждого критерия

Объем пирамиды

Пирамидой называют многогранник, основанием которого является произвольный многоугольник, а все грани представляют собой треугольники с общей вершиной, являющейся вершиной пирамиды.



$$V = 1/3 \cdot S \cdot h.$$

где S – площадь основания, а h – высота пирамиды.

Высотой пирамиды называется прямая, опущенная из ее вершины к основанию под прямым углом. Соответственно, чтобы найти объем пирамиды, необходимо определить какой многоугольник лежит в основании, рассчитать его площадь, узнать высоту пирамиды и найти ее объем.

Вначале найдем объем V для общей области. Таким же образом найдем и для каждого критерия V_1, V_2, \dots, V_n . Находим вес каждого критерия $V_1/V, V_2/V, \dots, V_n/V$.

Эти оценки весов могут использоваться экспертами, что важно при создании однородных групп методом парного сравнения

4. Формирование экспертной группы методом взвешенного парного сравнения

Введем матрицу $\bar{A} = \{a_{ij}\}$, где a_{ij} – во сколько раз i -й критерий важнее j -го; $a_{ij} = 1/a_{ji}$. Каждый эксперт указал веса с учетом данных ему весов критериев.

Изложим последовательность операций [2]:

$$\bar{A} \cdot \bar{A}^T \rightarrow \bar{U} \bar{D}_1 \bar{U}^T;$$

$$\bar{A}^T \cdot \bar{A} = \bar{U} \bar{D}_2 \bar{V}^T;$$

$$\bar{D} = \bar{U}^T \cdot \bar{A} \cdot \bar{V}.$$

Находим самое большое собственное число. Все остальные диагональные элементы обнуляем:

Находим

$$\hat{A} = \bar{U} \hat{D} \bar{V}^T.$$

На основе полученных данных строится остоное дерево. Создание однородных групп проводится методом разрезания кратчайшего остоного дерева.

Заключение. Предложенный метод позволяет создавать однородные группы экспертов для многокритериальной адаптивной системы управления. Адаптацию целесообразно проводить по количеству однородных групп.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Технологии обработки информации: учебное пособие / Т.Г. Авачева [и др.]; 2017. – 92 с.
2. Г.Дэвид Метод парных сравнений Москва, Статистика, 1978, 140с.

УДК 004

Мельникова И.С.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ ПОВЫШЕНИЯ КАЧЕСТВА РАБОТЫ И КОНКУРЕНТОСПОСОБНОСТИ ПРЕДПРИЯТИЙ РОЗНИЧНОЙ ТОРГОВЛИ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина г. Рязань

В настоящее время всё больше предприятий переходит на автоматизированные системы для получения наивысшего результата своей деятельности, это связано с развивающимися технологиями, более быстрой и эффективной работой предприятий.

Оценка конкурентоспособности необходима для разработки мероприятий по повышению конкурентоспособности, выбора контрагентов для совместной деятельности, составления программы выхода компании на новые рынки сбыта, осуществления государственного регулирования экономики, осуществления инвестиционной деятельности. Цель осуществления оценки конкурентоспособности компании – определить положение ее на исследуемом рынке.

В настоящее время всё больше предприятий переходит на автоматизированные системы для получения наивысшего результата своей деятельности, это связано с развивающимися технологиями, более быстрой и эффективной работой предприятий. Из этого следует, что чем больше автоматизации на предприятии, тем эффективнее его деятельность, в этом заключается актуальность темы данной статьи.

Торговля в розницу претерпевает на сегодняшний день пору радикальных перемен. Ранее отечественный ритейл демонстрировал постоянное увеличение числа фирм и их упрочение. Но в рамках экономического кризиса, спада покупательской способности, ожесточенной конкуренции

и возросшей требовательности клиентов, методики, применяющиеся в местах, торгующих в розницу, перестают быть жизнеспособными. Отрицательно сказываются падение покупательской способности, ужесточение конкурентных отношений, падение уровня выручки, площади и количества торговых площадей, появление иностранных конкурентов на внутреннем рынке.

Чтобы сохранить свои позиции и сопротивляться негативным обстоятельствам, фирмы, специализирующиеся на торговле в розницу, должны своевременно реагировать на потребности клиентов и на изменения во внешней среде. Большая часть розничного сегмента рынка вынуждена использовать методы увеличения продуктивности ведущейся предпринимательской деятельности.

Одна из таких методик – процессный менеджмент, позволяющий перейти фирме на новый качественный уровень путем перманентного усовершенствования внутренних предпринимательских процессов, сокращения расходов и применения новых информационных методик.

Опыт сегодняшнего дня показывает, что применение инновационных технологий повышает конкурентоспособность предприятия, повышает ее продуктивность и ряд иных показателей, например, уровень прибыли.

На сегодняшний день конкурентная борьба между сетями розничной торговли в самом разгаре, поэтому большая их часть озабочена проблемой увеличения продуктивности бизнеса. Продуктивность компании в немалой степени зависит от уровня оснащенности необходимыми информационными технологиями и программным обеспечением, помогающим взаимодействовать с огромными массивами данных. Осознавая конкурентный потенциал информационных систем, представители розничной торговли ведут борьбу на уровне IT-технологий. Успешное внедрение передовой информационной технологии – существенное преимущество над конкурирующими компаниями.

Существует необходимость в оценке конкурентоспособности какого-либо предприятия, так как в условиях рыночной экономики оценка конкурентных позиций является неотъемлемым элементом деятельности любого хозяйствующего субъекта. Анализ конкурентов и условий конкуренции определенной сферы требуется предприятию для того, чтобы определить, в чем его достоинства и недостатки перед конкурентами, получить результаты для выработки предприятием собственной успешной конкурентной стратегии и поддержания конкурентного преимущества. Определение конкурентоспособности компании является важным элементом деятельности любого хозяйствующего субъекта. [1]

Оценка конкурентоспособности необходима для разработки мероприятий по повышению конкурентоспособности, выбора контрагентов для совместной деятельности, составления программы выхода компании

на новые рынки сбыта, осуществления государственного регулирования экономики, осуществления инвестиционной деятельности.

Цель осуществления оценки конкурентоспособности компании – определить положение ее на исследуемом рынке.

Продуктовые методы оценки основываются на том, что оценка конкурентоспособности предприятия может быть произведена через оценку конкурентоспособности его продукции: чем выше ее конкурентоспособность, тем выше конкурентоспособность компании. К преимуществам рассматриваемого метода относится то, что он учитывает одну из наиболее важных составляющих конкурентоспособности – конкурентоспособность продукции. Основным недостатком этого метода является то, что он позволяет получить ограниченное представление о преимуществах и недостатках в работе предприятия, так как его конкурентоспособность принимает вид конкурентоспособности продукции и не затрагивает другие аспекты его работы. [2]

Матричные модели получили свое название в связи с использованием матричной формы отражения результатов оценки и анализа. Другой характерной чертой данной модели явился ярко выраженный акцент на маркетинговой оценке деятельности предприятия, в результате чего компания рассматривается как совокупность различных бизнес-единиц (продуктовый портфель).

Достоинством модели является то, что здесь используются объективные индикаторы привлекательности и конкурентоспособности, уменьшая вероятность субъективизма. Метод отражает финансовое взаимодействие в рамках портфеля компании и финансовые соображения, которые следует принимать во внимание, поясняет, почему могут различаться приоритеты в распределении ресурсов между отдельными предприятиями компании. К недостаткам относится то, что метод основывается на концепции кривой опыта. Степень важности накопленного производственного опыта, с точки зрения снижения удельных затрат в различных отраслях, может быть различна. Особенно непредсказуемой эта связь является в условиях современной экономики. [2]

Основным фактором, обеспечивающим высокую продуктивность компаний есть высокая степень усовершенствования бизнес-решений. Подобное возможно выполнить, воспользовавшись решением класса ERP. «ERP-система» (Enterprise-Resource Planning), согласно словарю APICS, определяется как система, которая позволяет идентифицировать и учитывать ресурсную базу компании, позволяющую производить изготовление, закупку, реализацию продукции.

Разработанные ERP-системы, предназначенные для розничных продаж, включают в свой состав модули, предназначенные для выполнения специализированных задач. Они дают возможность организовать деятельность каждого отдела предприятия розничной торговли: автоматизировать

ценообразование, предугадать спрос на ту или иную продукцию, контролировать финансовые потоки. Основное преимущество подобных узкоспециализированных систем – малые сроки ввода в эксплуатацию и приемлемая цена настройки.

Самыми востребованными системами в России и за границей являются SAP, Microsoft Dynamics AX, 1С, Oracle E-Business [2]. Применение подобных систем для руководства сетью розничной торговли позволяет увеличить эффективность деятельности компании: рост количества обработанных заказов на 45-55 %, рост прибыли на 5-7 % и снижение количества неликвидного товара на 65 %. Большая доля ритейлеров уже внедрили ERP-системы, что заметно увеличило эффективность их деятельности.

В условиях экономического кризиса предприятию, работающему в сфере розничной торговли, следует следить за собственной конкурентоспособностью. Для этого нужно, чтобы все отделы компании взаимодействовали как часовой механизм, для этого следует изменить ИТ-рельеф предприятия.

В настоящее время важное значение имеет процессный подход к менеджменту. Такой подход в предпринимательской деятельности – способ получить конкурентное преимущество.

ERP-система стремится «повторить» бизнес-процесс фирмы в виде программного обеспечения и поддерживать функции какого-либо представителя персонала, давая ему вспомогательные инструменты в виде данных, возможности контроля и планирования и прочие функции, что делает важным подстройку ERP-системы в разрезе инициализации бизнес-процессов компании, так как каждая ERP настраивается под определенный тип предпринимательской деятельности [2].

Таким образом отделы и филиалы компании интегрируются в общую систему на базе единой информационной базы. Также есть положительные отличия инициализации бизнес-процессов: наглядность для персонала обновляемых правил работы системы, данные процессы легче формировать и изменять согласно переменам на предприятии, процессы являются почвой для последующей автоматизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Караева Ф.Е. Конкурентоспособность промышленного кластера, экономическая и управленческая категория. Экономика и предпринимательство, 2013. 871-874 с.

2. Лазаренко А.А. Методы оценки конкурентоспособности, 2014, 374-377 с.

УДК 681.3

Кабанова Е.О., Крошилин А.В.

АВТОМАТИЗИРОВАННОЕ РАНЖИРОВАНИЕ ЗАДАЧ ПО ПРИОРИТЕТАМ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина г. Рязань

В статье рассматривается автоматизированное ранжирование задач по приоритетам, основываясь на теории планирования методом Эйзенхауэра.

Ранжирование (сортировка) любого типа задач по приоритетам зачастую является ресурсозатратным и трудоемким процессом, поскольку от исполнителя требуется одновременно проанализировать и запомнить, такие значительные характеристики задач, как важность и срочность, затем также необходимо расставить их в правильном порядке – по приоритетам.

Одним из способов упрощения процесса ранжирования и анализа задач является метод матриц Эйзенхауэра, который подразумевает разделение задач и действий на четыре группы:

- 1) срочные и важные;
- 2) важные, но не срочные;
- 3) срочные, но не важные;
- 4) не срочные и не важные.

Конечная цель метода Эйзенхауэра – помочь отфильтровать второстепенные дела от важных решений и сосредоточиться на том, что действительно имеет значение [1].

Если представить метод Эйзенхауэра в виде матрицы, то он будет выглядеть так:

	СРОЧНЫЕ	НЕ СРОЧНЫЕ
ВАЖНЫЕ	КВАДРАНТ I	КВАДРАНТ II
НЕ ВАЖНЫЕ	КВАДРАНТ III	КВАДРАНТ IV

Рис. 1. Матрица Эйзенхауэра

Таким образом, для того чтобы ранжировать задачи, необходимо их распределить в соответствующие квадранты, каждый из которых имеет следующее описание:

- Квадрант I – «Сделайте это немедленно» (срочные и важные);
- Квадрант II – «Решите, когда вы это сделаете» (важные, но не срочные);
- Квадрант III – «Делегируйте кому-нибудь» (срочные, но не важные);
- Квадрант IV – «Сделайте это позже» (не важные, не срочные) [2].

Однако при использовании данного метода могут возникнуть следующие проблемы:

1. Сложность оценивания важности задачи, особенно если эти задачи из разных областей.

2. Не сбалансированное распределение, которое может привести к проблемам в будущем. Например, имеются несколько срочных и важных задач (квadrант I) с большой сложностью, а также несколько не важных, но срочных задач (квadrант III). При данном распределении начинаем выполнять задачи из I квadrанта, поскольку они являются наиболее приоритетными, однако решение этих задач может занять больше времени, чем планировалось, в результате до задач из квadrанта III рабочего времени может не хватить. Хотя, они, были совсем не трудозатратными и могли быть завершены в короткие сроки, т.е. их можно было выполнить без значительного вреда для наиболее приоритетных задач.

Однако данные проблемы еще больше проявляются, если поставленная цель – это планирование финансово-хозяйственной деятельности на предприятии, где количество статей плана может быть очень большим, а сравнить их важность является еще более сложной и трудно оцениваемой задачей.

Для того, чтобы иметь возможность ранжировать задачи автоматически, необходимо запрограммировать алгоритм сортировки, а также иметь простые и сравниваемые входные параметры, у которых задан определенный вес.

К таким параметрам могут относиться: важность, срочность, время выполнения и т.д. Данные параметры могут варьироваться в зависимости от специфики деятельности компании.

Так параметр «важность» можно представить в виде значения из предопределенного списка, например, увеличение прибыли, сокращение расходов, повышение репутации, уменьшение выбросов. Каждому значению из данного списка будет соответствовать определенный коэффициент, в зависимости от стратегических целей компании на период планирования.

Параметр «срочность» будет представляться в виде сроков окончания задачи.

«Время выполнения» - выражается в часах, которые будут затрачены на реализацию задачи.

Также, помимо собственного значения, каждый параметр имеет коэффициент (вес), который определяет степень значимости данного параметра при расчете приоритета задачи. Что позволяет сделать настройку более гибкой.

