

Школа Инженерная школа ядерных технологий

Направление подготовки 01.03.02 «Прикладная математика и информатика»

Отделение школы (НОЦ) Отделение экспериментальной физики

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка образовательного портала с применением методов искусственного интеллекта

УДК 004.738.1:004.896

Студент

Группа	ФИО	Подпись	Дата
0В51	Курбонов Шамил Алиевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ ИЯТШ	Семенов Михаил Евгеньевич	к.ф.-м.н., доцент		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и
ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Конотопский Владимир Юрьевич	к.э.н. доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Исаева Елизавета Сергеевна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ ИЯТШ	Крицкий Олег Леонидович	к.ф.-м.н., доцент		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
0B51	Курбонову Шамилу Алиевичу

Школа	ИЯТШ	Отделение (НОЦ)	ОЭФ
Уровень образования	Бакалавр	Направление/специальность	01.03.02 Прикладная математика и информатика

Тема ВКР:

Разработка образовательного портала с применением методов искусственного интеллекта	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Рабочая зона где установленные компьютерные сервера, и данные сервера поддерживают образовательный портал studybro.ml для глобального доступа.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	– ГОСТ Р 22.0.02-2016; – СНиП 23-05-95; – ГОСТ 12.1.038-82 – Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 01.04.2019)
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	– Недостаточная освещенность рабочей зоны; – Повышенный уровень статического электричества; Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.
3. Экологическая безопасность:	– анализ воздействия объекта на атмосферу (выбросы); – анализ воздействия объекта на гидросферу (сбросы); – анализ воздействия объекта на литосферу (отходы);
4. Безопасность в чрезвычайных ситуациях:	– перечень возможных ЧС на объекте; – выбор наиболее типичной ЧС; – разработка превентивных мер по предупреждению ЧС; – разработка мер по повышению устойчивости объекта к данной ЧС; – разработка действий в результате возникшей ЧС и мер по ликвидации её последствий

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Исаева Е.С.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0B51	Курбонов Шамил Алиевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ,
РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
0B51	Курбонову Шамилу Алиевичу

Школа		ИЯТШ	Отделение школы (НОЦ)	ОЭФ
Уровень образования		Бакалавр	Направление/специальность	01.03.02 Прикладная математика и информатика

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	
2. Планирование и формирование бюджета научных исследований	
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	

Перечень графического материала (с точным указанием обязательных чертежей):

1. Линейный график работ

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский Владимир Юрьевич	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0B51	Курбонов Шамил Алиевич		

Реферат

Выпускная работа включает в себя 4 главы, 94 страниц, 14 таблиц, 15 рисунков и 24 источника литературы, соответствует утвержденной теме.

В работе рассматриваются практические вопросы понимания естественного языка, построения методов искусственного интеллекта, а также обучение модели.

В ходе работы был проведен анализ предметной области, в качестве инструментов для реализации образовательного портала были выбраны языки программирования PHP и SQL.

Проведена нормализация запроектированной базы данных, затем она была реализована в СУБД MySQL. В базе данных выделено 5 сущностей, из которых две стержневых, одна обозначающая и две ассоциативных.

Для интеллектуальной генерации тестовых заданий к материалам образовательного портала мы использовали библиотеку DeepPavlov, которая показала удовлетворительные результаты при поиске ответов на однозначные и простые вопросы.

Проведено тестирование и апробация портала. Разработанный портал доступен по ссылке: <http://studybro.ml/>

Оглавление

Введение	7
1 Теоретическая часть	9
1.1 Превращение искусственного интеллекта в товар	9
1.2 Машинное обучение	10
1.3 Типы машинного обучения	14
1.4 Язык	15
1.4.1 Понимание естественного языка	16
1.4.2 Обработка естественного языка	17
1.5 Компьютерное зрение	19
1.6 Библиотека DeepPavlov	20
1.6.1 Основные понятия	20
1.6.2 Open-domain question answering (ODQA)	20
1.6.3 Описание модели	22
2 Практическая часть	24
2.1 Построение инфологической модели данных	24
2.1.1 Проектирование целостности базы данных	26
2.1.2 Нормализация таблиц	29
2.2 Создание таблиц на языке SQL	30
2.3 Реализация проектируемой базы данных в СУБД MySQL	34
2.4 Разработка интерфейса пользователя и администратора	36
2.6 Требования к модели	40
2.7 Взаимодействие с моделью	40
2.8 Обучение модели	41
2.9 Результаты модели	42
Заключение	44
3. Социальная ответственность	45
3.1 Правовые и организационные вопросы обеспечения безопасности	46
3.2 Производственная безопасность	49
3.3 Анализ опасных и вредных производственных факторов и обоснование мероприятий по их устранению	51

3.3.1 Недостаточная освещенность рабочего места	51
3.3.2 Повышенный уровень статического электричества	52
3.3.3 Повышенное значение напряжения в электрической цепи	52
3.4 Экологическая безопасность	53
3.5 Безопасность в чрезвычайных ситуациях	54
Вывод	56
4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	57
4.1 Организация и планирование работ	57
4.1.1 Продолжительность этапов работ	59
4.2 Расчет сметы затрат на выполнение проекта	64
4.2.1 Расчет затрат на материалы	64
4.2.2 Расчет заработной платы	65
4.2.3 Расчет затрат на социальный налог	66
4.2.4 Расчет затрат на электроэнергию	66
4.3 Расчет амортизационных расходов	68
4.3.1 Расчет прочих расходов	69
4.3.2 Расчет общей себестоимости разработки	69
4.3.3 Расчет прибыли	70
4.3.4 Расчет НДС	70
4.3.5 Цена разработки НИР	71
4.3 Оценка экономической эффективности проекта	71
Список литературы	72
Приложения	75
Приложение 1. Построение даталогической модели данных	75
Приложение 2	76
Приложение 3	77
Приложение 3	78
Приложение 4	80
Приложение 5 скрипт разработки модели	82

Введение

Мы живем в потоке данных. Согласно недавним оценкам, ежедневно генерируется 2,5 квинтиллиона (10^{18}) байт данных. Это такой огромный объем данных, что более 90% информации, которую мы храним в наши дни, было сгенерировано за все прошедшее десятилетие. К сожалению, люди не способны воспользоваться подавляющей частью этой информации. Данные либо лежат за пределами возможностей стандартных аналитических методов, либо они просто слишком обширны, чтобы человеческий мозг смог их проанализировать.

Благодаря методам машинного обучения мы наделяем компьютеры способностями обрабатывать большие объемы данных, которые в противном случае стояли бы непроницаемой стеной, даем им возможность обучаться на этих данных и извлекать из них практические выводы. От массивных суперкомпьютеров, которые обеспечивают работу поисковиков Google, Yandex и т.д. до смартфонов, которые мы носим в наших карманах, – везде мы опираемся на машинное обучение, которое приводит в действие значительную часть окружающего нас мира, и нередко мы об этом даже не догадываемся.

Цель данной работа разработка образовательного портала с применением методов искусственного интеллекта. Для достижения поставленной цели необходимо:

1. Провести анализ предметной области и выбрать инструменты для реализации образовательного портала.
2. Разработать инфологическую и даталогическую модели данных.
3. Реализовать запроектированную базу данных.
4. Разработать интерфейса пользователя и администратора портала.
5. Провести тестирование и сделать выводы по работе.

Разработанный портал доступен по ссылке: <http://studybro.ml/> .

1 Теоретическая часть

1.1 Превращение искусственного интеллекта в товар

За последние годы произошел взрывной, почти экспоненциальный рост объема информации. В условиях ежедневно дешевого пространства для хранения данных крупные корпорации, равно как и небольшие стартапы, стали сохранять свои архивы, чтобы в какой-то момент их проанализировать и получить важную информацию, способную принести выгоду своему предприятию. Эта тенденция в значительной части была поддержана облачной революцией. Сама облачная революция обусловлена появлением компьютеров с более высоким быстродействием и дешевеющим пространством хранения данных. Услуги облачного вычисления и хранения данных в облаке, предоставляемые такими компаниями, как Amazon AWS и Microsoft Azure, настолько дешевые, что уже больше не имеет смысла уничтожать даже журнальные (лог-файлы) данные десятилетней давности, произведенные серверами и программным обеспечением предприятия. В результате каждый день и каждый час компании производят ошеломляющие объемы данных. Этот большой объем данных называется *большими данными* (big data). Большие данные применяются почти во всех секторах экономики, таких как банковское дело, розничная торговля, ИТ, социальные сети, здравоохранение, наука, спорт, управление и т. д.

Чтобы вообразить масштаб больших данных, обратите внимание на приведенную ниже статистику.

По состоянию на декабрь 2018 г. поисковик Google обрабатывал примерно 100 миллиардов поисковых запросов в месяц. Это больше 1,2 триллиона запросов в год! Google анализирует свои поисковые данные с

целью определения поисковых тенденций среди различных географических регионов и демографических групп.

Социальная сеть Facebook в день обрабатывает свыше 300 миллионов фотографий из своей базы пользователей. Facebook анализирует свои данные, посты и фотографии с целью предоставления своим пользователям более точных объявлений.

Крупнейшая в мире сеть оптовой и розничной торговли Wal-Mart каждый час обрабатывает свыше 1 миллиона покупательских транзакций. Wal-Mart анализирует эти данные, чтобы понимать, какие товары реализуются лучше других, какие товары реализуются вместе, и получать другую розничную аналитическую информацию.

Традиционные методы обработки данных стали нежизнеспособными из-за сложности данных и времени, которое требуется для их полного анализа. Чтобы проанализировать этот огромный объем данных, был необходим радикально новый подход. Как оказалось, с большими данными могут использоваться и методы машинного обучения, применяемые для тренировки сложных систем, – *искусственный интеллект* (ИИ). В результате сегодня ИИ больше не является прерогативой крупных частных и государственных научно-исследовательских институтов. ИИ и его разнообразная методология используются для построения и поддержки программных продуктов для всех видов компаний.

1.2 Машинное обучение

Технология машинного обучения (МО) находится в самой основе искусственного интеллекта. В рамках МО проводятся исследования и конструирование алгоритмов, которые могут обучаться на ранее обработанных данных и прогнозировать, основываясь на результатах

этого обучения. Именно МО приводит в действие интеллектуальную машину, и именно на МО функционирует искусственный интеллект.

Традиционный не-МО алгоритм машинного перевода имеет статические программные инструкции, которые идентифицируют, на каком языке написано предложение: используемые слова, грамматическая структура и т. д. Не-МО алгоритм идентификации лиц аналогичным образом имеет жестко запрограммированное определение лица: нечто округлое с цветом кожи и двумя небольшими темными областями в верхней части (глазами) и т. д. С другой стороны, у МО алгоритма нет такого жесткого программирования; он учится на примерах. Если в процессе тренировки использовать большое количество предложений на французском языке и добавить небольшое количество предложений на другом языке, то МО алгоритм научится определять французские предложения [11].

Целый ряд реальных практических задач, таких как перевод с языка на язык, предсказание погоды, фильтрация спама в почтовых сообщениях, предсказание результатов выборов, задачи классификации (такие как различение видов птиц по фотографиям) и т. д., имеет нелинейный характер. МО является идеальным решением для таких нелинейных задач, где проектирование и программирование задаваемых явным образом алгоритмов с использованием статических программных инструкций попросту не работает.

Технология МО является многосложной и многогранной. МО является подобластью информатики, включающей в себя несколько тем, в частности связанных с математикой и статистикой. Хотя для охвата всех аспектов МО потребуется не одна дипломная работа, давайте рассмотрим на общепринятые термины, которые с ним связаны (рис. 1 и 2).

Алгоритм машинного обучения, в частности нейронная сеть, которая «обучается» заданной теме на тренировочных данных. Результатом такого обучения является *натренированная модель* (рис. 1).

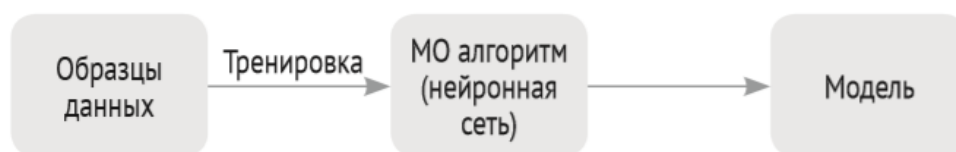


Рисунок 1 – Алгоритм машинного обучения

Натренированная модель затем может принимать новые или ранее известные данные и делать информированные предсказания (рис. 2).

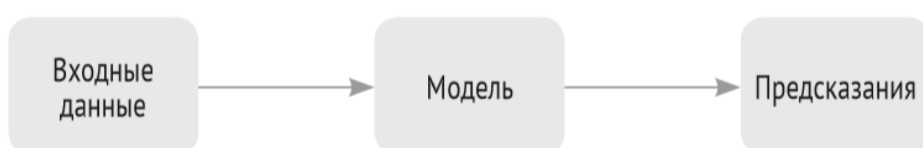


Рисунок 2 – Натренированная модель

Прежде чем система машинного обучения сможет начать разумно отвечать на вопросы по теме, она должна этой теме обучиться. Для этого МО в значительной степени опирается на начальный набор данных по теме обучения. Эти исходные данные называются тренировочными данными. Чем больше тренировочных данных, тем больше шаблонов (образов) наша машина в состоянии идентифицировать и тем точнее она сможет отвечать на вопросы – новые и ранее известные – по этой теме. Для получения надежных результатов обычно недостаточно нескольких сотен или даже тысяч записей тренировочных данных. Действительно точные, человекоподобные машины были натренированы с использованием миллионов записей или нескольких гигабайтов данных в течение дней, месяцев или даже лет. Продолжительность непрерывной работы персонального компьютера с хорошей вычислительной мощностью и высококачественной видеокартой для тренировки алгоритма машинного перевода данными объемом свыше 1 Гб только для

одной пары языков составит больше месяца [9]. Качество тренировочных данных и то, как модель сконструирована, имеют равноценное значение. Используемые данные должны быть точными, очищенными и поступать через надежные источники.