Для расчета приоритета можно воспользоваться различными методами решения многокритериальных задач. Простейшим являться метод объединения всех критериев в одну функцию приспособленности, используя линейное соотношение.

$$\text{Приоритет Задачи} = \Pi_1 \times \text{Вес}_{\Pi_1} + \Pi_2 \times \text{Вес}_{\Pi_2} + \dots + \Pi_i \times \text{Вес}_{\Pi_i}$$

где, P_1, P_2, P_3 – значения параметра, $Вес_{P_1}, Вес_{P_2}, Вес_{P_3}$ – вес параметра

Для более точных вычислений можно воспользоваться, например, методами, основанными на границах и множествах Парето, также называемые методы поиска Парето – оптимальных решений для многокритериальных задач [3].

После вычисления значения приоритета для каждой из задач, производится их сортировка.

Использование автоматизированного алгоритма для ранжирования задач, позволяет сократить время планирования, уменьшить трудозатратность, исключить вероятность ошибки, а также дает возможность моделировать различные ситуации в зависимости от значений весов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кузьмин А.М. Матрица Эйзенхауэра. Один из инструментов управления развитием бизнеса / А.М. Кузьмин, Е.А. Высоковская // Методы менеджмента качества – 2016 –19 с.

2. Крицкая М.В. Матрица Эйзенхауэра: простой инструмент для определения задач / Крицкая М.В. // Контур – электронный журнал – 2018.

3. Sean Luke, Essentials of Metaheuristics / Sean Luke – Lulu, 2016. – 133-147 с.

УДК 681.3

Пахомова Н. Ф., Крошилина С.В.

ОБЗОР СУЩЕСТВУЮЩИХ ИНФОРМАЦИОННЫХ СИСТЕМ И ВЫБОР ЭФФЕКТИВНОГО СПОСОБА АВТОМАТИЗАЦИИ АТЕЛЬЕ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина г. Рязань

В работе рассматриваются, особенности и трудности автоматизации. Целью работы является рассмотрение информационных систем, выявление их плюсов и минусов, а так же выбор эффективного способа для автоматизации ателье.

Ключевые слова: автоматизация, способы реализации, информационная система, ателье.

Введение

В современном мире люди любят выделяться и показывать свою индивидуальность, это выражается даже в одежде. Не всегда можно найти в магазине то, что тебе по душе, однако не каждый может сшить себе одежду или просто отремонтировать изделие, поэтому ателье по ремонту и пошиву одежды приходят на помощь и пользуются большим спросом.

На данный момент, в работе ателье по ремонту и пошиву одежды нет единой серверной части, и вся работа ведется в бумажном виде. Это приводит к сложному поиску нужной информации по услугам и фурнитуре, усложняет расчет заказов и оформление прихода фурнитуры, на что ухо-

дит много времени. Так же при расчете вручную человек может допустить ошибку, что может привести к потере прибыли.

Учитывая данные факты можно оценить высокую значимость и чрезвычайную важность исследования способа автоматизации ателье.

В настоящее время существует множество способов, позволяющих автоматизировать работу ателье, например, внедрение готовой информационной системы, использование различных платформ в сети интернет и даже обработка данных в программе MS Excel. Но чтобы точно сказать, какой из перечисленных вариантов является наилучшим, следует проанализировать работу ателье и разобрать каждый из предложенных способов.

Актуальность научно-исследовательской работы заключается в том, что для организации особенно важна автоматизация процессов для увеличения производительности работы ателье. Предметом исследования является ателье по ремонту и пошиву одежды.

Целью научно-исследовательской работы является сбор материала по теме эффективность автоматизации заказов в ателье, его обобщение в целом и анализ проблемы для того, чтобы выявить оптимальный способ автоматизации.

Необходимость и трудности автоматизации ателье

Несмотря на широкое распространение интернета, персональных компьютеров и возможность автоматизации деятельности, ещё остаются немало предприятий, которые никак не оптимизируют свою работу [1]. И ателье – не редкость, хотя эффективность оптимизации работы после внедрения автоматизированных систем наглядно демонстрируется на реальных примерах.

Рассматривая частные ателье, можно выделить целый ряд причин, по которым автоматизация тормозится. Основными причинами можно назвать.

1. Отсутствие чёткого понимания экономической выгоды от автоматизации деятельности.

Вполне логичным, что сокращение трудозатрат посредством использования программного обеспечения ведёт к сокращению времени, которое затрачивается на выполнение рутинных действий. А если на задачу тратится меньше время, то освобождается время, которое можно потратить на ответ на телефонный звонок, приём ещё одного клиента и своевременную подготовку отчетных данных.

2. Отсутствие необходимых для вложения в программное обеспечение денежных средств.

Узкоспециализированные программы могут стоить огромных денег, которые не под силу малому бизнесу, а программы общей специализации не всегда удобны и могут нуждаться в доработке, которая, тоже, стоит вложений.

3. Используя неадаптированную под цели программу, пользователи могут разочароваться и вернуться к ведению бумажной документации.

В данном случае следует проанализировать рынок узкоспециализированного программного обеспечения и осуществить подбор доступной по стоимости программу, удовлетворяющую целям фирмы.

Сотрудники, не являющиеся техническими специалистами, даже представить не могут, какие функции необходимы им в работе. Поэтому, подбирая программное средство для внедрения, нужно тщательно, проанализировать, кто будет работать с системой и какие функции им необходимы для удобной работы с программой.

Важнейшим и эффективным из плюсов автоматизации является оптимизация затрачиваемого времени на выполнение рутинных задач.

Основные способы реализации автоматизации ателье

Как оговаривалось ранее, существует большое количество способов автоматизации ателье. Рассмотрим наиболее популярные способы с точки зрения прикладного использования.

1. Способ внедрение готовой информационной системы.

Одним из способов автоматизировать ателье является внедрение системы управления базами данных (СУБД). Эти программы позволяют автоматизировать весь производственный цикл, отслеживать выпуск продукции, вести учет остатков материалов на складе, знать, сколько заработали сотрудники, получать данные по количеству изготовленной продукции и остаткам в производстве. Например, программа «Швейка 8» [3].

Внедрение системы, «Швейка 8», имеет такие плюсы, как полный контроль со стороны руководства, анализ прибыли и минимизация потерь клиентов. Минусы у подобных СУБД также имеются. Она не предоставляет функционала для аналитики неиспользовавшихся услуг.

2. Способ использование различных платформ в сети интернет.

Способом автоматизации ателье можно также назвать использование «облачных» сервисов в сети интернет, например такого как «РемОнлайн». Функционал их включает в себя порядок на складе, в заказах, оплатах и документах, готовый прайс и автоматический расчет стоимости заказов, учет выполненных работ и анализ эффективности сотрудников, возможность управления с любого удобного устройства: смартфона, планшета, ПК. Работа производится только с использованием технологии интернет [4].

Способ использования различных платформ в сети интернет по функционалу, аналогичен функционалу СУБД. Плюсом является отсутствие приобретать компьютерную технику с большим объёмом памяти, поскольку все работы выполняются в сети интернет, в «облаке». Минус в том, что при отсутствии интернета такая система будет неработоспособна.

3. Способ использования программы Microsoft Office Excel

Важно также отметить, что имеет право на существование способ обработки данных с использованием программы MS Excel.

Как сказано в информации на официальном сайте программы [6], в последние версии Excel были добавлены интеллектуальные функции и

теперь создание электронных таблиц на базе шаблонов и с использованием формул стало максимально простым. Диаграммы и графики, которые предоставлены, позволяют наглядно представлять данные. Способ обработки данных в программе MS Excel является не самым популярным. Его плюсом является простота использования, минусом – то, что аналитические данные потребуются вносить вручную и, хотя программа «сама» посчитает по нужным формулам отчетную информацию, придется изрядно потрудиться для её сбора.

Выбор эффективного способа для автоматизации ателье

Наглядно видно, что в системах представлены разнообразнейшие функции. Однако не каждое ателье имеет большой штат сотрудников и несколько отделов с различными направлениями. А потому можно сказать, что не для каждого ателье указанные системы станут эффективным решением. Проанализировав способы автоматизации ателье, можно отметить, что эффективным будет именно разработка своей информационной системы [5].

Заключение

Произведя тщательный анализ, пришли к выводу, что эффективным способом является разработка программного обеспечения информационной системы ателье.

Обобщая материал научно-исследовательской работы можно отметить, что цель научно-исследовательской работы была достигнута за счет анализа проблемы, за счет рассмотрения существующих информационных систем для того, чтобы выявить оптимальный способ автоматизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алан Р. Саймон. Стратегические технологии баз данных. – М.: Финансы и статистика, 2001.
2. Белов В.В., Чистякова В.И. Проектирование информационных систем: учебник. – М.: Академия, 2013. – 352 с.
3. «Швейка 8» - программа №1 в России для автоматизации швейных производств <https://www.shveyka8.ru/>
4. Программа №1 для учета и автоматизации бизнеса в сфере услуг [Сайт] – <https://remonline.ru/>.
5. Кинг Д. Создание эффективного программного обеспечения. – М.: Мир, 1991.
6. Программа для работы с электронными таблицами Microsoft Excel, бесплатная пробная версия Excel [сайт]. – URL: <https://products.office.com/ru-ru/excel>.

УДК 004**Миронова А.В.**
МЕТОДОЛОГИЯ БЭМ**Рязанский государственный радиотехнический университет им. В. Ф. Уткина г. Рязань**

В статье рассматривается методология, позволяющая разбить интерфейс на набор таких абстракций, как блок, элемент и модификатор.

Методология БЭМ (блок, элемент, модификатор) является совокупностью единых правил реализации и хранения кода независимо от используемых технологий (например, CSS, Sass, HTML, JavaScript или React), которые повышают его масштабируемость и переиспользуемость.

Бэм упрощает решение таких проблем как:

- 1) понимание старого или написанного другим разработчиком кода, а так же поддержка уже выполненных проектов;
- 2) написание самодокументирующегося кода, так как в классе любого элемента есть его описание;
- 3) переиспользование любого блока кода, так как можно использовать уже готовые наработки;
- 4) получение более стабильных предсказуемых результатов после применения стилей;
- 5) упрощение коммуникации между членами команды, так все называют все сущности одинаково;
- 6) уменьшение порога входа разработчика на проект, так как во всех проектах с использованием данной методологии одинаковая структура организации и одинаковые правила именования сущностей;
- 7) ускорение процесса отладки кода и разработки в целом.

По БЭМ разработка идет на основе классов (один из глобальных атрибутов HTML-тега, описывающий семантическое назначение), а не идентификаторов.

В БЭМ-проекте любой интерфейс можно поделить на блоки, которые могут содержать (или нет) элементы.

Блок – это логически и функционально независимый компонент страницы, элемент – это составная часть блока, которая не может быть использована в отрыве от него.

Принципы работы с блоками:

- 1) название блока описывает семантику, а не состояние;
- 2) блок не должен влиять на свое окружение и иметь внешнюю геометрию, то есть отступы или границ;
- 3) допустима вложенность блоков друг в друга;
- 4) допустима любая вложенность.

```
<!-- Блок `header` -->
<header class="header">
  <!-- Вложенный блок `logo` -->
  <div class="logo"></div>

  <!-- Вложенный блок `search-form`
  <form class="search-form"></form>
</header>
```

Рис. 1. Пример разметки блока по методологии БЭМ

Принципы работы с элементами:

- 1) название элемента описывает семантику, а не состояние;
- 2) имя элемента определяется по шаблону: имя-блока_имя-элемента, где имя элемента отделяется от имени блока двойным подчеркиванием
- 3) допустимо вложенность элементов друг в друга;
- 4) допустима любая вложенность;
- 5) элемент всегда является часть блока, а не другого элемента;
- 6) элемент необязательный элемент блока.

```
<div class="block">
  <div class="block_elem1">
    <div class="block_elem2">
      <div class="block_elem3"></div>
    </div>
  </div>
</div>
```

Рис. 2. Пример блока и элементов по методологии БЭМ

Если код можно использовать повторно и он не зависит от других компонентов, то нужно создавать блок, а если этот код невозможно использовать независимо без родительской сущности, то нужно создавать элемент.

Так же можно расширить, описание блока или элемента и использовать модификатор. Модификатор – это сущность, которое определяет вид, состояние или поведение.

Принципы работы с модификаторами:

- 1) название определяет внешний вид, состояние или поведение;
- 2) имя модификатора определяется по шаблону: имя-блока_имя-элемента_имя-модификатора или имя-блока_имя-элемента_имя-модификатора_значение-модификатора и отделяется от имени блока или элемента одинарным подчеркиванием;
- 3) может быть булевый (важно только наличие или отсутствие модификатора) и ключ-значение (важно значение модификатора)

Так же можно использовать миксы, позволяющие совмещать поведение и стили нескольких сущностей без дублирования и файловую структуру, которая применяется к организации проекта.

Таким образом, БЭМ-методология уменьшает временные затраты на реализацию проекта, делает программную реализацию более понятной и прозрачной, что позволяет сократить денежные затраты и вероятность ошибок.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. БЭМ-методология – URL: <https://ru.bem.info/methodology/> (дата обращения: 8.11.2020).
2. BEM-methodology – URL: <http://getbem.com/introduction/> (дата обращения: 8.11.2020).
3. Организация CSS-кода – URL: https://developer.mozilla.org/ru/docs/Learn/CSS/Building_blocks/Organizing (дата обращения: 8.11.2020).

УДК 004

Гранкин А.Н., Крошилин А.В.

КОНЕЧНЫЕ АВТОМАТЫ В ПРОЕКТИРОВАНИИ ВЕБ-ИНТЕРФЕЙСОВ ИНФОРМАЦИОННЫХ СИСТЕМ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

В статье рассматривается использование конечных автоматов в качестве инструмента для управления состоянием веб-интерфейсов информационных систем.