Модель должна быть разработана с использованием сценариев реальной практической деятельности. Так что в следующий раз, когда ваше приложение распознавания образов неправильно идентифицирует захваченный объект или ваше любимое приложение машинного перевода выдаст смешной перевод, вините в этом качество тренировочных данных или модель, которая использовалась. Кроме того, важно отметить, что обучение не является процедурой начального этапа: это непрерывный процесс. Первоначально машина обучается на тренировочных данных; позже она делает это на пользователях.

Исследование в области ИИ привело к развитию нескольких подходов к реализации машинного обучения. *Искусственная нейронная сеть* (ИНС) является одним из самых популярных подходов. ИНС, или просто нейросеть, является обучаемым алгоритмом. Основой для разработки нейросети послужили структура и функциональные аспекты биологических нейронных сетей. Вычисления структурированы с точки зрения взаимосвязанных групп искусственных нейронов, обрабатывающих информацию, используя ассоциативный подход к компьютерным вычислениям. Они используются для моделирования сложных связей между входами и выходами с целью отыскания этих связей в данных шаблонов (образов). Другие популярные подходы представлены глубоким обучением, системами на основе продукционных правил (если – то), разветвленными алгоритмами решений и байесовскими сетями [9].

Таким образом, когда в нейронные сети подается достаточный объем тренировочных данных, мы получаем то, что называется натренированной моделью. Модели – это математические и

статистические функции, которые могут создавать прогнозы (информированные предположения) в отношении предоставленных входных данных. Например, основываясь на метеорологической информации за прошедшие 10 лет (тренировочных данных), модель машинного обучения может научиться предсказывать погоду на несколько дней вперед.

1.3 Типы машинного обучения

Обучение с учителем, или контролируемое обучение, выполняется, когда тренировочные данные помечены. Если в процессе обучения алгоритма распознавания языка предложения, передаваемые в этот алгоритм, будут помечены, то такое обучение называется обучением с учителем. То есть в этом случае требуется передавать алгоритму предложения, явным образом помеченные языком, на котором они написаны. Например, предложение на французском и не на французском языке, на испанском и не на испанском языке и т. д. Поскольку предварительная разметка выполняется людьми, эта работа увеличивает трудозатраты и стоимость поддержания таких алгоритмов.

Обучение без учителя, или неконтролируемое обучение, выполняется, когда тренировочные данные не помечены. Из-за отсутствия меток, разумеется, алгоритм не может волшебным образом научиться правильно определять язык предложения, но он может дифференцировать один язык от другого. Таким образом, используя обучение без учителя, алгоритм МО может научиться определять, что французские предложения отличаются от испанских, которые отличаются от персидских и т. д.

Обучение с подкреплением, или стимулированное обучение, выполняется, когда машине не передаются тренировочные данные явным образом. Чтобы достигнуть поставленной цели, алгоритм должен

взаимодействовать с окружающей средой. Из-за отсутствия тренировочных данных он должен обучаться самостоятельно с нуля и полагаться на метод проб и ошибок, чтобы принимать решения и обнаруживать свои собственные правильные пути. У каждого выполняемого машиной действия есть последствие, и каждому последствию задано числовое вознаграждение. Таким образом, если действие приводит к желательному результату, оно получает «хорошие» отметки. И если результат имеет катастрофические последствия, то он получает отметки «очень и очень плохо». Как и люди, машина стремится максимизировать свое итоговое числовое вознаграждение – то есть получить как можно больше «хороших» и «очень хороших» отметок, не повторяя свои ошибки. Этот метод машинного обучения особенно полезен, если машина будет работать в быстро изменяющихся средах, где создание и поставка тренировочных данных попросту невозможны.

1.4 Язык

Приведем краткую классификацию языков и их типов, т.к. данная работа тесно связана с обработкой текста.

Люди взаимодействуют друг с другом одним из трех способов: словесно, письменно и жестами. Эти три способа объединены в одно понятие – «язык». *Язык* – это свод правил общения, который одинаков для каждого человека. Общение может быть устным и письменным, и между этими двумя способами общения есть различия. Формальной формой общения является *письмо*. При этом язык знаков, язык жестов полностью отличается. Большая часть усилий, приложенных в исследовании ИИ, состояла в том, чтобы наделить машины способностью понимать людей так же естественно, как это делают сами люди. Поскольку машинам легче понимать письменную речь, чем

устную, мы начнем наше обсуждение с основ понимания языка в письменной форме.

1.4.1 Понимание естественного языка

Понимание *естественного языка* (ЕЯ) – это способность машины понимать людей посредством естественных языков. Компьютер изначально разработан, чтобы понимать биты и байты, программный код и логику, программы и инструкции, а не естественные языки. Иными словами, компьютер умело управляет структурированными данными, чего нельзя сказать о неструктурированных данных.

Естественный язык регламентируется некими *правилами* (грамматикой), однако во время ежедневного и неофициального общения эти правила не всегда соблюдаются. В результате люди легко могут понимать письменные или словесные высказывания с плохой грамматикой, неправильным произношением, просторечиями, сокращениями и т. д. Можно смело сказать, что естественные языки регламентируются гибкими правилами.

Технология понимания ЕЯ превращает неструктурированные исходные данные (рис. 1.4), регламентированные гибкими и плохо определенными правилами, в структурированные данные, которые машина способна понять. Отметим, что именно эта технология делает Алису Яндекс, Apple Сири и Google assistant человекоподобными [10].

В качестве примера обработки текста на естественном языке введем текст и сформулируем к нему вопрос (рис. 4). В поле «Введите текст» приведен фрагмент текста из Википедии про республику Таджикистан, в поле «Введите вопрос» мы сформулировали вопрос «Столица республики Таджикистан?», в поле «Результаты» показан ответ на вопрос – «город Душанбе».

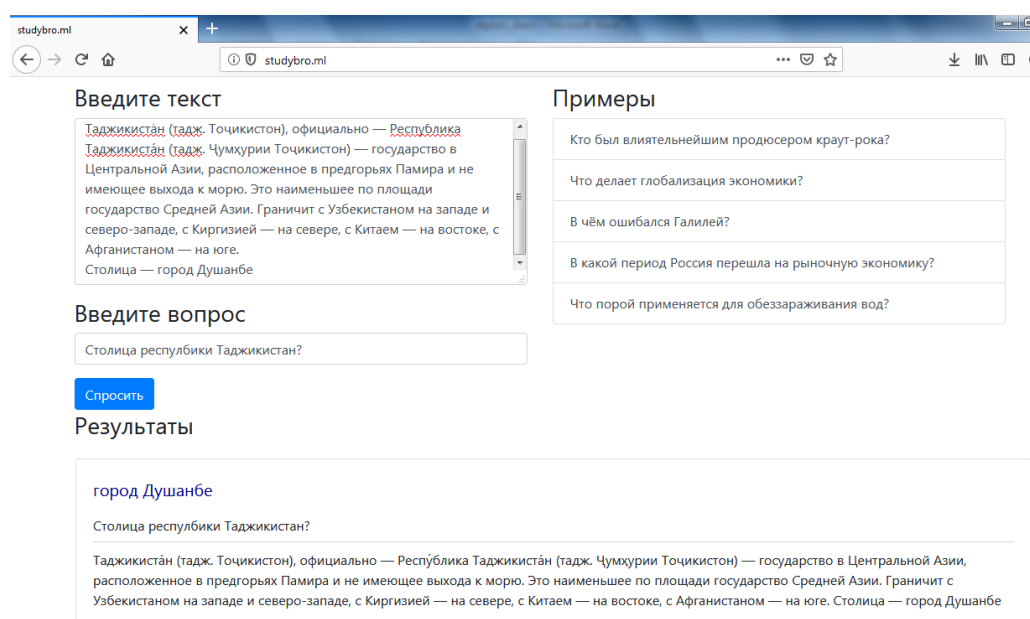


Рисунок 4 – Результат интеллектуального анализа текста на портале studybro.ml

Технология понимания ЕЯ анализирует каждое предложение с целью получения двух вещей: *намерения* (смысла, значения или предполагаемого действия) и *сущностей*. В этом примере извлеченная информация о столице является распознанным намерением, а город (Душанбе) и республика Таджикистан являются сущностями. Пользователь может задать тот же самый вопрос множеством различных способов, и хорошая система понимания ЕЯ всегда будет в состоянии извлечь из пользовательского запроса правильное намерение и сущности. Далее программная система сможет использовать эту ранее извлеченную информацию для подготовки модели и обучения, об этом говорили в первом разделе [13].

1.4.2 Обработка естественного языка

Разумеется, в человеко-компьютерном взаимодействии гораздо больше задач, чем только одно понимание намерения заданного предложения. Технология обработки ЕЯ охватывает все, что имеет

отношение к человеко-компьютерному взаимодействию на естественном языке. И технология понимания ЕЯ является всего лишь подмножеством задач в большем множестве, которым является технология обработки ЕЯ. Технология обработки естественного языка, в частности, включает в себя такие задачи, как:

- *машинный перевод*: преобразование текста из одного языка в другой;

- *генерация естественного языка*: обратная по отношению к пониманию ЕЯ задача; преобразование структурированных данных (обычно из баз данных) в текстовые предложения, понятные человеку. Например, на основе сравнения двух строк в базе данных с информацией о погоде может быть сформулировано вот такое предложение: «сегодня в Дели температура воздуха составляет 26 градусов по Цельсию, что ниже на 2 градуса, по сравнению со вчерашним днем» [13].

- *анализ мнений*: просканировать фрагмент текста (твит, обновление статуса Facebook, обзоры, комментарии и т. д.), касающийся товара, человека, события или географического места, чтобы определить общее мнение (отрицательное или положительное) в отношении соответствующего субъекта;

- *распознавание именованных сущностей*: в том или ином тексте определить, какие фрагменты в тексте соотносятся с именами собственными, связанными с людьми или географическими объектами, и тип каждого определенного имени (человек, местоположение, организация);

- *извлечение связей*: извлечение связей между сущностями, присутствующими во фрагменте текста, в частности родственные отношения, причины, признаки и т. д. [13].

Обработка ЕЯ намного шире, чем несколько упомянутых выше задач, причем каждая из них является предметом независимого исследования.

1.5 Компьютерное зрение

В данном пункте обсудим методы искусственного интеллекта, с помощью которых происходит обработка визуальных данных. *Визуальные данные* включают в себя как неподвижные изображения – фотографии, так и динамически изменяющееся изображение – видео. Общее название этого термина – компьютерное зрение, т. е. способность компьютера «видеть». Компьютеры по своей природе не способны обрабатывать изображения так же хорошо, как текст. Методы обработки изображений в соединении с интеллектуальными алгоритмами ИИ позволяют машинам «видеть» изображения, идентифицировать и распознавать объекты и людей [11].

Обнаружение объектов. Сцена на фотографии может содержать десятки или даже сотни объектов. Но обычно в зависимости от ситуации внимание фокусируется только на нескольких объектах. Давайте назовем их «интересующими». Распознавание изображения обозначает способность машины обнаруживать интересующие объекты, которые могут меняться в зависимости от ситуации [11].

В качестве возможного развития данного проекта можно указать направление по использованию распознавания изображения. Например, сформировать тестовые задания исходя из изображений, которых распознает искусственный интеллект. На самом деле уже во многих крупных интернет-магазинах и поисковых системах используют распознавание изображений. Например, на Amazon можно делать фотографию объекта, идентифицировать его с вероятностью 99%, а затем купить его.

1.6 Библиотека DeepPavlov

1.6.1 Основные понятия

Лаборатория нейронных систем и глубокого обучения МФТИ с 2017 года активно разрабатывает библиотеку DeepPavlov – открытую библиотеку для создания диалоговых систем. Она содержит набор претренированных компонент для анализа языка, с помощью которых можно эффективно решать задачи обработки текстов на естественном языке. Например, в автоматическом режиме требуется организовать ответы на часто задаваемые вопросы клиентов. В нашем проекте мы использовали данную библиотеку для интеллектуального анализа текста [16].

Разработка навыка. Навыком (skill) в DeepPavlov называется сущность, которая независимо от функционала (text classification, open-domain question answering и др.) имеет унифицированный формат ввода и вывода. Навыки созданы для того, чтобы их можно было скомпоновать в единый стек простой диалоговой системы, которая при получении запроса берет ответ из навыка с наивысшей уверенностью (confidence). Классификация текста (text classification) может решить следующие проблемы:

1. Распознавать намерения пользователя в любой платформе.
2. Различают спам и не спам.
3. Определите настроение отзывов клиентов.

1.6.2 Ответы на вопросы в открытой области

Умение отвечать на фактологические вопросы является ключевой особенностью любой системы. Формально говоря, дать ответ, основанный на собрании документов, охватывающих широкий спектр

тем, называется *ответом на вопросы в открытой области* (Open Domain Question Answering, ODQA). Задача ODQA сочетает в себе проблемы поиска документов (поиск соответствующих статей) с задачей машинного понимания текста (определение диапазона ответов из этих статей). Система ODQA может использоваться во многих приложениях. Чат-боты применяют ODQA для ответа на запросы пользователей, в то время как бизнес-ориентированные решения Natural Language Processing (NLP) используют ODQA для ответа на вопросы, основанные на внутренней корпоративной документации. На рис. 5 показан типичный диалог с системой ODQA [17].

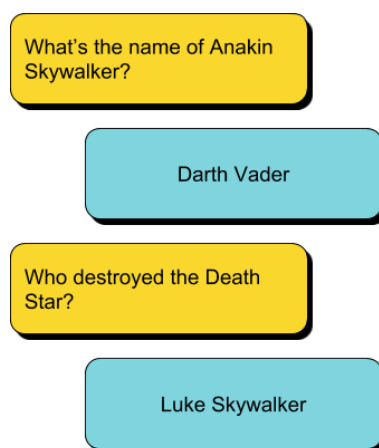


Рисунок 5 – диалог с системой ODQA

Существует несколько подходов к архитектуре системы ODQA. Модульная система ODQA состоит из двух компонентов: первый (ранкер) должен быть в состоянии найти соответствующие статьи в базе данных (например, Wikipedia), тогда как второй (читатель) извлекает ответ из одной статьи или небольшая коллекция статей, найденных ранкером. В дополнение к строго двухкомпонентным системам ODQA существуют гибридные системы, основанные на нескольких ранкерах, в которых последний ранкер в конвейере объединяется с модулем извлечения ответов, как правило, посредством обучения с подкреплением [16].

Далее в тексте мы приведем иллюстрации по обучению и использованию ODQA-системы в рамках проекта studybro.tj. Код, использованный в данной работе, приведен в приложениях, а описанием в практической части.

1.6.3 Описание модели

Архитектура навыка DeepPavlov ODQA является модульной и состоит из двух компонентов: ранкера и ридера. Чтобы ответить на любой вопрос, ранкер сначала получает несколько соответствующих статей из коллекции статей (или поля для ввода текста, источника), а затем ридер (читатель) тщательно сканирует их, чтобы найти ответ. Рейтинг основан на DrQA [15], предложенным Facebook Research. В частности, в подходе DrQA используется хэширование униграмм-биграмм и сопоставление TF-IDF (TF — term frequency, IDF — inverse document frequency), предназначенное для эффективного возврата подмножества соответствующих статей на основе вопроса. Считыватель основан на R-NET [16], предложенным Microsoft Research Asia, и его внедрении Wenxuan Zhou. Архитектура R-NET — это сквозная модель нейронной сети, цель которой — ответить на вопросы, основанные на данной статье. R-NET сначала сопоставляет вопрос и статью через рекуррентные сети на основе закрытого внимания, чтобы получить представление статьи с учетом вопросов. Затем механизм самосогласованного внимания уточняет представление, сопоставляя статью с самой собой, что эффективно кодирует информацию из всей статьи. Наконец, указатель сети находит позиции ответов в статье. На схеме ниже показана архитектура системы DeepPavlov ODQA.

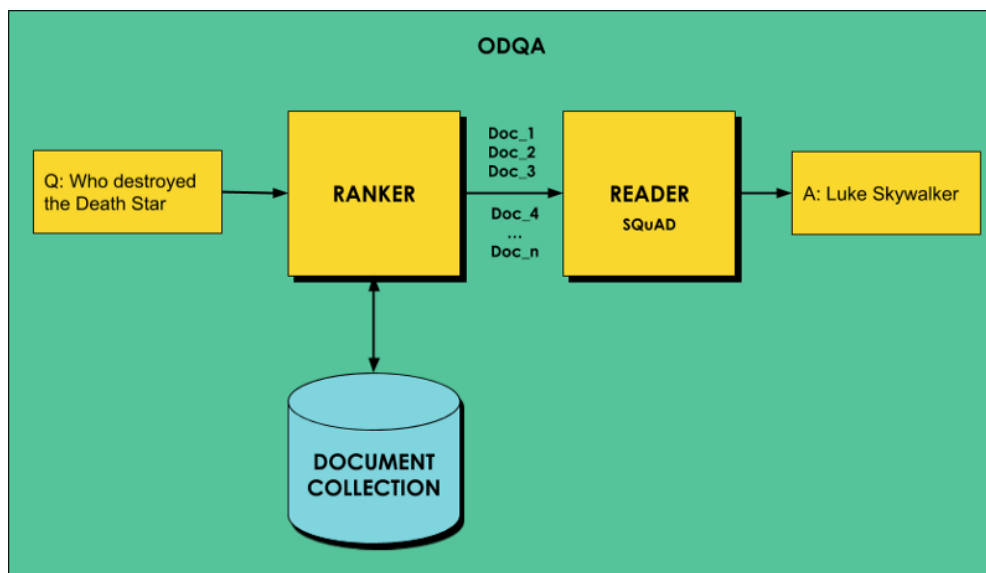


Рисунок 13 – архитектура системы DeepPavlov ODQA

Модели ODQA описаны в отдельных файлах конфигурации в папке `deeppavlov / configs / odqa`. В DeepPavlov файлы конфигурации определяют конвейер обработки данных и состоят из четырех основных разделов: `dataset_reader`, `dataset_iterator`, `chainer` и `train`.

Входные данные конфигурации `en_odqa_infer_wiki` – это пользовательский запрос (`question_raw`), а выходные данные – `best_answer` модели. Выходные данные модели зависят от ранкера, определенного как первый компонент канала ODQA, и считывателя, определенного как последний компонент канала ODQA.

2 Практическая часть

Практическая часть данной работы содержит описание процесса создания инфологической модели данных, даталогической модели данных, реализацию проектируемой базы данных в СУБД MySQL, разработку интерфейса и администрирование базы данных.

Инфологическая модель данных представлена в виде диаграммы типа «Таблица-связь» и на языке проектных решений. Физическая модель данных представляет собой базу данных в среде MySQL. Интерфейс пользователя разработан на PHP. Администрирование реализовано с помощью программы dbForge Studio for MySQL [3].

2.1 Построение инфологической модели данных

База данных предназначена для хранения и добавления данных о курсах, тестовых заданиях и пользователях. В ходе проведения анализа предметной области были выделены следующие объекты.

- Пользователи;
- Курсы;
- Задачи;

У объекта «Пользователи» можно выделить следующие атрибуты:

- 1) Номер пользователя (НП);
- 2) Логин
- 3) Пароль;
- 4) Фамилия;
- 5) Имя;
- 6) Email (не обязательный атрибут);

У объекта «Курсы» можно выделить следующие атрибуты:

- 1) Номер курса (НК);
- 2) Название;
- 3) Описание;

4) Общие баллы за курс;

У объекта «Задачи» можно выделить следующие атрибуты:

- 1) Код задачи;
- 2) Задача (в виде html документа);
- 3) Количество баллов за задачу
- 4) Ответ(ы) (с указанием правильного).

Анализ определенных выше объектов и атрибутов позволяет выделить сущности проектируемой базы данных и построить ее инфологическую модель.

К стержневым сущностям можно отнести:

1) **Пользователи** (НП(id_user), Логин, Пароль, Фамилия, Имя, Email). Эта сущность предназначена для хранения информации о пользователях.

2) **Курсы** (НК(id_kurs), Название, Описание, Общие баллы за курс). Сущность содержит информацию о курсах.

К обозначающим сущностям можно отнести:

1) **Задачи** (Код задачи, Задача; (в виде html документа), Количество баллов за задачу, Ответы) [НК(id_kurs)].

Стержневые сущности и обозначения связаны между собой ассоциациями:

1) **Мои курсы** [Пользователи М, Курсы N] (НК(id_kurs), НК(id_kurs));

2) **Заработанные баллы** [Пользователи М, Курсы N, Задачи P] (НК(id_kurs), НК(id_kurs), Код задачи, Заработанные].

Инфологическую модель на языке ER-диаграмм можно представить следующим образом (Рисунок 5):

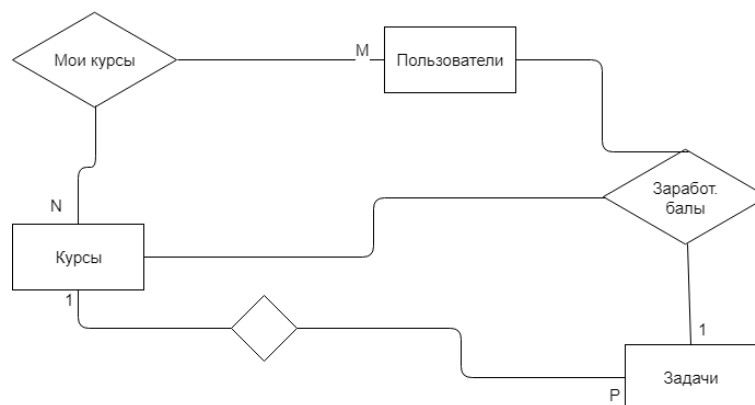


Рисунок 5 – Инфологическая модель базы данных "Образовательный портал", построенная с помощью языка ER-диаграмм

Инфологическую модель на языке "Таблицы-связи" можно изобразить следующим образом (Рисунок 6):

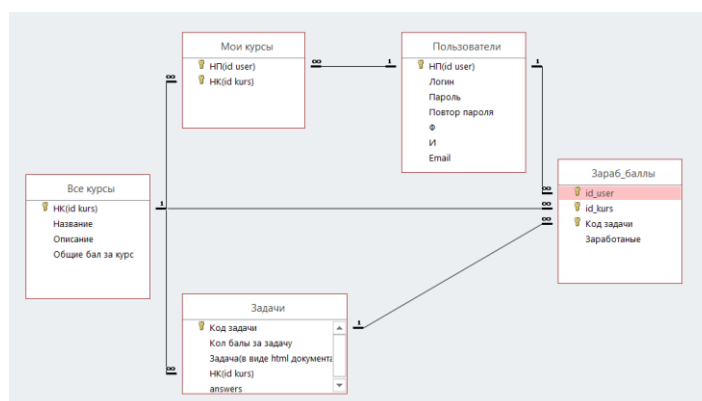


Рисунок 6 – Инфологическая модель базы данных "Образовательный портал", построенная с помощью языка «Таблица-СВЯЗЬ»

2.1.1 Проектирование целостности базы данных

В соответствии с процедурой с проектирования баз данных каждая из полученных сущностей должна быть представлена базовой таблицей. Далее в тексте желтым цветом выделены внешние ключи. Требуемые таблицы можно описать с использованием псевдокода следующим образом:

СОЗДАТЬ ТАБЛИЦУ **Пользователи** *(Стержневая сущность)
ПЕРВИЧНЫЙ КЛЮЧ (НП)

ПОЛЯ (НП Целое Без знака, Логин Текст 45, password Текст 45, Имя Текст 45, Фамилия Текст 45, Email);

СОЗДАТЬ ТАБЛИЦУ **Курсы** *(Стержневая сущность)
ПЕРВИЧНЫЙ КЛЮЧ (НК)

ПОЛЯ (НК Целое, Название Текст 50, Описание Текст 225, Общие балы за курс Целое);

СОЗДАТЬ ТАБЛИЦУ **Задачи** *(Обозначающая сущность)
ПЕРВИЧНЫЙ КЛЮЧ (Код задачи)

ПОЛЯ (Код задачи Целое, Задача (в виде html документа) Текст 225, Количество балы за задачу Целое, Ответы Текст т.к. там может быть все что угодно) ВНЕШНИЙ КЛЮЧ (НК ИЗ Курсы

NULL ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Курса ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Курс.НК ОГРАНИЧИВАЕТСЯ);

СОЗДАТЬ ТАБЛИЦУ **Мои курсы** *(Связывает Пользователи и Курсы)

ПЕРВИЧНЫЙ КЛЮЧ (НП, НК)

ВНЕШНИЙ КЛЮЧ (НП ИЗ Пользователи

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Пользователи ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Пользователи.НП

КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (НК ИЗ Курсы

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Курсы ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Курсы.НК КАСКАДИРУЕТСЯ)

ПОЛЯ (НП Целое, НК Целое)

ОГРАНИЧЕНИЯ (Значения полей НП и НК должны принадлежать набору значений соответствующих полей таблиц Пользователи и Курсы; при нарушении вывод сообщения "Такого курса не добавлять" или "Такого пользователя нет");

СОЗДАТЬ ТАБЛИЦУ Заработанные балы *(Связывает Пользователи и Курсы и Задачи)

ПЕРВИЧНЫЙ КЛЮЧ (НП, НК, Код задачи)

ВНЕШНИЙ КЛЮЧ (НП ИЗ Пользователи

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Пользователи ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Пользователи.НП КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (НК ИЗ Курсы

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Курсы ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Курсы.НК КАСКАДИРУЕТСЯ)

ВНЕШНИЙ КЛЮЧ (Код задачи ИЗ Задачи

NULL-значения НЕ ДОПУСТИМЫ

УДАЛЕНИЕ ИЗ Задачи ОГРАНИЧИВАЕТСЯ

ОБНОВЛЕНИЕ Задачи. Код задачи КАСКАДИРУЕТСЯ);

ПОЛЯ (НП Целое, НК Целое, Код задачи Целое)

ОГРАНИЧЕНИЯ (Значения полей НП, НК и Код задачи должны принадлежать набору значений соответствующих полей таблиц Пользователи, Курсы и Задачи; при нарушении вывод сообщения

"Такого курса не добавлять" или "Такого пользователя нет" или "Такого задачи нет").

2.1.2 Нормализация таблиц

Теперь следует проверить, не нарушены ли в данном проекте какие-либо принципы нормализации, т.е. что любое неключевое поле каждой таблицы: функционально зависит от полного первичного ключа, а не от его части (если ключ составной); не имеет функциональной зависимости от другого неключевого поля.