Конечный автомат – это абстрактная модель из теории алгоритмов, которая представляет собой сущность, имеющую бесконечное или конечное число возможных состояний, а также точки входа и выхода. На вход обычно передаётся некоторая единица информации, меняющая состояние конечного автомата, а на выходе мы получаем новую информацию являющуюся продуктом работы конечного автомата. Если конечный автомат является детерминированным, то есть следующее состояние однозначно зависит от поступившей информации и текущего состояния, то точно можем понять какую информацию получим на выходе [2]. Таким образом конечные автоматы можно представить как изолированную систему, состояние которой извне неизвестно, а взаимодействовать с ней можем только посредством передачи и получения информации через точки входа и выхода соответственно (см. рис. 1).

Конечные детерминированные автоматы нашли своё применение в современной веб-разработке интерфейсов, такая тенденция объясняется технологиями, которые использует браузер для построения веб страниц. Одной из них является язык JavaScript, который из-за своей динамической слабой типизации является недостаточно надёжным инструментом для проектирования безопасных и сложных веб приложений. В этом случае шаблоны проектирования, основанные на детерминированных конечных автоматах используются для достижения двух целей: безопасного управления данными внутри изолированной системы и отделения этих данных

от представления [3]. Такой подход в разработке особенно актуален для информационных систем и других видов приложений, которые подразумевают большое число интерактивных элементов связанных с динамически изменяющимся данными. Для проектирования веб-приложений которые имеют статический интерфейс достаточным является использование монолитного подхода, когда данные и представление никак не разделены.

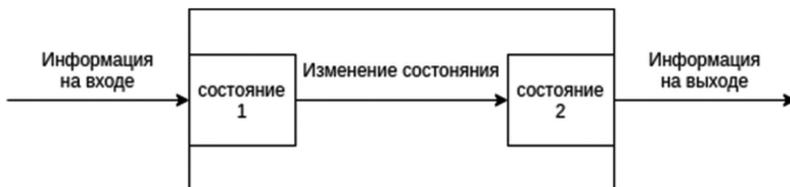


Рис. 1. Детерминированный конечный автомат

Реализовывать шаблоны проектирования основанные на конечных автоматах можно самостоятельно, в любом Тьюринг-полном языке программирования [2], однако ниже в этой статье будет описано уже готового решения, которое предоставляет библиотека Redux для языка JavaScript.

Redux состоит из трёх основных частей [1]:

1. «Store» – представляет из себя хранилище, которое содержит всю информацию необходимую для построения веб-интерфейса предложения.

2. «Action» – элемент, содержащий данные, которые нужно обработать, а так же название операции, необходимой для обработки этих данных.

3. «Reducer» – элемент который принимает «Action», и в зависимости от названия операции применяет ту или иную функцию изменения «Store», а так же возвращает результат выполнения этих изменений.

Алгоритм взаимодействия Redux с элементами интерфейса состоит из двух основных этапов: подписки и вызова [1]. Подписка представляет собой процесс установления связи между элементом интерфейса и хранилища. Событие вызова отправляет данные и название операции в хранилище, в ответ на некоторые установленные действия пользователя.

Важным моментом является то, что на каждое изменение данных в хранилище, мы должны отправлять ответ, а число возможных операций, которые мы можем применить к поступившей информации должно быть конечным. Если эти условия соблюдены, то можно утверждать, что шаблон проектирования основанный на конечных детерминированы автоматах реализован в полной мере.

Стоит отметить, что описанный выше подход позволяет отделить элементы от данных, однако для построения надёжного и расширяемого веб-интерфейса информационной системы необходимо так же изолировать компоненты между собой. Для решения этой задачи чаще всего прибегают к использованию библиотек для построения интерфейса. Одной из

цепочка символов языку *целых десятичных чисел со знаком*. Этот язык относится к регулярным языкам, т.к. существует *регулярная грамматика*, задающая этот язык [1]. Такой распознаватель можно реализовать разными средствами различных языков программирования. Мы, в качестве примера, реализовали распознаватель на языках PascalABC и C#.

Алфавит языка целых десятичных чисел со знаком – это множество *терминальных (основных) символов*: {0,1,2,3,4,5,6,7,8,9,-,+}. Для представления *алфавита терминальных символов* в программе на языке PascalABC используем тип данных «множество». Множество *терминальных символов* для удобства реализации в программе представим двумя множествами: *цифр* и *знаков*. Вначале проверяем первый символ введенной строки, если он входит в множество знаков или в множество цифр, то только тогда будет проверяться строка со второго символа. Программа-распознаватель на языке PascalABC:

Var

```
integerSet : set of char; // Множество терминальных символов - цифр
signSet : set of char; // Множество терминальных символов - знаков
newString : string; // исходная цепочка символов
resultString : string; // Строка с результатом распознавания цепочки
```

Begin

```
integerSet := ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'];
signSet := ['+', '-'];
resultString := 'Допустимая цепочка'; //инициализируем строку
Write('Введите исходную цепочку символов: ');
ReadLn(newString); // Ввод исходной цепочки
if not (Length(newString) = 0) // Если цепочка не пустая,
  then begin // то проверяем каждый её символ
    if ((newString[1] in signSet) or //Если первый символ цепочки –
      (newString[1] in integerSet)) // знак числа или цифра, то
      then // проверяем следующие символы
        for var i := 2 to Length(newString) do // Проверим все
          // символы цепочки, начиная со 2-го
          if not (newString[i] in integerSet) // Если хотя бы один из
            // символов (начиная со второго) цепочки не является
            then begin // цифрой, то это не допустимая цепочка
              resultString := 'Недопустимая цепочка';
              break; // Выход из цикла for
            end;
          end
        else // Если нет, то это недопустимая цепочка
          resultString := 'Недопустимая цепочка';
        end
      end
    else // Если цепочка пустая, то это не допустимая цепочка
```

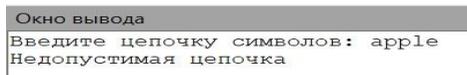
```
resultString := 'Недопустимая цепочка';  
WriteLn(resultString); // Выводим результат распознавания цепочки  
End.
```

Результаты выполнения программы на языке PascalABC:



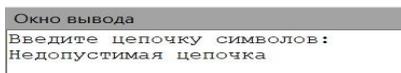
```
Окно вывода  
Введите цепочку символов: 137463747  
Допустимая цепочка
```

Рис. 1. Введена допустимая цепочка (целое число без знака)



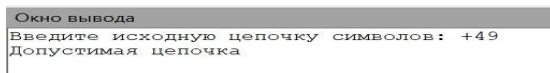
```
Окно вывода  
Введите цепочку символов: apple  
Недопустимая цепочка
```

Рис. 2. Введена недопустимая цепочка



```
Окно вывода  
Введите цепочку символов:  
Недопустимая цепочка
```

Рис. 3. Введена пустая цепочка символов



```
Окно вывода  
Введите исходную цепочку символов: +49  
Допустимая цепочка
```

Рис. 4. Введена допустимая цепочка (число со знаком)



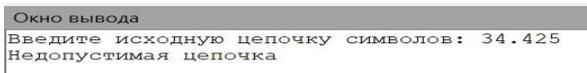
```
Окно вывода  
Введите исходную цепочку символов: 12+3  
Недопустимая цепочка
```

Рис. 5. Введена недопустимая цепочка (знак между цифрами числа)



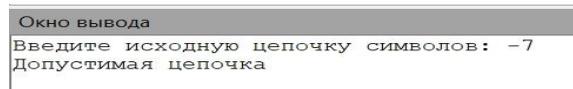
```
Окно вывода  
Введите исходную цепочку символов: 435-  
Недопустимая цепочка
```

Рис. 6. Введена недопустимая цепочка (знак в конце числа)



```
Окно вывода  
Введите исходную цепочку символов: 34.425  
Недопустимая цепочка
```

Рис. 7. Введена недопустимая цепочка (число с десятичной точкой)



```
Окно вывода  
Введите исходную цепочку символов: -7  
Допустимая цепочка
```

Рис. 8. Введена допустимая цепочка (отрицательное число)

Возможности современных языков программирования позволяют упростить реализацию данной задачи. Например, возможности языка C# [2] позволяют использовать механизм *регулярных выражений*. В этом

случае множество терминальных символов в явном виде в программе не задается, а представляется в виде *регулярного выражения*. Для использования регулярных выражений в языке C# необходимо в программе подключить пространство имен:

```
System.Text.RegularExpressions
```

и использовать *класс* и *объект* класса.

Для определения целого десятичного числа со знаком, необходимо создать объект (назовем его, например, **signedIntMask**) класса **Regex**:

```
Regex signedIntMask = new Regex(@"^[-+]?(\d+)$");
```

Выражение в скобках определяет *условия*, которым должна соответствовать проверяемая цепочка символов:

1) `^[-+]` – это выражение обозначает, что в начале цепочки должен стоять знак «-» или «+»;

2) знак «?» означает, что предшествующее выражение необязательно, т.е. знак числа может не указываться;

3) `\d` – обозначает, что должны использоваться только десятичные цифры;

4) знак «+» означает, что предшествующее выражение должно присутствовать один или более раз;

5) знак «\$» означает конец строки.

Собственно сама проверка цепочки символов выполняется методом **IsMatch** класса **Regex**.

Фрагмент программы на языке C# с использованием регулярного выражения:

```
// задаем массив test цепочек символов для проверки
string[] test = { "137463747", "apple", "", "+49", "12+3", "435-",
                 "34.425", "-7" }; // восемь цепочек символов
Regex signedIntMask = new Regex(@"^[-+]?(\d+)$");// создаем объект
//signedIntMask класса Regex
foreach (string str in test) // проверяем каждую цепочку str из
{
    //массива test и выводим результат
    Console.WriteLine($"Исходная цепочка: {str}. ");
    if (signedIntMask.IsMatch(str))
    { Console.WriteLine("Допустимая цепочка."); }
    else
    { Console.WriteLine("Недопустимая цепочка."); }
}
}
```

Результат выполнения программы на языке C#:

```

Исходная цепочка: 137463747. Допустимая цепочка.
Исходная цепочка: apple. Недопустимая цепочка.
Исходная цепочка: . Недопустимая цепочка.
Исходная цепочка: +49. Допустимая цепочка.
Исходная цепочка: 12+3. Недопустимая цепочка.
Исходная цепочка: 435-. Недопустимая цепочка.
Исходная цепочка: 34.425. Недопустимая цепочка.
Исходная цепочка: -7. Допустимая цепочка.

```

Рис. 9. Результат выполнения программы на языке С#

Также в языке программирования С# существует функция **TryParse**, которая является статическим методом структуры `Int32` и позволяет проверить, возможно ли преобразовать строку в число: `TryParse(string str, out int res)`, где `str` – входная строка, которая проходит проверку, `res` – число, которое получается в результате успешного преобразования, а сама функция возвращает логическое значение (`true` или `false`), которое означает, удалось преобразовать строку в целое число со знаком (`true`) или нет (`false`).

Фрагмент программы на С# с использованием функции TryParse:

```

int res;
foreach (string str in test) // массив test тот же самый, что и в
{ Console.WriteLine("Исходная цепочка: {str}."); //предыдущей программе
  if (Int32.TryParse(str, out res))
    { Console.WriteLine("Допустимая цепочка."); }
  else
    { Console.WriteLine("Недопустимая цепочка."); }
}

```

Результаты выполнения программы с функцией `TryParse` получились точно такие же, как и для предыдущей программы (рис. 9).

На примере этих данных можно сравнить скорость выполнения программ и выяснить, какая из них работает быстрее: с использованием **регулярных выражений** или с использованием функции **TryParse**. Проверка выполнялась на одном и том же компьютере, в одной и той же среде разработки. Повторим проверку этих восьми цепочек символов в цикле две тысячи раз:

```

Stopwatch stopwatch=new Stopwatch(); //Создаем новый объект
stopwatch.Start(); //Запускаем внутренний таймер объекта stopwatch
Regex signedIntMask = new Regex(@"^[+]?(\d+)$");
for (int i = 0; i < 2000; i++)
{ foreach (string str in test) // массив test тот же самый, что и в
  { // предыдущей программе
    if (signedIntMask.IsMatch(str)) { }
  }
}

```

```
}  
stopwatch.Stop(); // Останавливаем внутренний таймер  
double regexTime = stopwatch.ElapsedTicks / 10;  
Console.WriteLine($"Время работы программы с использованием  
регулярных выражений: {regexTime} мкс");
```

Аналогично этому вычислим время выполнения для программы с использованием функции **TryParse** для тех же самых исходных цепочек в цикле 2000 раз. Получим: программа с использованием метода TryParse языка C# в несколько раз быстрее программы, в которой используется механизм регулярных выражений, т.к. время ее выполнения составляет около 4 000 мкс, а время выполнения программы с регулярными выражениями занимает около 20 000 мкс.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. 3-е изд. – СПб.: Питер, 2010. – 400 с.
2. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. 4-е изд. – СПб.: Питер, 2013. – 896 с.

УДК 004.45

Никичкин Б.В., Горкин А.В., Ловкин М.В.

ПРОГРАММА-РАСПОЗНАТЕЛЬ ДЛЯ ЯЗЫКА ВЕЩЕСТВЕННЫХ ДЕСЯТИЧНЫХ ЧИСЕЛ С ФИКСИРОВАННОЙ ТОЧКОЙ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

Рассмотрены различные способы реализации программы-распознавателя для языка вещественных десятичных чисел.

Язык *вещественных десятичных чисел с фиксированной точкой* относится к *регулярным* языкам. *Алфавит* языка вещественных десятичных чисел – множество *терминальных (основных) символов*: {0,1,2,3,4,5,6,7,8,9, ., +, -}. Будем считать, что цепочки символов вида .35 или 9. *допускаются* в этом языке, т.е. запись ".35" эквивалентна "0.35", запись "9." эквивалентна "9.0". Символ '!' обязательно должен присутствовать в записи. Нами разработана программа-распознаватель, которая распознает, принадлежит (допустима) или не принадлежит (не допустима) заданная цепочка символов языку *вещественных десятичных чисел с фиксированной точкой*. Распознаватель реализован разными средствами языка программирования Java. Программу реализуем следующим образом:

- 1) множество *терминальных символов* языка определим в программе в виде *массива*;
- 2) проверим, состоит ли исходная цепочка символов из терминальных символов;

3) если состоит, то необходимо пройти с помощью цикла по всем символам исходной цепочки и проверить:

– если присутствует символ '+' или '-', то он должен находиться только в начале цепочки;

– если присутствует символ '.', то он должен быть единственным в цепочке.