Проверка показала, что:

1) Анализ сущностей «Заработ.Баллы» и «Мои курсы», состоящих из составного ключа и одного неключевого поля, показал, что неключевое поле в этих сущностях не зависит функционально от какой-либо части составного ключа;

2) В сущности «Пользователи» поле «Логин» стало неключевым из-за ввода цифрового первичного ключа НП, заменяющего текстовый возможный ключ «Логин». Это позволило уменьшить объем хранимых данных в таблицах «Заработ.Баллы» и «Мои курсы». Функциональных связей между неключевыми полями «Фамилия» и «Имя» или «Пароля нет», следовательно, сущность «Пользователи» нормализована;

3) В сущности «Курсы» поле «Название» стало неключевым из-за ввода цифрового первичного ключа НК, который заменяет текстовый ключ «Название». Это позволило уменьшить объем хранимых данных в таблицах «Заработ.Баллы», «Мои курсы» и «Задачи». Обратим внимание, что это исключает ошибку при создании курсов с одинаковыми названиями. Функциональных связей между неключевыми полями «Описание» и «Общ.баллы» нет, следовательно, сущность «Курсы» нормализована;

4) Сущность «Задачи» имеет несоставной первичный ключ «Код задачи». Функциональных связей между неключевыми полями нет, следовательно, сущность нормализована.

Синтаксис языка SQL для построения даталогической модели данных приведен в Приложении 1.

2.2 Создание таблиц на языке SQL

Прежде чем начать данный пункт хотел бы внести небольшие изменения в названиях т.к. Используемые в проекте языки программирования SQL и PHP некорректно работают с кириллицей, поэтому приведем необходимые названия с использованием букв латинского алфавита (Таблица 1). При обозначении атрибутов для наглядности мы использовали символ минус (“-“).

Таблица 1 – Соответствие в обозначение таблиц, атрибутов в dbForge Studio for MySQL

Название на русском	Название на английском
Пользователи (таблица)	users
id_user	id_user
-Логин	-login
-Пароль	-password
-Повтор пароля	-repeat_password
-И	-N
-Ф	-S
-Email	-email
Все курсы (таблица)	all_kurs
id_kurs	id_kurs
-Название	-name
-Описание	-description
-Общие балы за курс	-all_ball_kurs
Мои курсы (таблица)	my_kurs
id_user	id_user
id_kurs	id_kurs
Задачи (таблица)	tasks
Код задачи	id_task
-Задача (в виде html документа)	-task
-Кол бал за задачу	-ball_task

-Ответы	-answers
НК (id kurs)	id_kurs
Зараб.баллы (таблица)	points_earned
id_user	id_user
id_kurs	id_kurs
Код задачи	id_task
-Заработанные	-earned

Далее в тексте красным цветом выделены первичные ключи.

Создание БД в dbForge

Создаем базу данных в dbForge MySQL:

CREATE DATABASE webbro

CHARACTER SET utf8

COLLATE utf8_general_ci;

Создаем таблицу Пользователи

Создаем таблицу users и заполняем данными с файла dani.xlsx :

CREATE TABLE webbro.users (
id_users int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
login varchar(45) NOT NULL,
password varchar(45) NOT NULL,
N varchar(45) NOT NULL,
S varchar(45) NOT NULL,
email varchar(50) DEFAULT NULL,
PRIMARY KEY (id_users)
)
ENGINE = INNODB,
CHARACTER SET utf8,
COLLATE utf8_general_ci;

Создаем таблицу все курсы

Создаем таблицу all_kurs и заполняем данными из файла dani.xlsx :

CREATE TABLE webbro.all_kurs (

```

id_kurs int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
name varchar(150) NOT NULL,
description varchar(350) DEFAULT NULL,
id_user_creator int(11) UNSIGNED NOT NULL,
all_ball_kurs int(11) UNSIGNED NOT NULL,
PRIMARY KEY (id_kurs)
)
ENGINE = INNODB,
CHARACTER SET utf8,
COLLATE utf8_general_ci;

```

```

ALTER TABLE webbro.all_kurs
ADD CONSTRAINT FK_id_user_creator FOREIGN KEY
(id_user_creator)
REFERENCES webbro.users (id_users);

```

Создаем таблицу мои курсы

Создаем таблицу my_kurs и заполняем данными из файла dani.xlsx

:

```

CREATE TABLE webbro.my_kurs (
id_user int(11) UNSIGNED NOT NULL,
id_kurs int(11) UNSIGNED NOT NULL,
num_earned_ball int(11) UNSIGNED NOT NULL,
PRIMARY KEY (id_user, id_kurs)
)
ENGINE = INNODB,
CHARACTER SET utf8,
COLLATE utf8_general_ci;

```

```

ALTER TABLE webbro.my_kurs

```



```
ADD CONSTRAINT FK_my_kurs_id_kurs FOREIGN KEY (id_kurs)
REFERENCES webbro.all_kurs (id_kurs);
```

```
ALTER TABLE webbro.my_kurs
ADD CONSTRAINT FK_my_kurs_id_user FOREIGN KEY (id_user)
REFERENCES webbro.users (id_users);
```

Создаем таблицу Задачи

Создаем таблицу tasks и заполняем данными из файла dani.xlsx :

```
CREATE TABLE webbro.tasks (
    id_task int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    task varchar(1555) NOT NULL,
    ball_task int(11) UNSIGNED NOT NULL,
    id_kurs int(11) UNSIGNED NOT NULL,
    PRIMARY KEY (id_task)
)
ENGINE = INNODB,
CHARACTER SET utf8,
COLLATE utf8_general_ci;
```

```
ALTER TABLE webbro.tasks
ADD CONSTRAINT FK_tasks_id_kurs FOREIGN KEY (id_kurs)
REFERENCES webbro.all_kurs (id_kurs);
```

В среде dbForge структура данных выглядит следующим образом
(Рисунок 7):

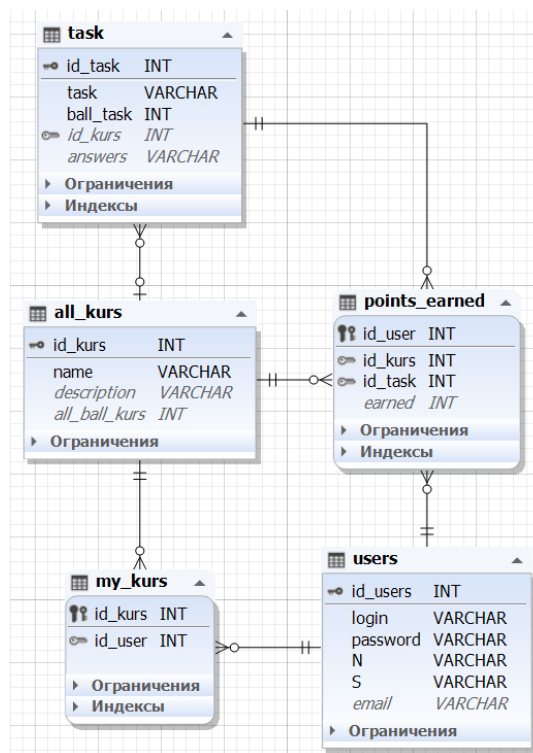


Рисунок 7 – Инфологическая модель базы данных "Образовательный портал", построенная с помощью языка «Таблица-связь» в dbForge.

2.3 Реализация проектируемой базы данных в СУБД MySQL

Получение данных. Данные для таблиц **Пользователи**, **Курсы** и **Задачи** были сгенерированы специально для данного проекта. В итоге для таблицы **Пользователи** данные выглядят следующим образом:

Таблица 2 – Пользователи

	id_users UNSIGNED INT(11)	login VARCHAR(45)	password VARCHAR(45)	N VARCHAR(45)	S VARCHAR(45)	email VARCHAR(50)	
▶	1	admin	951429850	admin	WebBro	(null)	
	2	shamil	95	Шамил	Курбонов	shamil-k@list.ru	
	3	muhsin	2342sd	Мухсин	Исмаев	ismatov.98@mail.ru	
	4	ansor	324reer43	Ансор	Султонов	sultonov32@yandex.ru	
	5	anil	dfg3423	Артём	Николаев	(null)	
	6	basir	g3deg4	Павел	Валуев	(null)	
	7	das	dfg453	Глеб	Данилов	sferg@mail.ru	
	8	gwes	fgh45s	Фёдор	Козлов	(null)	
	9	sdg	45hrfgh5	Ярослав	Смирнов	ofas1996@gmail.com	
	10	ssdwe	erh44h45	Виктор	Морозов	(null)	
	11	dsfs	dfhg45yhhf	Артур	Попов	(null)	

В итоге для таблицы **Курсы** данные выглядят следующим образом:

Таблица 3 Курсы

id_kurs UNSIGNED INT(11)	name VARCHAR(50)	description VARCHAR(255)	all_bal_kurs UNSIGNED INT(11)
1	Java Script для начинающих	Цель данного курса - познакомить слушателей с основами программирования на JavaScript и подготовить их для практического применения данного инструм...	80
2	Web-технологии	Курс Web-технологии посвящен разработке web-приложений среднего масштаба, иначе говоря сайтов. Курс является в большой мере обзорным и знакомит б...	45
3	Программирование на Python	В этом курсе по программированию на языке Python вы познакомитесь с базовыми понятиями программирования.	100
4	курс базы данных	Курс введения в базы данных знакомит слушателями с историей создания систем обработки структурированных данных, подходами к обработке информации...	70
5	Основы статистики	В рамках трехнедельного курса рассматриваются подходы к описанию получаемых в исследованиях данных, основные методы и принципы статистического ...	45
6	Математика	В данном курсе вы научитесь решать дифф ур второго рода!	25

В итоге для таблицы **Задачи** данные выглядят следующим образом:

Таблица 4 – Задачи

id_task UNSIGNED INT(11)	task VARCHAR(255)	ball_task UNSIGNED INT(11)	id_kurs UNSIGNED INT(11)	answers VARCHAR(125)
1	2*2 = <input type="text" name="v1" placeholder="Ответ">	2	1	4
2	Когда появился JavaScript?	3	1	3
3	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	4	1	14
4	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	5	3	true
5	<input class="btn1" type="text" name="uname" placeholder="Ответ" id="ourInput" >	6	3	3
6	Расширяемый язык разметки это?	7	2	xml
7	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	5	4	2
8	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	4	4	4
9	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	6	4	6
10	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	7	5	7
11	<input class="btn1" type="text" name="uname" placeholder="Username" id="ourInput" >	5	5	8

В итоге для таблицы **Мои курсы** данные выглядят следующим образом:

Таблица 5 – Мои курсы

id_user UNSIGNED INT(11)	id_kurs UNSIGNED INT(11)	
1	1	
1	2	
1	5	
2	2	
2	3	
2	4	
3	3	
4	1	
4	5	
6	1	
7	4	
9	3	

В итоге для таблицы **Заработанные баллы** данные выглядят следующим образом:

Таблица 6 – Заработанные баллы

id_user UNSIGNED INT(11)	id_kurs UNSIGNED INT(11)	id_task UNSIGNED INT(11)	earned UNSIGNED INT(11)	
1	1	2	2	
1	1	3	4	
2	2	7	3	
56	1	1	2	
56	1	2	3	
56	2	6	7	
71	1	1	2	

В итоге для каждой таблицы получено следующее количество записей:

- 1) Пользователи – 10 записи;
- 2) Курсы – 6 записи;
- 3) Задачи – 11 записей;
- 4) Мои курсы – 16 записи;
- 5) Заработанные баллы – 8 записей;

2.4 Разработка интерфейса пользователя и администратора

Основная задача интерфейса предоставить пользователю возможность просматривать информацию о курсах (всех, выбранных), отображать рейтинг. Функционально интерфейс разделен на пользовательский и администраторский. В связи с этим выделено две группы пользователей: Пользователь, Администратор. Текст скриптов приведен в приложениях 2-4.

В пользовательском интерфейсе реализованы следующие функции:

- просмотр профиля пользователя;
- диалог для входа в систему;
- просмотр и выбор курсов;
- просмотр и выполнение тестовых заданий.

Для этого было реализовано на языке PHP восемь скриптов.

1) index2.php

Назначение: Вывод каркаса

Скрипт обращается к базе данных и выводит информацию о пользователе и дает подключение к БД.

Визуальный результат работы скрипта index2.php приведен на рисунке 8.

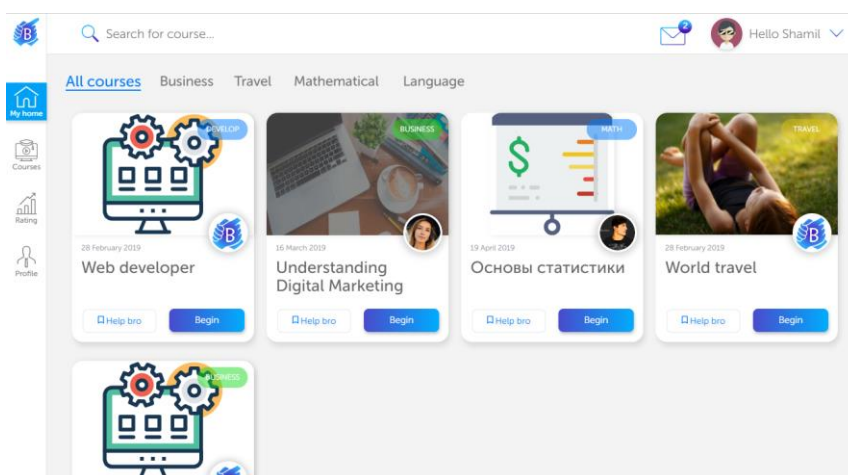


Рисунок 8 – Результат работы скрипта index2.php

2) allkurs.php

Назначение: Возможность просмотр всех курсов и выбор нужного курса.

Визуальный результат работы скрипта allkurs.php приведен на рисунке 9.