Фрагмент программы-распознавателя на языке Java:

```
final char[] ALLOWED_SYMBOLS = { '1', '2', '3', '4', '5',
    '6', '7', '8', '9', '0',    '.', '+', '-' }; // определяем в виде массива
// множество терминальных символов языка
void isNumber(String input){
char[] characters=input.toCharArray();//формирование массива символов
int dotCount= 0; // счётчик точек во входной цепочке
boolean isValid = true; // флаг, показывающий, допустима ли цепочка
// или нет
if(characters.length > 0) {
    for (int i = 0; i < characters.length; i++) {
        char character = characters[i]; // текущий символ

        boolean isCharValid = checkCharacter(character); // проверка на
//наличие введенного символа в массиве терминальных символов
        if (!isCharValid) {
            isValid = false; // если символа нет в массиве, то
            break; // выходим из цикла
        } else {
            if (i >= 1) { // + или - могут быть только в начале строки
                if (character == '+' || character == '-') { // если '+' (или '-')
                    isValid = false; // не первый символ строки, то
                    break; // выход из цикла
                }
            }
            if (character == '.') {
                dotCount++;
                if (dotCount > 1) {
                    isValid = false;
                    break;
                }
            }
        }
    }
}
if (characters.length== 1 && characters[0] == '.' || characters.length > 1
&& dotCount == 0) {
    isValid = false;
}
```

```

    }
  }
  if (isValid && characters.length > 0) {
    System.out.println("Цепочка " + input + " допустима");
  } else {
    System.out.println("Цепочка " + input + " недопустима");
  }
}
// Функция, проверяющая наличие символа в массиве терминальных
// символов
boolean checkCharacter(char c){
  boolean isValid = false;
  for (int i = 0; i < ALLOWED_SYMBOLS.length; i++){
    if (c == ALLOWED_SYMBOLS[i]){ //если символ найден
      isValid = true;    // в массиве терминальных символов
      break;
    }
  }
  return isValid;
}
}

```

Метод для проверки правильности функционирования программы:

```

public static void main(String[] args) {
  Main main = new Main();
  main.run();
}

```

```

void run() {
  isNumber("1.2");
  isNumber("0.2");
  isNumber("-.2");
  isNumber("+.2");
  isNumber("1+.23");
  isNumber("+1.2");
  isNumber("2.");
  isNumber("-2.");
  isNumber("-1.-2");
  isNumber("-1");
  isNumber(".");
  isNumber("-1.15.3");
  isNumber("chislo");
  isNumber("");
}

```

Вывод программы:

```

Цепочка 1.2 допустима
Цепочка 0.2 допустима
Цепочка -.2 допустима
Цепочка +.2 допустима
Цепочка 1+.23 недопустима
Цепочка +1.2 допустима
Цепочка 2. допустима
Цепочка -2. допустима
Цепочка -1.-2 недопустима
Цепочка -1 недопустима
Цепочка . недопустима
Цепочка -1.15.3 недопустима
Цепочка chislo недопустима
Цепочка недопустима

```

Ту же функциональность можно реализовать с помощью высокоуровневых средств языка Java. Одним из таких средств являются *регулярные выражения* [1] в языке Java. Для того, чтобы определить, является ли входная строка вещественным десятичным числом, требуется выполнить следующие действия, содержащиеся в методе `isNumber(String numberString)`:

1) описать регулярное выражение: `^[+-]?([0-9]+[.][0-9]*?|[.][0-9]+)$`
 где: `^` – знак, обозначающий совпадение в начале строки; `[+-] ?` – один символ из множества `{'+', '-'}` встречается 0 или 1 раз;

`([0-9]+[.][0-9]*?|[.][0-9]+)` – [цифра (1 или более раз), после чего точка и цифра (0 или более раз)] или [точка и цифра(1 или более раз)];

`$` – знак, обозначающий совпадение в конце строки;

2) создать объект класса `Pattern`, используя статический метод `compile`, куда передается текст регулярного выражения;

3) создать объект класса `Matcher`, используя метод `matcher` у `Pattern`-объекта, куда передается входная (проверяемая) строка;

4) используя метод `find` объекта класса `Matcher`, определить, является ли входная строка вещественным числом или нет.

Фрагмент программы на языке Java с регулярными выражениями:

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

boolean isNumber(String numberString) {
    String decimalRealRegex = "^[+-]?([0-9]+[.][0-9]*?|[.][0-9]+)$";
    Pattern pattern = Pattern.compile(decimalRealRegex);
    Matcher matcher = pattern.matcher(numberString);
    if (matcher.find()) {
        return true;
    } else {
        return false;
    }
}

void test(String input) {
    boolean isInputNumber = isNumber(input);
    if (isInputNumber) {
        System.out.printf("цепочка %s допустима%n", input);
    } else {
        System.out.printf("цепочка %s недопустима%n", input);
    }
}
```

<p>Метод для проверки правильности функционирования программы:</p> <pre>public static void main(String[] args) { Main main = new Main(); main.run(); } void run() { test("1.2"); test("0.2"); test("-.2"); test("+.2"); test("1+.23"); test("+1.2"); test("2."); test("-.2"); test("-1.-2"); test("-1"); test("."); test("-1.15.3"); test("chislo"); test(""); }</pre>	<p>Вывод программы:</p> <pre>Цепочка 1.2 допустима Цепочка 0.2 допустима Цепочка -.2 допустима Цепочка +.2 допустима Цепочка 1+.23 недопустима Цепочка +1.2 допустима Цепочка 2. допустима Цепочка -.2 допустима Цепочка -1.-2 недопустима Цепочка -1 недопустима Цепочка. недопустима Цепочка -1.15.3 недопустима Цепочка chislo недопустима Цепочка недопустима</pre>
--	---

Существуют также уже готовые средства языка Java для выполнения этой задачи. Одним из них является статическая функция **parseDouble**(String s) класса Double. Если входная строка не является вещественным числом, то функция генерирует исключение NumberFormatException. Пример программы с функцией parseDouble:

```
void test(String input){
    try {
        Double.parseDouble(input);
        System.out.printf("Цепочка %s допустима%n", input);
    } catch (NumberFormatException e){
        System.out.printf("Цепочка %s недопустима%n", input);
    }
}
```

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Хорстманн Кей. Java. Библиотека профессионала. Том 2. Расширенные средства программирования. 10-е изд. СПб.: «Альфа-книга», 2017.

УДК 005.519.8**Головкин Н.В., Цуканова Н.И.****РАЗРАБОТКА НОУТБУКА В СРЕДЕ GOOGLE COLAB
ПО ТЕМЕ «КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ
С ПОМОЩЬЮ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ»****Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань**

Рассматривается ноутбук, описывающий методику классификации изображений с помощью глубоких нейронных сетей, разработанных на языке Python

В век постиндустриального общества с каждым годом появляется все больше различных технологий, целью которых является облегчение жизни человека, избавление его от рутинных, монотонных действий, которые могут быть автоматизированы.

Не так давно компания Google представила сервис Google Colab, позволяющий любому желающему бесплатно использовать технические мощности компании в исследовательских целях. Он позволяет разрабатывать ноутбуки – документы, которые могут содержать ячейки как с компьютерным кодом на языке программирования Python, так и с текстовыми элементами, и выполнять содержащийся в них код с применением мощных графических (GPU) и тензорных (TPU) процессоров, которые, в частности, существенно повышают скорость обучения глубоких нейронных сетей.

В настоящее время большое внимание уделяется решению практических задач с помощью глубоких нейронных сетей. Одной из таких задач является задача классификации изображений. Она состоит в том, чтобы присвоить конкретному примеру одну из множества различных меток. Примерами такой задачи могут служить распознавание лиц камерами на улицах города в целях идентификации людей или распознавание образов беспилотными автомобилями, однако, сфера применения не ограничивается только ими.

В настоящей статье рассматривается задача классификации изображений одежды. Решение данной задачи выполнено в виде ноутбука.

Ноутбук как учебник по применению нейронных сетей к решению практических задач может создаваться в соответствии со следующим планом:

- постановка задачи;
- описание, понимание и исследование набора данных, предназначенного для обучения нейронной сети;
- предобработка набора данных (особенно текста);
- формирование нейронной сети, определение ее архитектуры и гиперпараметров;
- обучение нейронной сети;
- оценка качества нейронной сети с графической иллюстрацией результатов оценки;

– применение обученной нейронной сети к новым данным.

В качестве обучающего набора был выбран набор Fashion-MNIST. Он состоит из черно-белых (всего 256 тонов) изображений одежды размером 28 x 28 пикселей. Всего в наборе 70000 экземпляров, где 60000 из них приходится на выборку для обучения, а 10000 – на тестовую выборку. Каждому изображению присвоен 1 из 10 возможных классов. Каждому классу соответствует 7000 изображений.

Немаловажную роль при работе с данными для обучения нейронной сети является их преобработка. Поскольку нейронная сеть работает только с информацией, представленной в числовом виде, то преобработка включала в себя следующие шаги: преобразование изображения в трехмерный тензор вида (ширина, высота, количество каналов изображения), где каждой комбинации ширины, высоты и количества каналов соответствовало значение от 0 до 255; нормализация данных – преобразование значений в полученных тензорах в значения от 0 до 1; преобразование меток в формат One Hot Encoding. Каждый из этих шагов описан и разъяснен в текстовых полях ноутбука.

Для классификации изображений существует большое количество разнообразных методов. В работе рассмотрена классификация изображений автокодировщиком и полносвязной нейронной сетью.

Решение задачи выполняется в два этапа. На первом этапе автокодировщик используется для предобучения нейронной сети, а на втором - к автокодировщику добавляется полносвязная нейронная сеть для классификации.

Автокодировщик – это модель глубокого обучения, состоящая из кодировщика и декодировщика, которая преобразует данные из многомерного пространства в малоразмерное (в кодировку). Он работает, кодируя данные, независимо от их размера, в одномерный вектор, а затем декодирует его, чтобы получить исходные данные. При обучении автокодировщика ставится цель добиться как можно большей близости данных на выходе к исходным данным [1]. Благодаря этому на выходном слое кодировщика определяются самые важные (признаки) объектов обучающей выборки. Размерность вектора признаков значительно меньше размерности входного изображения. Можно сказать, что автокодировщик обобщает информацию.

Теперь эти важные признаки можно подать на полносвязную нейронную сеть и решить задачу классификации [2]. Для этого часть, соответствующая кодировщику, замораживается (ставится запрет на изменение его весовых коэффициентов), к ней присоединяются полносвязные слои и вновь созданная сеть обучается [2].

Кодировщик в данной работе имеет 4 сверточных блока. Каждый блок имеет сверточный слой, за которым следует слой нормализации. Слой выбора максимального значения из соседних (MaxPooling) исполь-

зуется после первого и второго блоков свертки. Декодировщик состоит из 3-х свёрточных блоков. Каждый блок имеет свёрточный слой, за которым следует слой нормализации. После второго и третьего блоков свертки используется слой передискретизации (UpSampling).

Полносвязная нейронная сеть состоит из двух слоев – входного и выходного. В качестве активационной функции входного слоя используется *relu*, а активационной функции выходного слоя – *softmax*.

Классификация изображений с использованием автокодировщика показала более хорошие результаты. В сравнении с полносвязной нейронной сетью, разница в точности на тестовой выборке составила ~4 % (87 % и 91 %).

Обучение автокодировщика заняло 2,5 часа, в то время как полносвязной нейронной сети – 2 минуты.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем / пер. с англ. — СПб.: ООО «Альфа-книга», 2018. — 688 с.

2. Шитова К.Г., Цуканова Н.И. О применении глубоких нейронных сетей к оценке кредитоспособности предприятия. // Вестник Рязанского государственного радиотехнического университета. 2019. № 68. С.44-53

УДК 681.3

Старикова К.Ю., Крошила С.В., Крошила А.А.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ НА ПРЕДПРИЯТИЯХ МАЛОГО БИЗНЕСА

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

В статье рассматриваются возможности эффективного ведения малого бизнеса на основе внедрения информационных технологий.

Важным условием повышения эффективности деятельности малого предприятия является внедрение современных информационных технологий (ИТ). С их помощью возможно наиболее эффективно проводить глубокий всесторонний анализ эффективности деятельности, прогнозировать для принятия управленческих решений в сфере малого предпринимательства.

Технологии для малого бизнеса являются возможностью применять ограниченный бюджет рациональными и результативными методами. Значительная часть предпринимателей стараются применять информационные технологии уже при основании малого бизнеса ввиду того, что эффективность в получении прибыли и универсальность благ строятся на первоначальном этапе.

Внедрение и развитие ИТ приводит к трансформации рынков, что требует от руководителей переосмысления бизнес-моделей компании и определения новых возможностей развития бизнеса. Малый бизнес зани-

мает ведущие позиции в сфере услуг и служит площадкой в целях внедрения инноваций в различных сферах деятельности.

Малый бизнес стимулирует производителей ИТ создавать специализированные продукты, предусматривающие особенность деятельности и применение информационных технологий этой категорией субъектов экономики [1].

Рассмотрим некоторые распространенные программы, которые использует значительная часть малых предприятий. Одной из наиболее востребованной и используемой программой в современном предпринимательстве является пакет Microsoft Office. Основные приложения пакета это MS Word, MS Excel, MS Office, MS Access, MS Power Point и MS Outlook. С помощью данного функционала возможно анализировать, обрабатывать данные, быстро и оперативно отслеживания информацию, подготавливать отчеты, совместно использовать данных посредством сети Интернет [2].

Рассмотренное программное решение обеспечивает безопасную вычислительную инфраструктуру деятельности, обеспечивает надежную инфраструктуру на основе технологий для бизнеса, также позволяет минимизировать стоимость ИТ-инфраструктуры.

Отдельным столбом в софте для введения бизнеса, в том числе и малого, стоит продукт 1С:Предприятие. Гибкость системы предоставляет возможность применить данный продукт в разных сферах ведения бизнеса: автоматизация малых предприятий различных сфер деятельности; оперативность управления организацией; автоматическое управление хозяйственной деятельности предприятия; предоставление анализа всех сфер деятельности предприятия; расчет заработной платы и управление персоналом и т.д. [3].

Отметим, что внедрение информационных технологий на предприятиях малого бизнеса значительно способствует процессу их развития. Для того, чтобы использование ИС было результативным, необходимо внедрение ИТ-систем во все сферы его деятельности. Данный подход создает фундамент для успешного дальнейшего развития малого бизнеса.

Для каждого отдельного предприятия важно выбирать программные продукты, принимая во внимание специфику деятельности, возможности и потребности. Неоспоримым является тот факт, что информационные технологии незаменимы абсолютно во всех сферах бизнеса. В настоящее время существует огромное количество решений для любого вида деятельности. Стремительно растущий потенциал информационных технологий обеспечивает столь же стремительное снижение издержек на малом предприятии, увеличение доходов и достойные позиции среди конкурентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Булгакова М.В. Информационное обеспечение малого бизнеса в современных условиях // Вестник ЮУрГУ. Серия Компьютерные технологии, управление, радиоэлектроника // 2015, с. 154-159.

2. Svetlana Kroshilina, Valeriia Tishkina, Aleksandr Pylkin, Aleksandr Kroshilin, Alexei Evseev Enterprise Management Mobile Assistant based on Using the Theory of Fuzzy Logic and Fuzzy Sets // 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA), pp. 247-249.

3. Провалов В.С. Информационные технологии в малом бизнесе: особенности использования // Естественные и математические науки в современном мире: Сб. ст. по материалам XIX междунар. науч.-практ. конф. №6 (18). 2014, с. 43-48.

УДК 004

Захаркин А.В.

ТЕНДЕНЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В 2020 ГОДУ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

ИТ-индустрия становится неотъемлемой частью жизни всех сфер деятельности, благодаря чему разработка программного обеспечения постоянно совершенствуется и изменяется. Рассмотрим тенденции разработки программного обеспечения в 2020 году.

Говоря об «облачном сервисе», хочется отметить, что рост его популярности прогнозировался еще в 2019 году. Кроме того, из-за появления пандемии в мире и перевод на удаленную работу сотрудников, понизил спрос на традиционные сегменты ИТ, такие как сервера и ПО для локальных ЦОДов, и дал существенный рывок для облачных сервисов [6].

Говоря о языках программирования, хочется отметить стремительный рост Python. Его важность особенно подчеркивается в машинном обучении, искусственном интеллекте и анализе данных. По версии сервиса ТЮВЕ, данный язык по популярности обходят лишь Java и C: [3]. Так же наблюдается укрепление позиций языка программирования Rust. Этот язык хорош тем, что особое внимание в нём уделяется безопасной работе [5].

В 2020 году очень стала популярна разработка приложений с низким уровнем кода. Разработка происходит через интерфейсы программирования, пользователю остается лишь использовать готовые компоненты, а код создается автоматически. Стали они популярны из-за улучшенной функциональности – широкими возможностям интеграции и внедрению AI/ML, а так же благодаря поддержке микросервисов. Наиболее популярными на сегодняшний день малокодовыми платформами являются PowerApps и Power Automate – корпоративные разработки от Microsoft [4, 6, 7].

В уходящем году выросло использование платформ с открытым исходным кодом, например, таких как Docker и Kubernetes. Open source проекты стали популярны еще в 2019 году. В 2020-ом применение таких проектов стало еще более распространённое. Открытый код становится источником инноваций в сфере программного обеспечения, а системы с open source удерживают лидерство в корпоративном секторе. Microsoft тоже активно поддерживает использование таких платформ, например подсистема Windows для Linux (WSL) имеет открытый исходный код [6].

В 2020 году вырос спрос на прогрессивные приложения PWA. Эта технология переносит мобильное приложение в браузер, сохраняя его внешний вид и функционал. На апрель 2020 года офлайн-работу с PWA могут обеспечить iOS, Android, Windows, Linux, macOS и Chrome OS в браузерах Chrome, Safari, Firefox, Edge и Samsung Internet. Интернет-магазины активно используют PWA. Крупнейший из них – сайт AliExpress. Внедрение данной технологии помогло площадке добиться вдвое больше посещений страниц за сеанс. Время сеанса увеличилось в среднем на 74 % во всех браузерах [4,5].

Программное обеспечение AI-first так же набирает обороты. Искусственный интеллект знаменит такими технологиями, как распознавание речи, машинное зрение и глубокое обучение. На основе AI разрабатываются виртуальные помощники, способные определять эмоции покупателей. Такие помощники применяются в маркетинговой сфере для того, чтобы оценивать уровень обслуживания. В сельском хозяйстве ИИ могут определять состояние растений и находить сорняки [1, 2, 4].

На конференции Build 2020 компания Microsoft представила одного из пяти самых мощных суперкомпьютеров в мире. Его планируют использовать для тренировки следующего поколения массивных распределительных ИИ-моделей на базе организации OpenAI [6].

REST API теряет свою популярность из-за медленной загрузки данных, которую он выполняет с нескольких URL-адресов по отдельности. Новой альтернативой архитектуре REST является GraphQL. GraphQL — это язык запросов, используемый клиентскими приложениями для работы с данными. Он извлекает все релевантные данные с нескольких сайтов с помощью одного запроса. Он требует меньше кода, позволяя тем самым совершать сложные запросы в несколько простых строк [5, 7].

В 2017 году был создан новый язык программирования Q#. К 2020 году он существенно набирает обороты. Ожидается создание компьютеров, которые смогут взламывать наименее стойкие традиционные шифры. Ведущие мировые ИТ-вузы начали обучение специалистов для работы с квантовыми компьютерами. В России такое обучение проводит МГУ, Университет Иннополис, ИТМО и Политех. На конференции Build 2020 Microsoft презентовала новые бесплатные учебные модули для платформы Microsoft Learn, которые позволяют пройти обучение по работе с

квантовыми вычислениями с использованием языка программирования #Q и Quantum Development Kit [6].

Делая выводы, хочется отметить, что разработка программного обеспечения становится обширнее не только среди потребителей, но и среди разработчиков. Рынок динамично развивается и расширяется, идет тенденция на упрощение разработки ПО, экономии времени и материальной составляющей. IT-компаниям нужно двигаться вперед, осваивая новые технологии ИИ, VR, AR, MR, развивающиеся языки программирования, а так же автоматизацию процессов с использованием AI.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бессмертный И.А. Системы искусственного интеллекта : учеб. пособие для СПО / И. А. Бессмертный. – 2-е изд., испр. и доп. – М. : Издательство Юрайт, 2018.