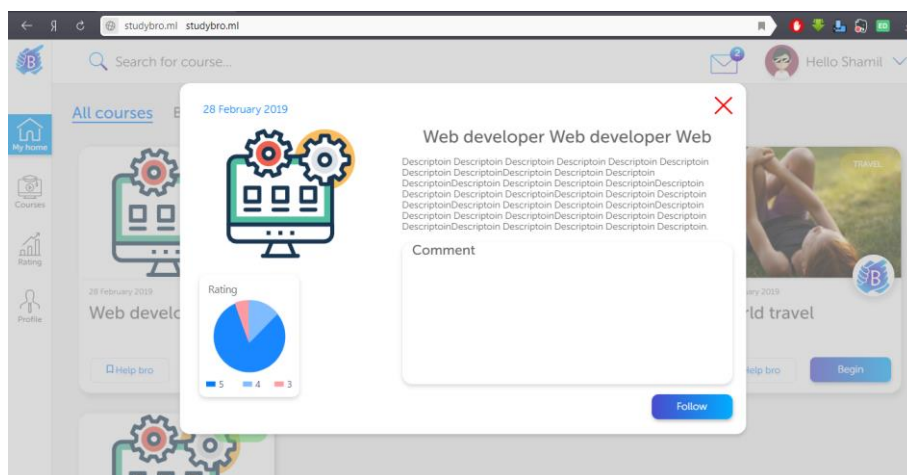


Рисунок 9 – Результат работы скрипта allkurs.php

3) mykurs.php

Назначение: Вывод информации о курсах пользователя

Обращается к базе данных и выдает информацию о курсах пользователя в виде таблиц, предоставляет возможность просмотра рейтинга.

Визуальный результат работы скрипта mykurs.php приведен на рисунке 10.

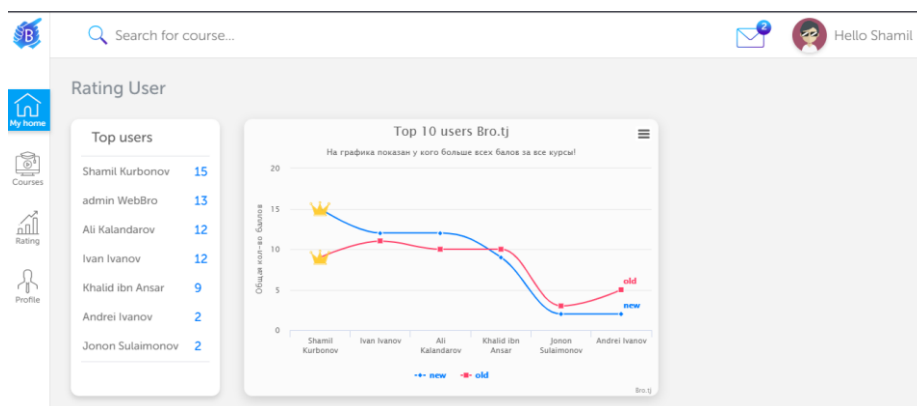


Рисунок 10 – Результат работы скрипта mykurs.php

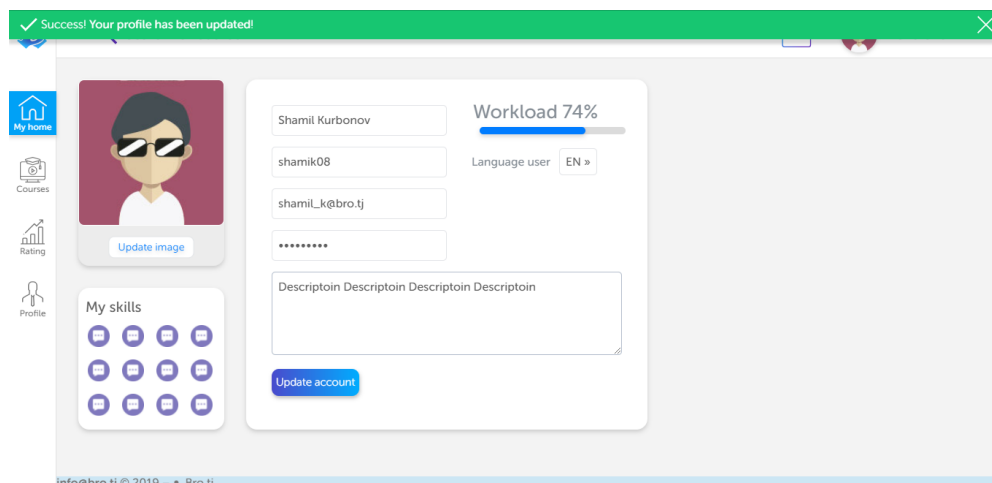


Рисунок 11 – Результат работы скрипта user.php

В интерфейсе администратора реализованы следующие функции:

- редактирование таблиц
- добавление таблиц
- удаление таблиц
- просмотр таблиц

Назначение: Редактирование таблиц (вставка, удаление)

Для каждой таблицы из базы данных был написан отдельный скрипт.

Визуальный результат работы скрипта редактирование таблиц приведен на рисунке 12.

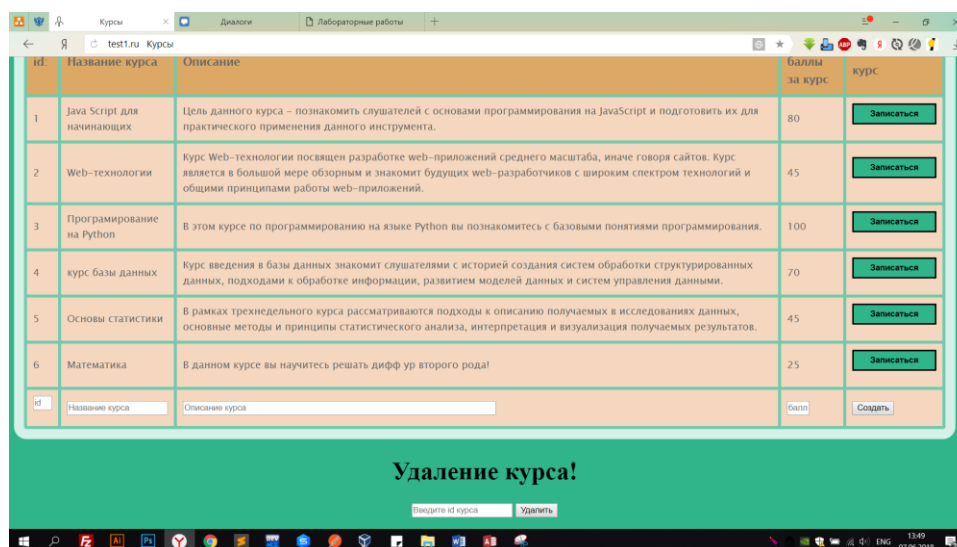


Рисунок 12 – Пример интерфейса администратора. Таблицы Курсы

2.6 Требования к модели

Для использования модели сначала требуется установить библиотеку и выполнить настройки:

```
python -m deeppavlov install  
deeppavlov/configs/odqa/en_odqa_infer_wiki.json
```

Система DeepPavlov ODQA имеет две модели на основе Википедии. Английская модель Википедии требует 35 ГБ локального хранилища, тогда как русская версия занимает около 20 ГБ. Дампы Википедии могут быть восстановлены с помощью шагов, описанных в документации. Обе модели требуют около 24 ГБ оперативной памяти. Их можно запустить на компьютере объемом 16 ГБ, но размер подкачки должен быть не менее 8 ГБ.

2.7 Взаимодействие с моделью

Мы можем взаимодействовать с моделями через командную строку:

```
python -m deeppavlov interact  
deeppavlov/configs/odqa/en_odqa_infer_wiki.json -d
```

где **-d** обозначает загрузку всех файлов, требуемых моделями.

Кроме того, мы можете запустить модель через код Python:

```
from deeppavlov import configs  
from deeppavlov.core.commands.infer import build_model  
odqa = build_model(configs.odqa.en_odqa_infer_wiki, download =  
True)  
answers = odqa(["Where did guinea pigs originate?", "When did  
the Lynmouth floods happen?", "When is the Bastille Day?"])  
["Argentina", "15-16 August 1952", "14 July 1789"]
```


2.8 Обучение модели

Оба компонента системы ODQA должны проходить обучение отдельно. Компонент для чтения предварительно обучен на Стэнфордском наборе вопросов и ответов (SQuAD). SQuAD – это набор данных для понимания прочитанного, состоящий из вопросов, заданных к статьям Википедии, где ответ на каждый вопрос представляет собой фрагмент текста чтения. При этом вопрос может не иметь ответа [17], Тем не менее документация DeepPavlov полностью описывает, как обучить компонент чтения для данных.

В качестве учебного корпуса мы будем использовать корпус вычислительной биологии PLoS [16]. Он состоит из 300 статей по компьютерной биологии, каждая из которых хранится в отдельном текстовом файле. Для простоты мы будем использовать те же файлы конфигурации, которые используются для системы ODQA на основе Википедии;

Сначала загрузим набор данных и поместим его в папку.

```
from deeppavlov import configs  
from deeppavlov.core.common.file import read_json  
from deeppavlov import configs, train_model  
model_config = read_json(configs.doc_retrieval.en_ranker_tfidf_wiki)  
model_config["dataset_reader"]["data_path"] =  
"/content/SentenceCorpus/unlabeled_articles/plos_unlabeled"  
model_config["dataset_reader"]["dataset_format"] = "txt"  
ranker = train_model(model_config)  
docs = ranker(['cerebellum'])
```

Список документов содержит топ-30 соответствующих статей для запроса «мозжечок».

Также построим модели ODQA и запустил запрос:

```
from deeppavlov import configs
```

```
from deeppavlov.core.commands.infer import build_model
odqa = build_model(configs.odqa.en_odqa_infer_wiki, download =
False)
a = odqa(["what is tuberculosis ?"])
['a disease for which a new drug is desperately needed']
```

2.9 Результаты модели

В примере приведенным в теоретической части (с. 13) мы показали, как модель работает с короткими вопросами, при точности почти 100%. Теперь рассмотрим тот же текст, но вопрос сформулируем следующим образом: “С кем граничит республика Таджикистан?”

Результаты

с Узбекистаном на западе и северо-западе, с Киргизией

С кем граничит

Таджикистан (тадж. Тоҷикистон), официально — Республика Таджикистан (тадж. Ҷумҳурии Тоҷикистон) — государство в Центральной Азии, расположенное в предгорьях Памира и не имеющее выхода к морю. Это наименьшее по площади государство Средней Азии. Граничит с Узбекистаном на западе и северо-западе, с Киргизией — на севере, с Китаем — на востоке, с Афганистаном — на юге. Столица — город Душанбе. Государственный язык — таджикский, при этом Таджикистан — единственное персоязычное государство в бывшей советской Средней Азии. Большинство населения Таджикистана исповедует ислам суннитского толка.

Рисунок 14 – результат вопроса.

Как мы видим на рис. 14 наша модель выдала ответ, но это не точный ответ, так как модель не использовала все выявленные географические сущности (рисунок 15).

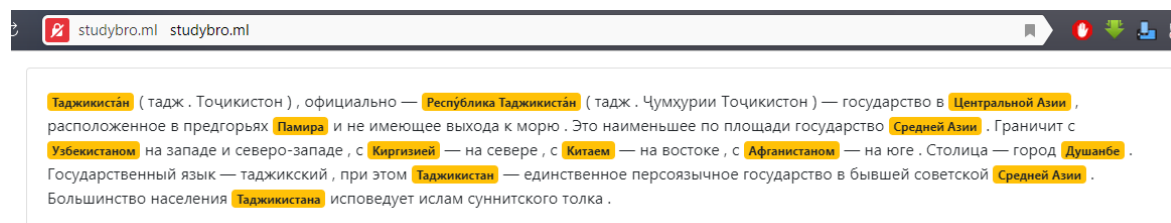


Рисунок 15 – Выявленные географические сущности

Исходя из полученных результатов можно сделать вывод, что при коротких и однозначных вопросах модель выдает точный результат, но при длинных и многозначных вопросах точность результата падает.

В данной работе я описал модель ODQA платформы DeepPavlov. Эта модель основана на двухкомпонентном подходе. При разработке архитектуры ODQA мы всегда должны стремиться к балансу между производительностью модели и требованиями к ресурсам. В настоящее время наиболее эффективные модели основаны на архитектуре BERT (двунаправленные представления кодировщиков от трансформаторов) [8]. Однако системы на основе BERT требуют огромных вычислительных ресурсов.

Заключение

Данная работа посвящена разработке образовательного портала с применением методов искусственного интеллекта.

1. В ходе работы был проведен анализ предметной области, в качестве инструментов для реализации образовательного портала были выбраны языки программирования PHP и SQL.
2. Проведена нормализация спроектированной базы данных, затем она была реализована в СУБД MySQL. В базе данных выделено 5 сущностей, из которых две стержневых, одна обозначающая и две ассоциативных.
3. Разработан интерфейс пользователя и администратора портала (16 скриптов).
4. Для интеллектуальной генерации тестовых заданий к материалам образовательного портала мы использовали библиотеку DeepPavlov, которая показала удовлетворительные результаты при поиске ответов на однозначные и простые вопросы.
5. Проведено тестирование и апробация портала. Разработанный портал доступен по ссылке: <http://studybro.ml/>

3. Социальная ответственность

Занимаясь научно-исследовательской деятельностью, стоит уделить внимание обеспечению безопасности охраны труда и окружающей среды. Таковые факторы рассматривает социальная ответственность, а именно: состояние рабочего места, помещения, режим работы, обеспечение мероприятий по защите трудящихся в случае возникновения чрезвычайной ситуации. Существует международный стандарт ИСО 26000 «Руководство по социальной ответственности», разработанный в 2012 г., в соответствии с которым осуществляется разработка системы социальной ответственности на предприятии. При применении организацией данного стандарта для разработки системы социальной ответственности, она решает ряд проблем, таких как: проблемы, связанные с потребителями, трудовые практики, участие в жизни сообществ и их развитие, права человека, окружающая среда, добросовестные деловые практики.