2. Иванов В.М. Интеллектуальные системы: учеб. пособие для СПО / В.М. Иванов; под науч. ред. А.Н. Сесекина. – М.: Издательство Юрайт, 2019.
3. Федоров Д.Ю. Программирование на языке высокого уровня python: учеб. пособие для СПО / Д.Ю. Федоров. – М.: Издательство Юрайт, 2019.

УДК 004.457

Попков И.А.

АЛГОРИТМЫ И ОПТИМИЗАЦИЯ ОБМЕНА ПЛАТЕЖНЫХ ДАННЫХ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

Упрощение обмена платежных данных клиентов с помощью QR кода и его преимущества.

При работе с информационной системой 1С [1], в которой нет унифицированной интегрированной системы обмена данными [2], сотрудники организации и физические лица часто сталкиваются с проблемами заполнения платежных данных для оплаты услуг электросетевой компании. Поэтому тема упрощения и ускорения обмена платежными данными между организацией и банком для физических лиц важная задача, которая требует своевременного решения. Создание алгоритма, который упрощает передачу платежных данных физических лиц от организации к банку и информации о платеже от банка к организации, сильно ускорит и упростит процедуру оплаты услуг электросетевой организации для физического лица.

Для выполнения оптимизации процесса обмена платежных данных между типовыми продуктами компании 1С и банком необходимо исследовать алгоритмы анализа данных [3]. В интеллектуальном анализе данных (или машинном обучении) *алгоритм* – это набор эвристики и вычислений, который создает на основе данных модель. Чтобы создать модель,

алгоритм сначала анализирует предоставленные данные, осуществляя поиск определенных закономерностей и тенденций. Интеллектуальный анализ данных можно выполнить с относительно скромными системами баз данных и простыми инструментами. Платформа 1С располагает достаточным набором инструментов для осуществления интеллектуального анализа данных.

Рассматривая методы и решения в сфере обмена данными, можно обнаружить очень удобный и современный способ передачи информации. QR-код – это одно из направлений дополненной реальности. Она объединяет реальные параметры с виртуальными. В настоящее время существует огромное количество специальных сервисов, позволяющих в несколько кликов сгенерировать такой код.

Технология оплаты по QR-коду для России новая, но в других странах подобные решения существуют довольно давно. Многие платежные системы предоставляют подобные сервисы. Отмечая преимущества данного вида передачи платежных данных для организации (продавца) можно выделить универсальность для всех видов платежей, скорость передачи данных в банк и возврат информации о платежах, экономию ресурсов.

Данный вид передачи данных необходим для интеграции в процесс оплаты услуг электросетевой организации. В основном, покупатели услуг проживают в малонаселенных городских поселениях Рязанской области. Для них данный вид передачи информации полезен тем, что с его помощью можно практически полностью исключить томительные ожидания заполнения данных в банках, множественную сверку данных, а так же упростить сам процесс оплаты услуг.

Выбор правильного алгоритма для использования в конкретной аналитической задаче может быть достаточно сложным, поэтому разработка унифицированной системы обмена данных для электросетевой организации является не простым, однако важным направлением.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Профессиональная разработка в системе «1С:Предприятие 8»: в 2 т. / В.А. Ажеронок, А.П. Габец, Д.И. Гончаров, Д.В. Козырев, Д.С. Куклевский, А.В. Островерх, М.Г. Радченко, Е.Ю. Хрусталева. – 2-е изд. – М.:1С-Паблишинг, 2012.

2. Arenas, M.; Barceló, P.; Libkin, L.; Murlak, F. (2014). Foundations of data exchange. Cambridge University Press.

3. Структуры данных и алгоритмы (2000). Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман.

4. IT-Блог AndreyEx. Тенденции разработки программного обеспечения 2020 года: преимущества и сценарии использования [Электронный ресурс] // IT-Блог AndreyEx, 2020. – Режим доступа: URL: <https://andreyex.ru/programmnoe-obespechenie/tendentsii-razrabotki-programmnogo-obespeche>

niya-2020-goda-preimushhestva-i-stsenarii-ispolzovaniya/ (Дата обращения 13.11.2020)

5. RUVDS. Тренды программирования: чего можно ожидать в 2020 году? [Электронный ресурс]//Сообщество IT-специалистов Хабр, 2019. – Режим доступа: URL: <https://habr.com/ru/company/ruvds/blog/462009/> (Дата обращения 13.11.2020)

6. Кристина Рудич, Артур Киреев. 7 трендов в разработке ПО: что изучать программистам в 2020 г.

7. Электронный ресурс]//СМИ «Хайтек», 2020. – Режим доступа: URL: <https://hightech-fm.turbopages.org/hightech.fm/s/2020/06/01/7trends-microsoft> (Дата обращения 13.11.2020).

8. Kristen Carter. Top 5 Software Development Practices to Follow in 2020 [Электронный ресурс]//Сообщество IT-специалистов Хабр, 2019. – Режим доступа: URL: <https://habr.com/ru/post/476308/> (Дата обращения 13.11.2020).

УДК 004.05

Самохина Л.А.

КРЕДИТНЫЕ СИСТЕМЫ И ОНЛАЙН ПРОДУКТЫ БАНКОВ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина г. Рязань

Кредит является неотъемлемой частью современного общества, он позволяет значительно снизить время для удовлетворения хозяйственных и личных потребностей субъекта. В статье рассматриваются понятие кредитной системы, существующие продукты банков и пути их развития.

Понятие кредитной системы [2] можно истолковать как совокупность банковских финансовых учреждений, позволяющих обеспечивать кредитные отношения. Кроме того, под кредитной системой могут пониматься непосредственно кредитные отношения, выраженные различными формами кредита и методами кредитования. Банковская система является частным случаем кредитной системы и включает в себя совокупность банков страны.

К фундаментальным основам системы кредитования относятся объект и субъект кредита, условия. С точки зрения экономики выделяют такие аспекты, как виды, методы кредитования, документация, процедура выдачи, порядок погашения. С точки зрения организации можно выделить содержание этапов кредитования, управление кредитом и рисками.

Основным элементом кредитной системы является Центральный банк, осуществляющий руководство над финансовой деятельностью страны. Он обеспечивает регулирование кредита и денежного обращения внутри страны, контроль обменного курса национальной валюты. Центральный банк [3] является представителем правительства. Основная функция центрального банка – разрабатывать и проводить кредитно-

денежную политику. Центральный банк России контролирует деятельность всех кредитных организаций, выдаёт и отзывает у них лицензии на осуществление банковских операций, диктует определенные процентные ставки, под которые может выдаваться кредит. Все остальные банки должны придерживаться политики Центрального банка.

Одними из крупнейших банков России являются Сбербанк, ВТБ, Тинькофф. В рейтинге по количеству отделений, сумме вкладов и кредитов они занимают лидирующие позиции. Перспективным направлением развития для банков в настоящая момент является осуществление операций онлайн. Особенно актуальным это стало в условиях пандемии. Основные удаленные каналы, в которых могут проводиться банковские операции – непосредственно отделение банка, веб и мобильные приложения, POS-терминалы.

У каждого из перечисленных банков есть различные программные системы, позволяющие привлекать клиентов онлайн. Улучшение веб и мобильных приложений банка является одним из способов их конкуренции. Особенностью банковских программных систем является их высокая нагруженность и критичность, к таким системам предъявляются высокие эксплуатационные требования по надежности, безопасности, отказоустойчивости.

Основным сценарием является подача кредитной заявки пользователем в приложении или веб-версии банка. Пользователь, клиент банка, заполняет необходимые параметры: сумму и срок кредита, анкету со своими данными и отправляет заявку на кредит на рассмотрение в банк. Как правило, решение по кредиту формируется в течение нескольких минут. Скорость получения ответа от банка является важным показателем как для банка, так и для клиента.

Пользовательский путь состоит из нескольких шагов, на каждом из которых клиент может совершать различные действия, например, перейти к следующему шагу, вернуться на предыдущий или выйти из приложения. Анализ того, на каком шаге клиент совершил то или иное действие, может помочь совершенствовать программный продукт в будущем и понять в каком направлении двигаться при его разработке.

Одним из важнейших показателей для бизнес-мониторинга является конверсия подачи заявки, которая представляет собой отношение числа пользователей, которые отправили заявку на кредит, к числу всех пользователей. Для улучшения банковских онлайн продуктов и увеличения количества операций ИТ-специалисты проводят различные UX (User Experience) исследования, A/B тестирование, осуществляют мониторинг приложения [1].

Для осуществления бизнес-мониторинга приложения руководство определяет список метрик, которые необходимо собирать для анализа работы приложения. Одними из самых популярных систем сбора метрик

являются сервисы «Яндекс.Метрика» и Google Analytics. У них есть как преимущества, так и недостатки. Главным критерием по выбору системы мониторинга для банка является конфиденциальность передаваемых в эти системы данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Клифтон Брайан. Google Analytics для профессионалов. – ООО «И.Д. Вильямс», 2017. – 608 с.
2. Лаврушин О.И. Банковское дело: современная система кредитования: учебное пособие. – КНОРУС, 2007. – 264 с.
3. Федеральный закон от 10.07.2002 № 86-ФЗ «О Центральном Банке Российской Федерации (Банке России)» (ред. от 26.04.2007).

УДК 004

Лукьянова Т.А.

АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ В DATA MINING

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

Рассматриваются наиболее популярные алгоритмы кластеризации данных. Приведены классификации алгоритмов. Целью работы является помощь в ориентировании в большом количестве алгоритмов кластеризации.

Ключевые слова: алгоритм, кластеризация данных, классификация данных.

Введение

Кластеризация – объединение в группы схожих объектов – является одной из фундаментальных задач в области анализа данных и Data Mining. Список прикладных областей, где она применяется, широк: сегментация изображений, маркетинг, прогнозирование, анализ текстов и многие другие. На современном этапе кластеризация часто выступает первым шагом при анализе данных. После выделения схожих групп применяются другие методы.

Задачу кластеризации в том или ином виде формулировали в таких научных направлениях, как статистика, распознавание образов, оптимизация, машинное обучение.

Кластеризация в Data Mining

Очень часто данные, с которыми сталкивается технология Data Mining, имеют следующие важные особенности:

- высокая размерность и большой объем таблиц баз данных и хранилищ данных;
- наборы данных содержат большое количество числовых и категориальных атрибутов.

Все атрибуты или признаки объектов делятся на числовые и категориальные. Числовые атрибуты могут быть упорядочены в пространстве, соответственно категориальные – не могут быть упорядочены.

Потребность в обработке больших массивов данных в Data Mining привела к формулированию требований, которым, по возможности, должен удовлетворять алгоритм кластеризации. Рассмотрим их:

- 1) минимально возможное количество проходов по базе данных;
- 2) работа в ограниченном объеме оперативной памяти компьютера;
- 3) работу алгоритма можно прервать с сохранением промежуточных результатов, чтобы продолжить вычисления позже;
- 4) алгоритм должен работать, когда объекты из базы данных могут извлекаться только в режиме однонаправленного курсора.

Классификация алгоритмов

Выделено две основные классификации алгоритмов кластеризации.

1. Иерархические и неиерархические.

Классические иерархические алгоритмы работают только с категориальными атрибутами, когда строится полное дерево вложенных кластеров. Здесь распространены агломеративные методы построения иерархий кластеров – в них производится последовательное объединение исходных объектов и соответствующее уменьшение числа кластеров. Иерархические алгоритмы обеспечивают сравнительно высокое качество кластеризации и не требуют предварительного задания количества кластеров.

Неиерархические алгоритмы основаны на оптимизации некоторой целевой функции, определяющей оптимальное в определенном смысле разбиение множества объектов на кластеры.

2. Непересекающиеся и пересекающиеся.

Непересекающиеся алгоритмы каждому объекту выборки ставят в соответствие номер кластера, т.е. каждый объект принадлежит только одному кластеру. Пересекающиеся алгоритмы каждому объекту ставят в соответствие набор вещественных значений, показывающих степень отношения объекта к кластерам.

Алгоритмы квадратичной ошибки

Задачу кластеризации можно рассматривать как построение оптимального разбиения объектов на группы. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X, L) = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^{(j)} - c_j\|^2, \quad (1)$$

где c_j – «центр масс» кластера j (точка со средними значениями характеристик для данного кластера).

Алгоритмы квадратичной ошибки относятся к типу плоских алгоритмов. Самым распространенным алгоритмом этой категории является метод k -средних. Этот алгоритм строит заданное число кластеров, расположенных как можно дальше друг от друга. Работа алгоритма делится на несколько этапов:

- 1) выбирается число кластеров k ;
- 2) из исходного множества данных случайным образом выбираются k наблюдений, которые будут служить начальными центрами кластеров;
- 3) для каждого наблюдения исходного множества определяется ближайший к нему центр кластера. При этом записи, «притянутые» определенным центром, образуют начальные кластеры;
- 4) вычисляются центроиды – центры тяжести кластеров. Каждый центроид – это вектор, элементы которого представляют собой средние значения соответствующих признаков, вычисленные по всем записям кластера;
- 5) центр кластера смещается в его центроид, после чего центроид становится центром нового кластера;
- 6) 3-й и 4-й шаги итеративно повторяются. На каждой итерации происходит изменение границ кластеров и смещение их центров. В результате минимизируется расстояние между элементами внутри кластеров и увеличиваются междукластерные расстояния.

Остановка алгоритма производится тогда, когда границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации. На практике алгоритм обычно находит набор стабильных кластеров за несколько десятков итераций.

Преимуществом алгоритма являются скорость и простота реализации. Недостатки – неопределенность выбора начальных центров кластеров, а также то, что число кластеров должно быть задано изначально.

Нечеткие алгоритмы

Наиболее популярным алгоритмом нечеткой кластеризации является алгоритм c -средних (c -means). Он представляет собой модификацию метода k -средних. Шаги работы алгоритма:

- 1) выбрать начальное нечеткое разбиение n объектов на k кластеров путем выбора матрицы принадлежности U размера $n \times k$;
- 2) используя матрицу U , найти значение критерия нечеткой ошибки:

$$E^2(X, U) = \sum_{i=1}^N \sum_{k=1}^K U_{ik} \|x_i^{(j)} - c_k\|^2, \quad (2)$$

где $c_k = \sum_{i=1}^N U_{ik} x_i$ – «центр масс» нечеткого кластера k .

- 3) перегруппировать объекты с целью уменьшения этого значения критерия нечеткой ошибки;
- 4) возвращаться в п. 2 до тех пор, пока изменения матрицы U не станут незначительными.

Этот алгоритм может не подойти, если заранее неизвестно число кластеров, либо необходимо однозначно отнести каждый объект к одному кластеру.

Алгоритмы, основанные на теории графов

Суть таких алгоритмов заключается в том, что выборка объектов представляется в виде графа $G = (V, E)$, вершинам которого соответствуют объекты, а ребра имеют вес, равный «расстоянию» между объектами.

Достоинством графовых алгоритмов кластеризации являются наглядность, относительная простота реализации и возможность внесения различных усовершенствований, основанные на геометрических соображениях. Основными алгоритмам являются алгоритм выделения связанных компонент, алгоритм построения минимального покрывающего дерева и алгоритм послойной кластеризации.

Заключение

Приведены классификации алгоритмов кластеризации данных и рассмотрены наиболее популярные из них. Таким образом, не существует единого универсального алгоритма кластеризации. При использовании любого алгоритма важно понимать его достоинства и недостатки, учитывать природу данных, с которыми он лучше работает и способность к масштабируемости.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Bradley P., Fayyad U., Reina C. Scaling Clustering Algorithms to Large Databases, Proc. 4th Int'l Conf. Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, Calif., 1998.

2. Milenova B., Campos M. Clustering large databases with numeric and nominal values using orthogonal projections, Oracle Data Mining Technologies, 2002.

3. Ganti V., Gerhke J., Ramakrishnan R. CACTUS – Clustering Categorical Data Using Summaries. In Proc KDD'99, 1999.

4. Добыча данных в сверхбольших базах данных / В. Ганти, Й. Герке, Р. Рамакришнан // Открытые системы, №9-10, 1999.

5. Барсегян и др. Методы и модели анализа данных: OLAP и Data Mining. – СПб., 2004.

УДК 004

Алексашкин В.В., Баконин Р.В., Кирилова Л.М.

ФЕЙКОВЫЕ НОВОСТИ В СОВРЕМЕННОМ ОБЩЕСТВЕ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

В современном обществе СМИ получили очень большое влияние. Они формируют мнение людей об окружающем мире и о происходящих событиях. В связи с этим, СМИ стали выгодным местом для борьбы различного вида. Например, политической или финансовой. А фейковые новости – это сообщение, стилистически созданное как настоящая новость, но ложное полностью или частично, созданное для намеренного искажения фактов, распространения дезинформации с целью получения выгоды. Всё больше СМИ стало своеобразной площадкой борьбы между враждующими силами. Зачастую СМИ могут быть даже площадкой для противостояния между странами.

Поэтому все больше растет потребность в выявлении фейковых новостей. Наиболее острой проблема стала в 2016 году и к данному моменту официальные источники исследований по кибербезопасности сообщают, что ежегодно из-за заведомо ложных онлайн-новостей мировая экономика теряет по \$78 млрд.

Поиск подобной информации вручную становится невозможен из-за огромного объема недостоверного контента, представленного в сети интернет, поэтому его вытесняют разнообразные алгоритмы отсеивания данных с использованием искусственного интеллекта и машинного обучения. Анализ позволяет выделить пять видов фейковых новостей, которые в свою очередь так же делятся на подкатегории:

- в зависимости от соотношения достоверной и недостоверной информации;
- в зависимости от достоверности обстоятельств времени и места произошедшего события;
- в зависимости от состава лиц, упоминаемых в «новости»;
- в зависимости от целей создания и распространения;
- в зависимости от уровня восприятия достоверности.

После проработки большого пласта фейковых новостей, можно выделить основные особенности. Часто фальшивые новости выявляются с помощью нахождения определенных лексических конструкций, например, отсутствие ссылки на источник информации, наличие указаний на анонимные или же недостоверные источники. Главным субъектом фейковых новостей становятся реальные личности или известные организации. Кроме того, при распространении фейков используются информационные жанры, крайне редко – аналитические. Данное соотношение можно объяснить тем, что фейки направлены на максимальный охват аудитории, а большей популярностью аудитории пользуются именно короткие информационные жанры, не требующие много времени на прочтение. Внешняя структура фейковой новости обычно не отличается от структуры настоящей новостной заметки и строится по типу текст и изображение или текст без иллюстративного сопровождения. Практически не используется видеоконтент, что объясняется большей трудоемкостью его фальсификации по сравнению с другими типами контента. Касательно тональности материалов доминируют нейтральные сообщения, не содержащие экспрессивной или эмоционально окрашенной лексики. Выбор тональности также соотносится с внешней структурой фейковых сообщений, мимикрирующих под обычные новости в Сети. Использование нейтральной тональности направлено на то, чтобы ввести в заблуждение интернет-пользователей, не имеющих профессионального журналистского образования и не обладающих навыками для выявления недостоверной информации.

Так компания Facebook решила привлечь всеобщее внимание к решению проблемы фальшивых новостей и организовала обучение пользователей выявлению недостоверной информации. В 2017 году в изданиях The

Guardian, The Daily Telegraph и The Times, опубликован список советов, где говорится что нужно внимательно проверять источник публикаций, скептически относиться к заголовкам, привлекающим внимание своей скандальностью, проверять даты в материалах и искать первоисточник статьи.

Помимо личного анализа пользователем информации существуют программные продукты, предназначенных для автоматизации тех же задач. Приведем несколько примеров таких программ:

Hamilton68

Это платформа, позволяющая анализировать содержимое микроблогов (в основном социальной сети Twitter) на затрагиваемые темы, приводимые ссылки и используемые хэш-теги.

Основная цель данной платформы – удержание текущего положения дел в стране, путём минимизации вмешательства со стороны других стран, в частности через средства массовой информации. Принцип работы заключается в том, что определённый круг специалистов создают список подозрительных Twitter-аккаунтов, которые, по их мнению, являются пропагандистами, распространяющими недостоверную информацию. После этого система анализирует этот список аккаунтов некоторое время и, если аккаунт был замечен в публикации недостоверных новостей пропагандистской направленности – он добавляется в список «сомнительных» с точки зрения контента аккаунтов.

Fake box

Это открытая система для анализа текста на достоверность с использованием встроенных моделей машинного обучения и вручную подобранной базы данных. В качестве входных данных используются текст статьи, ее заголовок и адрес сайта, на котором была опубликована новость. Главным достоинством данной системы является возможность анализа любых интересных новостей.

УДК 004.9

Авачева Т.Г., Дорошина Н.В., Кабанов А.Н.

АДАПТИВНАЯ ТЕХНОЛОГИЯ ПРОГНОЗИРОВАНИЯ МНОГОМЕРНЫХ ВРЕМЕННЫХ РЯДОВ НА СКОЛЬЗЯЩЕМ ВРЕМЕННОМ ИНТЕРВАЛЕ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань
Рязанский государственный медицинский университет им. акад. И.П. Павлова, г. Рязань

Рассматривается задача повышения оперативности и точности прогнозирования многомерного процесса на скользящем временном интервале.

Постановка задачи

Целью работы является поиск решения для повышения оперативности и точности прогнозирования многомерного процесса на скользящем временном интервале. Предлагается технология обработки, которая состоит из этапов [1].

1. Кластеризации исходных многомерных процессов и создание однородных групп процессов на основе модификации алгоритма Прима и методов декомпозиции процессов.

2. Исключение сбойных результатов в процессах однородных групп и организации прогнозирования многомерного процесса на скользящем временном интервале.

Теоретическая часть

Исходный многомерный процесс описывается уравнением:

$$y(1, 0; n) = x(1, 0; n) \cdot b(2, 0) + e(1, 0; n),$$

где $Y(1, 0; n) = \{y_1(n), y_2(n), \dots, y_k(n)\}$ – наблюдаемые временные ряды; $x(0, 1; n) = \{x_1[n], x_2(n), \dots, x_l(n)\}$ – наблюдаемые параметры временных рядов; $b(2, 0) = \{b_{i,j}\}$, $i = \{1, \dots, k\}; j = \{1, \dots, l\}$ – оцениваемые коэффициенты временных рядов; $e(1, 0; n) = \{e_1(n), e_2(n), \dots, e_k(n)\}$ – помеха типа белого шума.

Необходимо оценить коэффициенты $\{b_{i,j}\}$ при наличии аномальных результатов измерений во временных рядах и выделить главные компоненты $x(0, 1)$ для однородных групп временных рядов.

Решение задачи

1. Множество наблюдаемых процессов необходимо разбить на однородные группы по наблюдаемым данным $\{y_1(n), y_2(n), \dots, y_k(n)\}$, $\{x_1[n], x_2(n), \dots, x_l(n)\}$. Для этого можно построить кратчайшее остовное дерево на основе распространения алгоритма Прима на многомерный случай.

2. После определения однородных групп необходимо исключить исключить аномальные измерения в отдельных процессах. Это можно выполнить с помощью метода наименьших модулей.

Для отдельного временного ряда уравнение имеет вид

$$Y[m] = X^T[m]b[n] + e[m]; \quad m = \overline{n - N, n},$$

где $X[m] = \{x_1[n], x_2(n), \dots, x_l(n)\}$ – вектор наблюдаемых линейно независимых факторов; b – вектор неизвестных и подлежащих оценке параметров; $e[m]$ – помеха типа белого шума.

В основу алгоритма положен итеративный метод решения на основе МНМ. При этом оценка \hat{b} , полученная на основе N результатов измерения, имеет вид

$$\hat{b}_N[n] = \hat{A}_N^{-1}[n]Z_N[n],$$

где $\hat{A}_N[n] = \sum_{m=n-N}^n X[m]R[m]X^T[m]$; $Z_N[n] = \sum_{m=n-N}^n X[m]R[m]Y[m]$,

$R[m] = (Y[m] - \hat{Y}[m])^{-1}$; $\hat{Y}[m] = X^T[m]\hat{b}_N[n]$ – оценка m -го измерения выходного сигнала.

Начальные значения $R[m] = 1$; $m = \overline{n - N, n}$ соответствуют определению параметров \hat{b} по методу наименьших квадратов (МНК). Далее вычисления оценок проводятся итерационно до тех пор, пока изменения оценок за одну итерацию не достигнут заданной малой величины. При этом наименьший весовой коэффициент $R[1^*]$ указывает на наиболее грубое 1^* – измерение.

3. После исключения сбойных результатов определяем главные компоненты для однородных групп с помощью нейронных сетей. Входные и выходные данные нейронной сети имеют одинаковую размерность.

Входные данные	Выходные данные
X1	Хапр1
X2	Хапр2
X3	Хапр3
XN	ХапрN

Обработка однородных групп временных рядов на скользящем постоянном интервале

Исходный процесс $y[n]$ представлен в матричном виде.

Модель процесса по результатам наблюдения на интервале $(n_0 - n_0 + N)$ имеет вид

$$yA(1, 0; n) = x(1, 0; n) \cdot \hat{b}(2, 0; n_0 + N);$$

Для квадратичного критерия оптимальные оценки определяются соотношением:

$$\hat{b}(2, 0; n_0 + N) = A^{-1}(1, 1; n_0 + N) \cdot z(2, 0; n_0 + N);$$

$$A(1, 1; n_0 + N) = \sum_{n=n_0}^{n_0+N} x(1, 0; n) \cdot x(0, 1; n);$$

$$z(2, 0; n_0 + N) = \sum_{n=n_0}^{n_0+N} x(1, 0; n) \cdot y(1, 0; n).$$

Для повышения оперативности вычислений на скользящем постоянном интервале используем известные матричные соотношения.

Если $A(1, 1) = B(1, 1) + x(1, 0) * x(0, 1)$, то

$$A^{-1} = B^{-1} - \frac{B^{-1} * x(1, 0) * x(0, 1) * B^{-1}}{x(0, 1) * B^{-1}(1, 1) * x(1, 0) + 1}.$$

Если $A(1, 1) = B(1, 1) - x(1, 0) * x(0, 1)$, то

$$A^{-1} = B^{-1} + \frac{B^{-1} * x(1,0) * x(0,1) * B^{-1}}{1 - x(0,1) * B^{-1}(1,1) * x(1,0)}.$$

При скользянии с постоянным интервалом составляющие процесса в начале интервала удаляются из матрицы A , последующие составляющие добавляются в матрицу A .

При использовании данного алгоритма наблюдается значительный выигрыш по числу вычислительных операций по сравнению с процедурой метода наименьших квадратов.

При многомерно-матричном представлении структура обратной многомерной матрицы получается транспонированием матрицы, а значения ее элементов совпадают со значениями обратной квадратной матрицы

Заключение. Предложенный метод позволяет оперативно прогнозировать многомерный производственный процесс на основе многомерно-матричных алгоритмов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Технологии обработки информации: учебное пособие/ Т.Г. Авачева [и др.]; Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань.

2. Рязанский государственный медицинский университет имени академика И.П. Павлова, г. Рязань. 2017. – 92 с.

УДК 004

Сбродов С.А.

ОБЗОР МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА МЕДИЦИНСКИХ ДАННЫХ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

В данной статье фрагментарно анализируется интеллектуальный анализ медицинских данных. В работе приводятся примеры применения главных методов на практике в медицине. В современном обществе продолжают внедряться методы интеллектуального анализа в медицинскую практику.

Ключевые слова: интеллектуальный анализ, медицинские данные, методы обследования, метод прогнозирования, метод классификации.

В XXI веке, в мире нанотехнологий и усовершенствованных систем, исследователи и ученые проявляют большой интерес к изучению методов интеллектуального анализа медицинских данных.

Прежде чем приступить к анализу методов интеллектуального анализа медицинских данных, стоит разобраться, что же это такое интеллектуальный анализ. Трактование вышеупомянутого анализа обозначает выявление нового знания, который извлекают из определенной базы данных. Немаловажно указать, что интеллектуальный анализ данных состоит из

большого количества направлений и в англоязычной литературе представлен как «Data Mining» (раскопка данных) и «Knowledge discovery» (обнаружение знаний). Изучая данный анализ, необходимо упомянуть о его главных задачах, основанных на поиске не только функциональных, но и логических закономерностей в накопленной информации; создании моделей и правил, которые могут детально объяснить аномалии, а так же прогнозирующее развитие некоторых процессов.

Анализируя специализированную литературу, стоит обратить внимание, в первую очередь, что интеллектуальный анализ способен выявить состояние здоровья человека, а так же увеличить аналитические возможности экспертов в медицине. Сегодня крупные медицинские организации активно применяют системы поддержки, основанные на методах интеллектуальной обработки данных, что в значительной мере упрощают задачу медицинским специалистам в постановки диагнозов, назначения курса лечения, прогнозирования развития заболеваний. В современной медицине с помощью интеллектуального анализа можно с легкостью сформировать выборку исходных данных, описать не только состояния пациентов, но и сгруппировать или упорядочить их на основе свойств и выраженных признаков больных [7].

Рассматривая методы интеллектуального анализа медицинских данных, стоит упомянуть о первоочередной проблеме, которая состоит в создании методов и инструментов интеллектуального анализа клинических текстов на русском языке. Методы структурируют в группы в зависимости от решаемых проблем, поэтому следующим этапом в данной работе будет анализ распространенных методов в современной медицине.

Одним из главных методов в медицине считают прогнозирование течения болезни, воздействия препарата или группы препаратов и уровня смертности. Ко второму методу относят обследование, целью которого является правильный поставленный диагноз, основанный на выявлении всех симптомов. Классификация представляет собой третий метод, один из самых распространенных методов, который формируется на уточнении диагноза [6, с.690-695].

Рассматривая проблематику исследуемой темы, стоит выделить главные применяемые методы интеллектуального анализа данных для более точной обработки медицинской информации. Зарубежные исследователи Харпер П. и Хонг Ху в своих научных трудах пишут о методах классификации на разных наборах медицинских данных из репозитория Kent Ridge Biomedical Dataset Repository [2, 3]. Анализ точности исследуемой классификации показал, что она зависит от набора анализируемых данных, применяемого метода извлечения и ранжирования ключевых признаков. Исследователь Маринов М. в своей научной работе обращает внимание на то, что деревья принятия решений представляют собой один из самых

распространенных алгоритмов классификации в системах интеллектуальной обработки медицинских данных [4, с. 2431-2448].