В данной части бакалаврской работе рассмотрен анализ вредных и опасных факторов производства, методы их предупреждения, организация мероприятий защиты в случае возникновения чрезвычайной ситуации. Научно исследовательская работа представляет собой разработку образовательная – онлайн платформа. Для эффективного ведения разработки обозначенной системы необходимо учитывать социальные, правовые и экологические вопросы охраны здоровья и безопасности жизнедеятельности.

В ходе данного исследования необходимо:

- определить и изучить возможные вредные факторы, влияющие на исполнителей при разработке программного продукта;
- разработать решения для минимизации их влияния;
- произвести анализ разработанных решений, с точки зрения социальной ответственности за моральные, общественные,

экономические и экологические возможные негативные последствия и ущерб здоровью человека в результате их внедрения.

3.1 Правовые и организационные вопросы обеспечения безопасности

Безопасность организации — это такое состояние, которое достигается посредством обеспечения и поддержания защищенности ее персонала и жизненно важных интересов организации от внутренних и внешних угроз с целью уменьшения отрицательных последствий нежелательных событий и достижения наилучших результатов деятельности.

Угроза безопасности организации — это событие, действие или явление, которые посредством воздействия на персонал, финансовые, материальные ценности и информацию могут привести к нанесению вреда здоровью работников и ущерба организации, нарушению или приостановке ее функционирования.

Обеспечение безопасности организации — это деятельность ее должностных лиц, персонала, специального подразделения по безопасности, государственных правоохранительных органов и иных структур, направленная на предотвращение возможного нарушения ее нормального функционирования.

Система безопасности организации — это комплекс организационно-управленческих, экономических, правовых, социально-психологических, профилактических, пропагандистских, режимных и инженерно-технических мер и мероприятий, направленных на обеспечение безопасности организации и ее персонала. Определяющим и изначальным при формировании системы безопасности является концепция безопасности организации, которая представляет собой свод

основных документов, касающихся политики и стратегии безопасности, основных направлений, средств и методов ее обеспечения.

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ [18].

Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. Порядок исчисления нормы рабочего времени на определенные календарные периоды (месяц, квартал, год) в зависимости от установленной продолжительности рабочего времени в неделю определяется федеральным органом исполнительной власти, осуществляющим функции по выработке государственной политики и нормативно-правовому регулированию в сфере труда.

В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания. Время предоставления перерыва и его конкретная продолжительность устанавливаются правилами внутреннего трудового распорядка или по соглашению между работником и работодателем.

Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях, установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд.

Если пользователь постоянно загружен работой с ЭВМ, приемлемой является поза сидя. В положении сидя основная нагрузка

падает на мышцы, поддерживающие позвоночный столб и голову. В связи с этим при длительном сидении время от времени необходимо сменять фиксированные рабочие позы.

Исходя из общих принципов организации рабочего места, в нормативно- методических документах сформулированы требования к конструкции рабочего места. Основными элементами рабочего места программиста являются: рабочий стол, рабочий стул (кресло), монитор, клавиатура, мышь; вспомогательными - пюпитр, подставка для ног. Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования.

Рабочие места с ЭВМ должны располагаться на расстоянии не менее 1,5 м от стены с оконными проемами, от других стен – на расстоянии 1 м, между собой – на расстоянии не менее 1,5 м. При размещении рабочих мест необходимо исключить возможность прямой засветки экрана источником естественного освещения.

При размещении ЭВМ на рабочем месте должно обеспечиваться пространство для пользователя величиной не менее 850 мм. Для стоп должно быть предусмотрено пространство по глубине и высоте не менее 150 мм, по ширине – не менее 530 мм. Располагать ЭВМ на рабочем месте необходимо так, чтобы поверхность экрана находилась на расстоянии 400 – 700 мм от глаз пользователя. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы. Рабочее кресло обеспечивает поддержание рабочей позы в положении сидя, и чем длительнее это положение в течение рабочего дня, тем жестче должны быть требования к созданию удобных и правильных рабочих сидений. Высота

поверхности сиденья должна регулироваться в пределах 400 – 550 мм. Ширина и глубина его поверхности должна быть не менее 400 мм. Поверхность сиденья должна быть плоской, передний край – закругленным. Сиденье и спинка кресла должны быть полумягкими, с нескользящим, не электризующимся и воздухопроницаемым покрытием, материал которого обеспечивает возможность легкой очистки от загрязнения. Опорная поверхность спинки стула должна иметь высоту 280 – 320 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм. Расстояние сцинки от переднего края сиденья должно регулироваться в пределах 260 – 400 мм. Рабочее место должно быть оборудовано устойчивой и просто регулируемой подставкой для ног, располагающейся, по возможности, по всей ширине отводимого участка для ног. Подставка должна иметь ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте до 150 мм и по углу наклона опорной поверхности подставки до 20. Поверхность подставки должна быть рифленой, по переднему краю иметь бортик высотой 10 мм. При организации рабочего пространства необходимо учитывать индивидуальные антропометрические параметры пользователя с соответствующими допусками на возможные изменения рабочих поз и потребность в перемещениях.

Рациональной рабочей позой может считаться такое расположение тела, при котором ступни работника расположены на плоскости пола или на подставке для ног, бедра сориентированы в горизонтальной плоскости, верхние части рук – вертикальный угол локтевого сустава колеблется в пределах 70 – 90, запястья согнуты под углом не более чем 20, наклон головы – в пределах 15 – 20, а также исключены частые ее повороты.

3.2 Производственная безопасность

Вредным называется производственный фактор, воздействие которого на человека в определенных условиях, приводит к заболеванию или снижению работоспособности. В зависимости от уровня и продолжительности воздействия вредный производственный фактор может стать опасным [19].

Опасным считается производственный фактор, воздействие которого на человека в определенных условиях приводит к травме, а также другому внезапному резкому ухудшению здоровья [19].

Были определены вредные факторы, влияющие на условия труда разработчика, а также произведен анализ источников вредных факторов. Полученные данные представлены в виде сводной таблицы 1

Таблица 1 - Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Раз работка	Исс	Раб за ота персональ	
1. Недостаточная освещенность рабочей зоны;			+	1. Параметры освещения определены в СНиП 23-05-95 [1]; 2. Предельно допустимые уровни напряжений прикосновения и токов
2. Повышенный уровень статического электричества;			+	
3. Повышенное значение напряжения			+	

в электрической цепи, замыкание которой может произойти через тело человека. 4.Отклонения показатели микроклимата.				установлены в ГОСТ 12.1.038-82 [2]; 3. Мероприятия по устранению статического электричества перечислены в ГОСТ 12.1.030-81 [3]; 4. Правила защиты от поражения электрическим током приведены в ГОСТ Р 50571.3-94 [4].
--	--	--	--	--

3.3 Анализ опасных и вредных производственных факторов и обоснование мероприятий по их устранению

3.3.1 Недостаточная освещенность рабочего места

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени. Недостаточность освещения снижает производительность труда, увеличивает утомляемость и количество допускаемых ошибок, а также может привести к появлению профессиональных болезней зрения.

Достаточная освещенность рабочего места важна для эффективной работы. Она достигается использованием как естественного, так и искусственного света вместе.

Согласно «приложению В» СНиП 23-05-93 [20] проводимые помещения работы по разработке программного продукта относятся к V разряду зрительной работы. Согласно пункту 4.2 СНиП 23-05-95 [20]

уровень общей освещенности должен быть не меньше 200 лк. Величина нормируемой освещенности помещений составляет 300 лк, что является допустимым. Минимальный размер объекта различения 3-5 мм – работа малой точности. Нормы коэффициента пульсации для обозначенного вида работ составляют не более 20%. В помещениях присутствует как естественное, так и искусственное освещение. Последнее обеспечивается люминесцентными источниками света в потолочных светильниках и светильниками, установленными непосредственно на рабочем месте трудящегося.

3.3.2 Повышенный уровень статического электричества

Статическое электричество является одним из опасных факторов. Оно возникает в результате накопления заряда на поверхности или объеме диэлектриков или на изолированных проводниках. Опасность возникает при разряде этого заряда, поскольку заряд может быть велик, что может вызвать поломку оборудования, угрозу для жизни оператора в результате прохождения заряда через тело человека. Накопление заряда может быть вызвано изоляцией оборудования или оператора.

Для устранения необходимо обеспечить заземление компьютера, чтобы заряд мог уходить на землю, не накапливаясь на корпусе компьютера. Правила заземления и другие мероприятия по устранению статического электричества перечислены в ГОСТ 12.1.030-81 [21].

3.3.3 Повышенное значение напряжения в электрической цепи

При прикосновении к электроустановкам, по которым протекает ток возможно прохождение тока через тело человека, так как земля имеет нулевой потенциал. Сила поражения зависит от времени воздействия, характера тока, состояния человека, места соприкосновения, пути

прохождения тока. Поражающими факторами являются ожоги, сбои в работе сердечной мышцы, нервной системы. Причиной прикосновения является как правило нарушение изоляции изделия и нарушение техники безопасности. Для питания персональных компьютеров в России как правило используется сетевое напряжение 230В переменного тока при частоте 50 Гц. После прохождения через блок питания ток трансформируется в постоянный ток напряжения 5 и 12 В. Согласно ГОСТ 12.1.038-82 [22] допустимо прохождение через тело человека переменного тока с напряжением 2В, силой 0.3мА, постоянного тока 8В, силой 1мА. Очевидно, прикосновение к проводке компьютера в любой из его частей недопустимо, хотя после преобразования в блоке питания может не нести поражающих действий.

Для защиты от прохождения тока по телу необходимо соблюдать технику безопасности использования персонального компьютера, использовать ГОСТ Р 50571.3-94 [23] по защите от поражения электрического тока. При обнаружении нарушений изоляции необходимо в первую очередь отключить прибор от источника питания, использовать изоляционные материалы.

3.4 Экологическая безопасность

Сегодня во всем мире присутствуют проблемы рационального использования природных ресурсов, проблемы охраны окружающей среды. При этом особое внимание уделяется организации воздействия вредных факторов на природу.

Многое сырье, используемое в сборке компьютеров, является токсичным. При добыче и разработке материалов для сбора ПК также выбрасываются в атмосферу десятки вредных веществ, к примеру разработка пластика для покрытия корпуса и других частей монитора и системного блока по данным Википедии на 2018 год, в год во всем мире

производится около 380 миллионов тонн пластика. С 1950 года по 2018, было произведено около 6.3 миллиардов тонн пластика, из них было переработано около 9%, а сожжено - 12 %. Огромное количество пластиковых отходов неизбежно попадает в окружающую среду. Исследования предполагают, что тела 90% морских птиц содержат в себе пластик. Отходы производства также не исчезают, превращаясь в свалки, или перерабатываются, оказывая плохое влияние на экологию.

При выполнении работы на персональных компьютерах, некоторые из частей компьютера могут прийти в негодность или устареть, тогда необходимо их утилизировать. В качестве отходов в результате деятельности программирования может выступать бумага, отходы электроники, ртутьсодержащие лампы. Эти твердые бытовые отходы загрязняют окружающую среду и их необходимо перерабатывать, чтобы максимально снизить их вредное воздействие. Для складирования мусора перед утилизацией предлагается вывоз на территории, предназначенные для этого. Также, существует вариант нахождения способа вторичного использования оборудования. Утилизация электроники производится специальными компаниями и является обязательной, в особенности утилизация ртутьсодержащих ламп. Отработанные ртутьсодержащие лампы подлежат проведению демеркуризации.

3.5 Безопасность в чрезвычайных ситуациях

В процессе работы на персональном компьютере одним из чрезвычайной ситуации, которую создаёт компьютер может стать пожар. Также возможно другие ЧП непосредственно при работе с ПК к примеру, может случиться наводнение или землетрясение. Пожар в течении работы персонального компьютера может возникнуть в результате замыкания электропроводки, приводящее к возгоранию аппаратуры. Далее огонь

может распространиться на другие объекты рабочего помещения, создавая техногенную чрезвычайную ситуацию.

Федеральный закон №123 "Технический регламент о требованиях пожарной безопасности" [24] устанавливает основные принципы установления пожарной безопасности. Для предупреждения возникновения пожаров необходимо прежде всего следовать технике безопасности пользования персональным компьютером. Приведенный технический регламент предусматривает предотвращение пожаров путем исключения условий образования горючей среды (статья 49) и исключений условий образования в горючей среде источников зажигания (статья 50).

Исключение условий образования горючей среды в сложившихся рабочих условиях включает в себя: применение минимального количества горючих веществ и материалов и наиболее безопасного их размещения, изоляция источников зажигания от горючих веществ, понижение концентрации окислителя в горючей среде в защищаемом объеме, удаление из помещений пожароопасных отходов производства, отложений пыли и пуха. Исключение источников зажигания достигается путем применения оборудования, исключающих образование статического электричества, устройство молниезащиты зданий, сооружений и оборудования, применение устройств, исключающих возможность распространения пламени из одного объема в смежный.

При возникновении пожара необходимо обеспечить технические и организационные мероприятия по оповещению, эвакуации людей, тушению 75 пожара. При обнаружении пожара необходимо оповестить пожарную службу посредством пожарной сигнализации и провести эвакуацию через эвакуационные пути и выходы. Ликвидация последствий пожара обеспечивается автоматическими установками пожаротушения и пожарными службами.

Вывод

Проанализировав и оценив условия труда в рабочей зоне, где было разработано дипломная работа, можно сделать следующий вывод по производственной и экологической безопасности программиста и непосредственно работы серверных компьютеров:

Рассмотренная нами рабочая зона соответствует нормам трудового законодательства к условиям проекта, но соответственно для работы более эффективного и охлаждённого рабочего пространство было принято решение поставить кондиционер.

Рабочая зона в котором находятся сервера, относится к помещениям без повышенной опасности поражения электрическим током.

Что касается охране окружающей среды можно сказать, что деятельность рабочей зоны и в целом моего проекта не является экологически опасной.

По состоянию пожаробезопасности помещение сервера оборудовано всеми необходимыми датчиками и, следовательно, соответствует всеми нормами пожаробезопасности.

4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» бакалаврской работе является: разработка образовательного портала с применением методов искусственного интеллекта, также создание сервера, отвечающего современным требованиям в области ресурсоэффективности и ресурсосбережения.

Для реализации поставленной цели, в данном разделе производится расчет экономической оценки создания, реализации и тестирования алгоритмов распознавания объектов, для чего необходимо:

- Выполнить обзор существующих решений в исследуемой области, рассматривать технологии, библиотеки, фреймворки и прочее программное обеспечение и соответственно с ним организовать и спланировать работы.
- Разработать календарный план работ.
- Рассчитать затраты на материалы.
- Рассчитать заработную плату исполнителей.
- Рассчитать затраты на электроэнергию.
- Рассчитать амортизационные расходы.
- Рассчитать накладные расходы.
- Определить затрат на реализацию проекта.
- Оценить экономическую эффективность.

4.1 Организация и планирование работ

Организация и планирования проекта необходима для того, чтобы правильно спланировать занятость участников проекта и сроки проведения работ. В таблице 5.1 представлено планирование загрузки каждого исполнителя проекта для каждого этапа работ. Загрузка

исполнителей представлена в процентах, что представляет собой рациональную продолжительность. Этапы работ упорядочены в хронологическом порядке.

Таблица – 4.1 *Перечень работ и продолжительность их выполнения.*

Этапы работы	Исполнитель и	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР, И	НР – 100% И – 50%
Подбор и изучение материалов по тематике	НР, И	НР – 30% И – 100%
Разработка календарного плана	НР, И	НР – 100% И – 50%
Обсуждение литературы	НР, И	НР – 20% И – 100%
Выбор алгоритмов для нейронных сетей	НР, И	НР – 100% И – 80%
Разработка собственного алгоритма под поставленную задачу	НР, И	НР – 70% И – 100%
Получение и обработка данных	И	И – 100%
Оформление расчетно-пояснительной записки	И	И – 100%
Оформление графического материала	И	И – 100%
Подведение итогов	НР, И	НР – 60% И – 100%

Примечание к Таблице 4.1: И – исполнитель; НР – научный руководитель

4.1.1 Продолжительность этапов работ

Существует несколько методов для расчета продолжительности этапов работ, например технико-экономический и опытно-статистический. Первый из них применяется для случаев с хорошо развитой нормативной базой трудоемкости задач. Это обычно связано с их частой повторяемостью. Однако, это почти невозможно в большинстве случаев, и поэтому чаще используется опытно-статистический метод, реализуемый как аналоговый или экспертный.

В аналоговом случае имеем достаточно простой и незатратный способ, возможный лишь тогда, когда в поле досягаемости имеется 39 неустаревающий аналог того, что планируется выполнить в ходе выпускной квалификационной работы. Но в большинстве случаев его глобальное применение невозможно, однако, он отлично применяется локально, для небольших частей работы. Экспертный способ предусматривает привлечение экспертов предметной области для оценки трудоемкости каждого этапа работ. В таблице 4.2 представлена продолжительность каждого этапа работ и занятость каждого исполнителя. Столбцы с 3 по 5 содержат реализацию экспертного подхода.

Таблица – 4.2 Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
Постановка задачи	НР	2	4	2,8	3,36	0	4,05	0
Разработка и утверждение технического задания (ТЗ)	НР, И	3	4	3,4	3,67	0,41	4,42	4,49
Подбор и изучение материалов по тематике	НР, И	9	13	8,6	4,75	15,84	5,75	19,16
Разработка календарного плана	НР, И	3	6	4,2	4,03	1,01	4,86	1,22
Обсуждение литературы	НР, И	3	6	4,2	2,52	2,52	3,04	3,04
Выбор алгоритмов для анализа	НР, И	12	18	12,6	7,6	15,1	9,1	18,2
Разработка собственного алгоритма под поставленную задачу	НР, И	10	14	11,6	9,7	13,9	11,7	16,8

Получение и обработка данных	И	12	16	13,6	0	16,3	0	19,7
Оформление расчетно-пояснительной записки	И	11	15	11,6	0	13,9	0	16,8
Оформление графического материала	И	7	9	7,8	0	9,4	0	11,3
Подведение итогов	НР, И	5	7	5,8	4,2	7	5	8,4
Итого:		77	112	86,2	39,83	95,38	47,92	119,11

Таблица 4.3 *Линейный график работ (пример)*

Этап	НР	С	Март			Апрель			Май			Июнь	
			10	20	30	40	50	60	70	80	90	100	110
1	4,0 5	—											
2	4,4 2	4,4 9											
3	5,7 5	19, 16											
4	4,8 6	1,2 2											
5	3,4	3,2											
6	9,1	18, 2											
7	11, 7	16, 8											
8	—	19, 7											

9	–	16, 8											
10	–	11, 3											
11	5	8,4											

HP – ; И – 

4.2 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- командировочные расходы;
- оплата услуг связи;
- арендная плата за пользование имуществом;
- прочие услуги (сторонних организаций);
- прочие (накладные расходы) расходы.

4.2.1 Расчет затрат на материалы

Затраты на материалы включают в себя стоимость всех материалов, необходимых для выполнения работ, например, стоимость материалов, каких-либо покупных изделий, оборудование, стоимость которого не превышает 40 тысяч рублей. Для определения стоимости каких-либо материалов или оборудования, необходимо посмотреть ее в договоре поставки или в другом сопроводительном документе.

Также в затраты на материалы обычно включаются расходы на различные операции купли-продажи. По-другому эти операции называются транзакциями. Обычно они оцениваются приблизительно как 5-15 процентов от цены 44 оборудования или материала. Расчет затрат на материалы представлен в таблице 4.5.

Таблица – 4.5 *Расчет затрат на материалы*

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма, руб.
Бумага для принтера формата А4	250	1 уп.	250
Картридж для принтера	2500	1 шт.	2500
Канцелярские принадлежности	300	1 шт.	300
Итого:			3050

Допустим, что ТЗР составляют 5 % от отпускной цены материалов, тогда расходы на материалы с учетом ТЗР равны $C_{\text{мат}} = 3050 * 1,05 = 3202,5$ руб.

4.2.2 Расчет заработной платы

Здесь приводится описание расходов на зар. платы научного руководителя и исполнителя. Все расчеты выполняются учитывая трудоемкость выполнения каждого этапа и оклад исполнителя в месяц. Для расчета среднедневной тарифной заработной платы, можно воспользоваться следующей формулой:

$$ЗП_{\text{дн-т}} = МО / 24.83 \quad (4.6)$$

Она учитывает то, что всего в году 298 рабочих дней, а в месяце соответственно $298 / 12 = 24.83$. Здесь рассматривается шестидневная рабочая неделя.

В таблице 4.6 приведены расчеты заработной платы исполнителя (инженера) и научного руководителя. Здесь, из таблицы 4.2.2 взяты временные затраты по каждому исполнителю проекта (данные взяты в рабочих днях путем округления до ближайшего целого). Для того, чтобы учесть в составе заработной платы премии и дополнительные заработные платы, а также районные надбавки, будем использовать следующие коэффициенты: $K_{\text{ПР}} = 1,1$; $K_{\text{доп.ЗП}} = 1,188$; $K_{\text{р}} = 1,3$. Таким образом, для перехода от базовой (тарифной) суммы заработной платы исполнителя, связанной с участием в проекте, к соответствующей зарплатной части сметы (полному заработку) необходимо первую умножить на интегральный коэффициент $K_{\text{и}} = 1,3 * 1,188 * 1,1 = 1,7$. Значение $K_{\text{доп.ЗП}}$,

упомянутое выше, применяется при шестидневной рабочей неделе. В случае пятидневной рабочей недели оно равно 1,113 и в этом случае интегральный коэффициент $K_{\text{и}} = 1,62$.

Таблица – 4.1.2 *Затраты на заработную плату*

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб. день	Затраты времени, раб. дни	Коэффициент	Фонд з/платы, руб.
НР	33 664	1342,09	39	1,699	88457
С	9489	533	95	1,62	81016
Итого:					169473

4.2.3 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН), включающий в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30 % от полной заработной платы по проекту, т.е. $S_{\text{соц.}} = S_{\text{зп}} \cdot 0,3$.
Итак, в нашем случае $S_{\text{соц.}} = 169473 \cdot 0,3 = 50842$ руб.

4.2.4 Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot t_{\text{об}} \cdot \text{Ц}_{\text{э}} \quad (4.7)$$

где $P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт;

$\text{Ц}_{\text{э}}$ – тариф на 1 кВт·час;

$t_{об}$ – время работы оборудования, час.

Для ТПУ ЦЭ = 5,748 руб./кВт·час (с НДС).

Время работы оборудования вычисляется на основе итоговых данных таблицы 5.2 для инженера ($T_{рд}$) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{об} = T_{рд} * K_t, \quad (4.8)$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{рд}$, определяется исполнителем самостоятельно. В ряде случаев возможно определение $t_{об}$ путем прямого учета, особенно при ограниченном использовании соответствующего оборудования.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном.} * K_C \quad (4.9)$$

где $P_{ном.}$ – номинальная мощность оборудования, кВт;

$K_C \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_C = 1$.

Пример расчета затраты на электроэнергию для технологических целей приведен в таблице 4.7.

Таблица – 4.7 Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $\mathcal{E}_{об}$, руб.
Персональный компьютер	$95,38 * 8$ $= 763,04$	0,33	1492,62

Лазерный принтер	10	0,4	20,8
Итого:			1513,42

4.3 Расчет амортизационных расходов

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта.

Используется формула

$$C_{AM} = \frac{H_A * C_{OB} * t_{pf} * n}{F_D}, \quad (4.10)$$

где H_A – годовая норма амортизации единицы оборудования;

C_{OB} – балансовая стоимость единицы оборудования с учетом ТЗР. При невозможности получить соответствующие данные из бухгалтерии она может быть заменена действующей ценой, содержащейся в ценниках, прейскурантах и т.п.;

F_D – действительный годовой фонд времени работы соответствующего оборудования, берется из специальных справочников или фактического режима его использования в текущем календарном году. При этом второй вариант позволяет получить более объективную оценку C_{AM} . Например, для ПК в 2019 г. (298 рабочих дней при шестидневной рабочей неделе) можно принять $F_D = 298 * 8 = 2384$ часа;

t_{pf} – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

n – число задействованных однотипных единиц оборудования.

При использовании нескольких типов оборудования расчет по формуле делается соответствующее число раз, затем результаты суммируются.

Для определения N_A следует обратиться к приложению 1, содержащему фрагменты из постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы». Оно позволяет получить рамочные значения сроков амортизации (полезного использования) оборудования $\equiv C_A$. Например, для ПК это $2 \div 3$ года. Необходимо задать конкретное значение C_A из указанного интервала, например, 2,5 года. Далее определяется N_A как величина обратная C_A , в данном случае это $1 : 2,5 = 0,4$.

Стоимость ПК 45000 руб, время использования 95.38 часа, тогда для него $C_{AM}(ПК) = (0,4 \cdot 45000 \cdot 95.38 \cdot 1) / 2408 = 7129,73$ руб. Стоимость принтера 12000 руб., его $F_d = 500$ час.; $N_A = 0,5$; $t_{pf} = 30$ час., тогда его $C_{AM}(Пр) = (0,5 \cdot 12000 \cdot 30 \cdot 1) / 500 = 360$ руб.

Итого начислено амортизации 7489,73 руб.

4.3.1 Расчет прочих расходов

Для вычисления прочих расходов, необходимо их принять как 10% от суммы всех остальных расходов.

$С_{проч.} = (3202,5 + 169473 + 50842 + 1124,42 + 7489,73) \cdot 0,1 = 23213,16$ руб.

4.3.2 Расчет общей себестоимости разработки

Общая себестоимость проекта по категориям представлена в таблице 4.8.

Таблица 4.8 Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	3202,5
Основная заработная плата	$C_{\text{зп}}$	169473
Отчисления в социальные фонды	$C_{\text{соц}}$	50842
Расходы на электроэнергию	$C_{\text{эл.}}$	1124,42
Амортизационные отчисления	$C_{\text{ам}}$	7489,73
Прочие расходы	$C_{\text{проч}}$	23213,16
Итого:		255344,81

Таким образом, затраты на разработку составили $C = 255344,81$ руб.

4.3.3 Расчет прибыли

Прибыль от реализации проекта в зависимости от конкретной ситуации (масштаб и характер получаемого результата, степень его определенности и коммерциализации, специфика целевого сегмента рынка и т.д.) может определяться различными способами. Если исполнитель работы не располагает данными для применения «сложных» методов, то прибыль следует принять в размере $5 \div 20 \%$ от полной себестоимости проекта. В нашем примере она составляет 50821,2 руб. (20 %) от расходов на разработку проекта.

4.3.4 Расчет НДС

НДС составляет 20% от суммы затрат на разработку и прибыли. В нашем случае это $(254106,55 + 50821,2) * 0,2 = 60985,57$ руб.

4.3.5 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае

$$C_{\text{НИР(КР)}} = 254106,55 + 50821,2 + 60985,57 = 365913,3 \text{ руб.}$$

4.3 Оценка экономической эффективности проекта

Оценка экономической эффективности выходит за рамки представленной работы и предполагает проведение отдельного системного исследования. При условии задание конкретной части применение.