Немаловажно выделить исследователей Дэвида В. Бэйтс и Суши Сария, которые анализируют метод прогнозирования и априорное определение стоимости лечения пациента в своем труде. Исследователи доказали, что данный метод построен на основе деревьев принятия решений, представленная информация собиралась на протяжении трех лет и дала практически точный результат (85%). Прогнозирование, как доказанный факт, применяется для нахождения скрытых зависимостей между признаками [1, с. 1123-1131.]

Исследуя метод кластеризации, который довольно часто применяется для анализа массивов генетической информации, стоит упомянуть научный труд зарубежного исследователя Лауры Дж. [5, с. 530-536.] Исследователь подчеркивает, что данный метод применяется для повышения точности прогнозирования развития опухолей. Данный метод использует кластеризацию массива, в ее работе было проанализировано ДНК 86 видов опухолей груди в двух кластерах, было выявлено, что данный метод помогает врачам не только для выявления точного диагноза, но и назначения курса лечения, а так же прогнозирования развития заболеваний.

Итак, подводя итоги фрагментарного обзора методов интеллектуального анализа медицинских данных, было выявлено, что на сегодняшний день современная медицина не может существовать без применения точных и надёжных методов прогнозирования, обследования и классификации. В современном обществе продолжают внедряться методы интеллектуального анализа, которые облегчают задачу медицинских специалистов в ведении электронных медицинских карт, создании межрегиональных медицинских баз данных, что позволяет быстро разрешать медицинские задачи, базируясь на анализах о пациентах. Методы идут в ногу со временем, специалисты разрабатывают новые системы, проверяют их на качество и применение на практике, устраняют ошибки, усовершенствуют и отработывают неточности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. David W. Bates, Suchi Saria, Lucila Ohno-Machado et al. Big data in health care: using analytics to identify and manage high-risk and high-cost patients // Health Affairs. 2014. V. 33. № 7. Pp. 1123-1131.

2. Harper P.R. A review and comparison of classification algorithms for medical decision making // Health Policy. 2005. V. 71. № 3. Pp. 315-331.

3. Hong Hu, Jiuyong Li, Ashley Plank et al. A comparative study of classification methods for microarray data analysis // Proceedings of the fifth Australasian conference on data mining and analytics / Australian Computer Society, Inc. 2006. V. 61. Pp. 33-37

4. Illhoi Yoo, Patricia Alafaireet, Miroslav Marinov et al. Data mining in healthcare and biomedicine: a survey of the literature // Journal of medical systems. 2012. V. 36. № 4. Pp. 2431-2448.

5. Laura J van't Veer, Hongyue Dai, Marc J Van De Vijver et al. Gene expression profiling predicts clinical outcome of breast cancer // Nature. 2002. V. 415. № 6871. Pp. 530-536.

6. Obenshain M. K. Application of data mining techniques to healthcare data // Infection Control and Hospital Epidemiology. 2004. V. 25. № 8. Pp. 690-695.

7. R.S. Santos, S.M.F. Malheiros, S. Cavalheiro, J.M. De Oliveira. A data mining system for providing analytical information on brain tumors to public health decision makers // Computer methods and programs in biomedicine. 2013. V. 109. № 3. Pp. 269-282.

УДК 004

Малистова М.Р.

МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕКСТОВЫХ ДАННЫХ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина г. Рязань

В работе рассматриваются разные различные методы интеллектуального анализа текстовых данных. Целью работы является сбор информации по различным методам, выявление их достоинств и недостатков и сравнение их друг с другом.

Ключевые слова: анализ текста, текстовые данные, интеллектуальный анализ.

Введение

Данные становятся важнейшим организационным ресурсом, обеспечивающим конкурентные преимущества, дают начало инициативам по менеджменту знаний. Ручная обработка и классификация данных является неэффективной и дорогостоящей. Ее стремятся либо полностью автоматизировать, либо использовать только на тех этапах работы, когда среди большого количества данных автоматически отобраны только нужные данные.

Интеллектуальный анализ текста, или text mining – это автоматизация извлечения сведений из текстовых данных. Его особенность (в отличие от анализа других данных) заключается в неформализованности исходной информации, т. е. ее не описать простой математической функцией.

Анализ текста позволяет не только извлекать полезные сведения из проектов по управлению неструктурированными данными, но и ожидать от них большего показателя окупаемости инвестиций. Для бизнеса это означает возможность получать выгоду от использования крупных массивов данных, избегая затратной ручной обработки: отставить в сторону нерелевантный материал и просто получать ответы.

Анализ текста помогает сориентироваться в огромном потоке научных публикаций: настроить получение релевантных статей, сэкономить деньги и время.

В настоящее время существует много различных методов анализа текстовых данных, каждый метод имеет свои достоинства и недостатки, а также разные методы применимы в различных сферах и для разных целей.

Целью работы является сбор материала о различных методах интеллектуального анализа текстовых данных, выявление их особенностей и преимуществ.

Обзор методов анализа текстовых данных

Для интеллектуального анализа данных существует много методов, которые, в общей сложности, представляют собой поиск определенной информации в соответствии с требованиями пользователя. Существует много различных методов поиска информации, но самыми распространенными и наиболее часто используемыми являются следующие: 1) метод, основанный на терминах, 2) метод, основанный на фразах, 3) концептуальный метод, 4) метод шаблонной таксономии. Рассмотрим подробнее каждый из этих методов:

1. Метод, основанный на терминах

Термин в документе – слово, имеющее смысловое значение. В данном методе документ анализируется на основе терминов и имеет преимущества эффективной вычислительной производительности. Эти методы появились за последние пару десятилетий в сообществах поиска информации и машинного обучения. В методах, основанных на терминах, присутствуют проблемы многозначности и синонимии [1]. Многозначность означает, что слово имеет несколько значений, а синонимия – это несколько слов, имеющих одно и то же значение. Смысловое значение многих обнаруженных терминов не может быть точно известно, и, следовательно, нельзя быть уверенным, что найденные слова соответствуют запросам пользователей ответить на вопрос, что хотят пользователи.

2. Метод, основанный на фразах

Фраза несет больше семантики, такой как информация, и меньше неоднозначна. Данным методом документ анализируется на фразовой основе, поскольку фразы менее двусмысленны и более разнообразны, чем отдельные термины [2]. Существует ряд причин, которые усложняют поиск фраз:

- 1) фразы имеют худшие статистические свойства, чем термины;
- 2) они имеют низкую частоту встречаемости;
- 3) присутствует большое количество повторяющихся и зашумленных фраз среди них.

3. Концептуальный метод

В данном методе термины анализируются по предложениям и уровням документа. Методы интеллектуального анализа текста в основном основаны на статистическом анализе слова или фразы. Статистический

анализ частоты использования термина показывает важность слова без документа. Два термина могут иметь одинаковую частоту в одном и том же документе, но один термин вносит более подходящий вклад, чем значение, которое несет другой термин [3]. Терминам, которые фиксируют смысл текста, следует придать большее значение, поэтому вводится новый концептуальный анализ. Эта модель состояла из трех компонентов. Первый компонент анализирует смысловую структуру предложений. Второй компонент создает концептуальный онтологический граф (COG) для описания смысловых структур, последний компонент извлекает основные концепции на основе первых двух компонентов для построения векторов с использованием стандартной модели векторного пространства. Концептуальная модель может эффективно различать неважные термины и значимые термины, которые описывают значение предложения [4]. Данная модель обычно опирается на методы обработки естественного языка. Выбор функций применяется к концепциям запроса для оптимизации представления и устранения шума и неоднозначности.

4. Метод шаблонной таксономии

В методе шаблонной таксономии документы анализируются на основе шаблонов. Анализ шаблонов широко изучается в сообществах интеллектуального анализа данных в течение многих лет. Шаблоны могут быть обнаружены методами интеллектуального анализа данных, такими как анализ ассоциативных правил, анализ частых наборов элементов, последовательный анализ шаблонов и анализ закрытых шаблонов [5]. Использование обнаруженных знаний (шаблонов) в области интеллектуального анализа текста сложно и неэффективно, потому что некоторые полезные длинные шаблоны с высокой специфичностью не имеют поддержки (например, проблема низкой частоты). Не все частые короткие шаблоны полезны, поэтому они известны как неправильная интерпретация шаблонов, и это приводит к неэффективной работе.

В методике на основе шаблонов используются два процесса: развертывание шаблона и развитие шаблона [6]. Этот метод уточняет обнаруженные закономерности в текстовых документах. Экспериментальные результаты показывают, что модель, основанная на шаблонах, работает лучше, чем не только другие методы, основанные на чистом интеллектуальном анализе данных или модель, основанная на концепциях, но и модели на основе терминов.

Заключение

В работе были рассмотрены основные методы интеллектуального анализа текстовых данных, такие как метод, основанный на терминах, метод, основанный на фразах, концептуальный метод, метод шаблонной таксономии. Также были рассмотрены их основные преимущества и недостатки.

В результате было замечено, что метод, основанный на терминах, страдает многозначностью и синонимией, тогда как подход на основе фраз работает лучше, так как фраза несет больше смысла и меньше неоднозначный. Два термина могут иметь одинаковую частоту в статистическом анализе, эта проблема может быть решена концептуальным подходом при помощи поиска терминов, дающих больше смысла. В подходе, основанном на шаблонах, таксономия шаблонов формируется для решения проблемы низкой частоты и проблемы неправильного толкования.

Подводя итог, можно отметить, что цель работы была достигнута за счет проведения обзора и сравнения основных методов интеллектуально-го анализа данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. G. Salton and C. Buckley, «Term-Weighting Approaches in Automatic Text Retrieval», *Information Processing and Management: An Int'l J.*, vol. 24, no. 5, pp. 513-523, 1988.

2. H. Ahonen, O. Heinonen, M. Klemettinen, and A.I. Verkamo, «Applying Data Mining Techniques for Descriptive Phrase Extraction in Digital Document Collections», *Proc. IEEE Int'l Forum on Research and Technology Advan.*

3. S. Shehata, F. Karray, and M. Kamel, «Enhancing Text Clustering Using Concept-Based Mining Model», *Proc. IEEE Sixth Int'l Conf. Data Mining (ICDM '06)*, pp. 1043-1048, 2006.

4. S. Shehata, F. Karray, and M. Kamel, «A Concept-Based Model for Enhancing Text Categorization», *Proc. 13th Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 629-637, 2007.

5. S.-T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen. «Automatic Pattern-Taxonomy Extraction for Web Mining», *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI '04)*, pp. 242-248, 2004.

6. S.-T. Wu, Y. Li, and Y. Xu. «Deploying Approaches for Pattern Refinement in Text Mining», *Proc. IEEE Sixth Int'l Conf. Data Mining (ICDM '06)*, pp. 1157-1161, 2006.

СОДЕРЖАНИЕ

Белов В.В., Чистякова В.И. СИСТЕМАТИЧЕСКИЙ ПОДХОД К ПРОЕКТИРОВАНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ.....	4
Дмитриева Т.А., Рунцо А.А. ОПИСАНИЕ ТРЕБОВАНИЙ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОЦЕССА ПОЛУЧЕНИЯ ГОТОВОЙ ПРОДУКЦИИ ПОТРЕБИТЕЛЯМИ	9
Шурыгина О.В., Цуканова Н.И. РАЗРАБОТКА НОУТБУКА В СРЕДЕ GOOGLE COLAB ПО ТЕМЕ «КЛАССИФИКАЦИЯ ТЕКСТОВЫХ СООБЩЕНИЙ С ПОМОЩЬЮ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ».....	13
Крошилина А.А., Благодаров Е.А., Благодарова Т.А. ПРИМЕНЕНИЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ	16
Белов В.В., Моудио Моудио Фернанд Елиссе СООТНОШЕНИЕ МЕЖДУ ФУНКЦИОНАЛЬНЫМ И ПРОЦЕССНЫМ ПОДХОДАМИ К УПРАВЛЕНИЮ БИЗНЕС-ПРОЦЕССАМИ	18
Карпушкина Д.А. ИНФОРМАЦИОННЫЕ СИСТЕМЫ	23
Новинская Д.В., Проказникова Е.Н. ОСОБЕННОСТИ АЛГОРИТМОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ	25
Дмитриева Т.А., Рябичев А.А. НЕОБХОДИМОСТЬ РАЗРАБОТКИ ПРИКЛАДНОГО ПРОГРАММНОГО \ ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ ЭЛЕКТРОННЫХ ТЕХНИЧЕСКИХ РУКОВОДСТВ ПО ОБСЛУЖИВАНИЮ И РЕМОНТУ ИЗДЕЛИЙ	26
Тюрина Е.М., Проказникова Е.Н. ОЦЕНКА ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ ПОЖАРНОЙ ЧАСТИ.....	30
Бухтина А.О., Горбатова Ю.А. ВЫБОР АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОСНОВНЫХ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ	32
Белов В.В., Филоненко И.Н. ПОКАЗАТЕЛЬ СПЕЦИАЛЬНОЙ ЛЕКСИЧЕСКОЙ СХОЖЕСТИ.....	33
Федосова Е.Б. РАЗРАБОТКА МОРФОЛОГИЧЕСКОГО АНАЛИЗАТОРА ВОПРОСНО-ОТВЕТНОЙ СИСТЕМЫ ДЛЯ МЕТРОЛОГИЧЕСКОЙ ЛАБОРАТОРИИ.....	37
Авачева Т.Г., Дорошина Н.В., Кабанов А.Н. ПОВЫШЕНИЕ ОПЕРАТИВНОСТИ И ТОЧНОСТИ АЛГОРИТМОВ ОПТИМИЗАЦИИ МНОГОМЕРНОЙ, МНОГОКРИТЕРИАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ФОРМИРОВАНИЯ ЭКСПЕРТНЫХ ГРУПП МЕТОДОМ ВЗВЕЩЕННОГО ПАРНОГО СРАВНЕНИЯ	39
Мельникова И.С. ИНФОРМАЦИОННЫЕ СИСТЕМЫ ПОВЫШЕНИЯ КАЧЕСТВА РАБОТЫ И КОНКУРЕНТОСПОСОБНОСТИ ПРЕДПРИЯТИЙ РОЗНИЧНОЙ ТОРГОВЛИ	42

Кабанова Е.О., Крошилилин А.В. АВТОМАТИЗИРОВАННОЕ РАНЖИРОВАНИЕ ЗАДАЧ ПО ПРИОРИТЕТАМ.....	46
Пахомова Н. Ф., Крошилилина С.В. ОБЗОР СУЩЕСТВУЮЩИХ ИНФОРМАЦИОННЫХ СИСТЕМ И ВЫБОР ЭФФЕКТИВНОГО СПОСОБА АВТОМАТИЗАЦИИ АТЕЛЬЕ.....	48
Миронова А.В. МЕТОДОЛОГИЯ БЭМ	52
Гранкин А.Н., Крошилилин А.В. КОНЕЧНЫЕ АВТОМАТЫ В ПРОЕКТИРОВАНИИ ВЕБ-ИНТЕРФЕЙСОВ ИНФОРМАЦИОННЫХ СИСТЕМ.....	54
Никичкин Б.В., Беликова В.О., Штринев А.Ю. РЕГУЛЯРНЫЙ ЯЗЫК И ПРОГРАММА-РАСПОЗНАВАТЕЛЬ.....	56
Никичкин Б.В., Горкин А.В., Ловкин М.В. ПРОГРАММА-РАСПОЗНАВАТЕЛЬ ДЛЯ ЯЗЫКА ВЕЩЕСТВЕННЫХ ДЕСЯТИЧНЫХ ЧИСЕЛ С ФИКСИРОВАННОЙ ТОЧКОЙ	61
Головкин Н.В., Цуканова Н.И. РАЗРАБОТКА НОУТБУКА В СРЕДЕ GOOGLE COLAB ПО ТЕМЕ «КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ».....	66
Старикова К.Ю., Крошилилина С.В., Крошилилина А.А. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ НА ПРЕДПРИЯТИЯХ МАЛОГО БИЗНЕСА.....	68
Захаркин А.В. ТЕНДЕНЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В 2020 ГОДУ	70
Попков И.А. АЛГОРИТМЫ И ОПТИМИЗАЦИЯ ОБМЕНА ПЛАТЕЖНЫХ ДАННЫХ.....	72
Самохина Л.А. КРЕДИТНЫЕ СИСТЕМЫ И ОНЛАЙН ПРОДУКТЫ БАНКОВ.....	74
Лукьянова Т.А. АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ В DATA MINING.....	76
Алексашкин В.В., Баконин Р.В., Кирилова Л.М. ФЕЙКОВЫЕ НОВОСТИ В СОВРЕМЕННОМ ОБЩЕСТВЕ	79
Авачева Т.Г., Дорошина Н.В., Кабанов А.Н. АДАПТИВНАЯ ТЕХНОЛОГИЯ ПРОГНОЗИРОВАНИЯ МНОГОМЕРНЫХ ВРЕМЕННЫХ РЯДОВ НА СКОЛЬЗЯЩЕМ ВРЕМЕННОМ ИНТЕРВАЛЕ	81
Сбродов С.А. ОБЗОР МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА МЕДИЦИНСКИХ ДАННЫХ.....	84
Малистова М.Р. МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕКСТОВЫХ ДАННЫХ	87

межвузовский сборник научных трудов

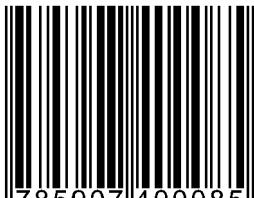
МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Подписано в печать 09.12.2020. Формат бумаги 60x84/16.
Бумага офсетная. Печать цифровая. Усл. печ. л. 5.875.
Тираж 100 экз. Заказ 3693
ФГБОУ ВО «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина»
390005, Рязань, ул. Гагарина, 59/1.

Издательство ИП Коняхин А.В. (Book Jet).

Отпечатано в типографии «Book Jet»
390005, г. Рязань, ул. Пушкина, д. 18
Сайт: <http://bookjet.ru>
Почта: info@bookjet.ru
Тел.: +7(4912)466-151

ISBN 978-5-907400-08-5



9 785907 400085 >