Список литературы

1) Лабораторные работы по курсу «Базы данных». [Электронный ресурс] – Режим доступа: http://orloff.am.tpu.ru/lab_data_base2/index.htm (дата обращения: 05.01. 2019).

2) Все о PHP, MySQL и не только. [Электронный ресурс] – Режим доступа: <http://www.php.su/>

3) Документация по MySQL. [Электронный ресурс] – Режим доступа: <http://www.mysql.ru/docs/>

4) HTML 4: Энциклопедия пользователя / Р. Дарнелл, Д. Бэсори-Киц, Дж. Г. Брайан, Б. Кемпбелл.—Киев: DiaSoft, 1998.—688 с.

5) MySQL. Справочное руководство по MySQL (последние версии на сайте разработчиков) [Электронный ресурс] – Режим доступа: <http://www.mysql.com/documentation>.

6) PHP: настольная книга программиста / Мазуркевич А., Дмитрий, Еловой Д. — Мн.: Новое знание, 2003. — 480 с.

7) Глушаков С.В., Ломотько Д.В. Базы данных. Учебный курс. –М.,2000

8) Коэльо, Л.П. Построение систем машинного обучения на языке Python [Электронный ресурс] / Л.П. Коэльо, В. Ричарт ; пер. с англ. А.А. Слинкин. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 302 с. — Режим доступа: <https://e.lanbook.com/book/82818>. — Загл. с экрана.

8) Рашка, С. Python и машинное обучение: крайне необходимое пособие по новейшей предсказательной аналитике, обязательное для более глубокого понимания методологии машинного обучения [Электронный ресурс] : руководство / С. Рашка ; пер. с англ. А.В. Логунова. — Электрон. дан. — Москва : ДМК Пресс, 2017. — 418 с. — Режим доступа: <https://e.lanbook.com/book/100905>. — Загл. с экрана.

9) Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных [Электронный ресурс] / П. Флах. — Электрон. дан. — Москва : ДМК Пресс, 2015. — 400 с. — Режим доступа: <https://e.lanbook.com/book/69955>. — Загл. с экрана.

10) Шарден, Б. Крупномасштабное машинное обучение вместе с Python [Электронный ресурс] : учебное пособие / Б. Шарден, Л. Массарон, А. Боскетти ; пер. с англ. А. В. Логунова. — Электрон. дан. — Москва : ДМК Пресс, 2018. — 358 с. — Режим доступа: <https://e.lanbook.com/book/105836>. — Загл. с экрана.

11) Компьютерное зрение. Теория алгоритмы. [Электронный ресурс] : учебное пособие / Клетте Р. ; пер. с англ. издательство ДМК. — Электрон. дан. — Москва : ДМК Пресс, 2019. — 506 с. — Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-702-2.pdf> . — Загл. с экрана.

12) Идеи машинного обучения. [Электронный ресурс]: учебное пособие / Шай Шалев-Шварц, Шай Бен-Давид ; пер. с англ. издательство ДМК. — Электрон. дан. — Москва : ДМК Пресс, 2019. — 432 с. — Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-673-5.pdf> . — Загл. с экрана.

13) Искусственный интеллект .NET: речь, язык и поиск. [Электронный ресурс]: учебное пособие / Патак Н.; пер. с англ. издательство ДМК. — Электрон. дан. — Москва : ДМК Пресс, 2019. — 298 с. — Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-605-6.pdf> . — Загл. с экрана.

14) Глубокое обучение с точки зрения практика. [Электронный ресурс]: учебное пособие / Паттерсон Дж., Гибсон А; пер. с англ. издательство ДМК. — Электрон. дан. — Москва: ДМК Пресс, 2019. — 418 с. — Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-481-6.pdf>. — Загл. с экрана.

15) Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

16) Group, Natural Language Computing (2017).R-NET: Machine Reading Comprehension with Self-Matching Networks.

17) Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

18) Трудовой Кодекс Российской Федерации [Электронный ресурс] // КонсультантПлюс. URL:

http://www.consultant.ru/document/cons_doc_LAW_34683/ ,свободный. — Загл. с экрана. — Яз. Рус. Дата обращения: 20.05.2019.

- 19) Назаренко О.Б. Безопасность жизнедеятельности: учебное пособие / О.Б. Назаренко; Национальный исследовательский Томский политехнический университет. – 2-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2010. – 144 с.
- 20) СНиП 23-05-95 "Естественное и искусственное освещение" [Электронный ресурс] // Гарант. URL: <http://base.garant.ru/2306278/>, свободный. – Загл. с экрана. – Яз. Рус. Дата обращения: 14.05.2019.
- 21) ГОСТ 12.1.030-81 Система стандартов безопасности труда. Электробезопасность. Защитное заземление. Зануление [Электронный ресурс]. // Электронный фонд правовой-нормативной документации. URL: <http://docs.cntd.ru/document/5200289>, свободный. – Загл. с экрана. – Яз. Рус. Дата обращения: 18.05.2019.
- 22) ГОСТ 12.1.038-82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов [Электронный ресурс] // Электронный фонд правовой-нормативной документации. URL: <http://docs.cntd.ru/document/5200313>, свободный. – Загл. с экрана. – Яз. Рус. Дата обращения: 18.05.2019.
- 23) ГОСТ Р 50571.3-94. Требования по обеспечению безопасности. Защита от поражения электрическим током [Электронный ресурс] // КонсультантПлюс. URL: <http://www.consultant.ru/cons/cgi/online.cgi?req=doc;base=STR;n=538#0>, свободный. – Загл. с экрана. – Яз. Рус. Дата обращения: 18.05.2019.
- 24) Федеральный закон "Технический регламент о требованиях пожарной безопасности" от 22.07.2008 N 123-ФЗ [Электронный ресурс] // КонсультантПлюс. URL: http://www.consultant.ru/document/cons_doc_LAW_78699/, свободный. – Загл. с экрана. – Яз. Рус. Дата обращения: 22.05.2019.

Приложения

Приложение 1. Построение даталогической модели данных

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] Имя_таблицы  
[(Create-определение,...)]  
[Опции_таблицы]  
[SELECT ... (любое корректное выражение SELECT)]
```

Create-определение может быть одним из вариантов:

Определение поля:

```
Имя_столбца    type    [NOT    NULL    |    NULL]    [DEFAULT  
Значение_по_умолчанию] [AUTO_INCREMENT]
```

Определение первичного ключа:

```
PRIMARY KEY (Имя_столбца,...)
```

Определение индекса (ключа):

```
{INDEX | KEY} [index_name] (Имя_столбца,...)
```

Определение внешнего ключа:

```
[CONSTRAINT имя] FOREIGN KEY [index_name] (Имя_столбца,...)  
[Определение_ссылки]
```

Определение_ссылки:

```
REFERENCES Имя_таблицы [(Имя_столбца,...)]
```

```
[ON DELETE Определение_стратегии]
```

```
[ON UPDATE Определение_стратегии]
```

Определение_стратегии:

```
RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

Опции_таблицы:

```
ENGINE = {InnoDB | MYISAM | ...}
```

и множество других опций.

Приложение 2

```
<?php session_start();
    $id = $_SESSION['id'];
    echo "<br>id_user: <b>", $id, "</b><br>";
    $db = mysql_connect("localhost", "A", "123456");
        mysql_select_db("webbro",$db);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Курсы</title>
    <link rel="stylesheet" href="css/styleMENU.css">

</head>
<body style="background: #30b58a;">
    <div class="wrapper">
        <div id = "pro" class="menuToggle">
            <span id="x"></span>
        </div>
        <ul class="menu clearfix">
            <li id="nav_li"><a href="allkurs.php">Все курсы</a></li>
            <li><a href="mykurs.php">Мои курсы</a></li>
            <li><a href="reiting.php">Рейтинг</a></li>
            <li><a href="profile.php">Профиль</a></li>
            <li><a href="help.html">Служба поддержки</a></li>
            <li class="log_us"> <?php
                $n = mysql_result(mysql_query("SELECT N FROM users WHERE
id_users = $id;"),0);
                echo "Здравствуйте <b>", ($n), "</b>!<br/>"; ?>
            </li>
        </ul>

    </div>

</body>
</html>
```

```

        <?php include 'index2.php'; ?>
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Все курсы</title>
</head>
<body>
    <div align="center">
        <table>
            <br/>
            <tr>
                <th>Название курса</th>
                <th>Описание</th>
                <th>Общие баллы за курс</th>
                <th>Записата на курс</th>
            </tr>
            <?php
                $result = mysql_query("SELECT * FROM `all_kurs`");

                while ($myrow = mysql_fetch_array($result)){
                    echo "<tr>";
                    $id_ks = $myrow['id_kurs'];
                    echo "<td>".$myrow['name']. "</td>";
                    echo "<td>".$myrow['description']. "</td>";
                    echo "<td>".$myrow['all_ball_kurs']. "</td>";
                    echo "<td class='zapisat'><form method='get'><input
type='submit' class='add' name='add$id_ks' value='Записаться' /></form></td>";
                    echo "</tr>";
                    if (isset($_GET["add$id_ks"])) {
                        echo "id_kurs: ", $id_ks, " id_user: ", $id;
                        //Делаем запрос для сохранения данных т.е. проверяем если не
подписан
                        $repat_kurs=mysql_query("SELECT id_kurs FROM my_kurs
WHERE id_user = '$id' AND id_kurs = '$id_ks'; ");
                        $num_rows = mysql_num_rows($repat_kurs);
                        if ($num_rows == 1) {
                            echo "<br>Вы подписаны на курс: ", $myrow['name'];
                        }
                        else {
                            mysql_query("
                                INSERT INTO my_kurs (id_user,id_kurs)
                                VALUES ('$id','$id_ks');
                            ");
                            mysql_close();
                            echo "<br>Вы подписались на курс: ", $myrow['name'];
                        }
                    }
                }
            };

```

```

?>
</table>
</div>
</body>
</html>

```

Приложение 3

```

<?php
    include 'index2.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Мои курсы</title>
</head>
<body>
    <div align="center">
        <table>
            <br/>
            <tr>
                <th>Название курса</th>
                <th>Описание</th>
                <th>Пройти тестирование</th>
            </tr>
            <?php
                ob_start("callback");
                // Вывод таблицы мои курсы конкретного пользователя
                $result = mysql_query("
                    SELECT name, description, my_kurs.id_kurs
                    FROM my_kurs, all_kurs, users
                    WHERE my_kurs.id_kurs = all_kurs.id_kurs
                    AND my_kurs.id_user = users.id_users
                    AND id_users = $id;");

                // print_r(mysql_result($result,0));
                while ($myrow = mysql_fetch_array($result)){
                    echo "<tr>";
                    echo "<td>".$myrow['name']. "</td>";
                    echo "<td>".$myrow['description']. "</td>";
                    $id_ks = $myrow['id_kurs'];
                    echo "
                        <td class='zapisat'><form method='get'><input
type='submit' class='add' name='add$id_ks' value='Посмотреть задачи' /></form></td>";
                    echo "</tr>";
                    // Запрос на показ задачи кданного курса
                    $task_kurs=mysql_query("
                        SELECT id_task, name, ball_task,answers
                        FROM all_kurs, task

```

```

        WHERE task.id_kurs = all_kurs.id_kurs
        AND all_kurs.id_kurs = $id_ks;
    ");
    if (isset($_GET["add$id_ks"])) {
        // для открытие данных т.д.
        echo "
            <tr align= 'center'>
                <td style = 'background: none;'></td>
                <th>Задачи</th>
            </tr>";
        while ($my_task = mysql_fetch_array($task_kurs)){
            $id_tk = $my_task['id_task'];
            echo "
                <tr class='add'>
                    <td style = 'background: #6fcbad;'></td>
                    <td style = 'background: none;'>
                        <form method='get' action='task.php'>
                            <input type='submit' class='add'
name='tk_add' value='$id_tk' />
                            <input type='hidden' class='add'
name='id_ks' value='$id_ks' />
                        </form>
                    </td>
                </tr>";
            }
        }
    };
    ?>
</table>
</div>
</body>
</html>

```

```

<?php include 'index2.php'; ?>
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Все курсы</title>
</head>
<body>
    <div align="center">
        <table>
            <br/>
            <tr>
                <th>id:</th>
                <th>Название курса</th>
                <th>Описание</th>
                <th>Общие баллы за курс</th>
                <th>Записата на курс</th>
            </tr>
        <?php
            $result = mysql_query("SELECT * FROM `all_kurs`");

            while ($myrow = mysql_fetch_array($result)){
                echo "<tr>";
                $id_ks = $myrow['id_kurs'];
                echo "<td>".$myrow['id_kurs'].</td>";
                echo "<td>".$myrow['name'].</td>";
                echo "<td>".$myrow['description'].</td>";
                echo "<td>".$myrow['all_ball_kurs'].</td>";
                echo "<td class='zapisat'><form method='get'><input
type='submit' class='add' name='add$id_ks' value='Записаться' /></form></td>";
                echo "</tr>";
                if (isset($_GET["add$id_ks"])) {
                    echo "id_kurs: ", $id_ks, " id_user: ", $id;
                    //Делаем запрос для сохранение данных т.е. проверяем если
не подписан
                    $repat_kurs=mysql_query("SELECT id_kurs FROM my_kurs
WHERE id_user = '$id' AND id_kurs = '$id_ks'");
                    $num_rows = mysql_num_rows($repat_kurs);
                    if ($num_rows == 1) {
                        echo "<br>Вы подписаны на курс: ", $myrow['name'];
                    }
                    else {
                        mysql_query("
                            INSERT INTO my_kurs (id_user,id_kurs)
                            VALUES ('$id','$id_ks');
                        ");
                        mysql_close();
                        echo "<br>Вы подписались на курс: ",
$myrow['name'];
                    }
                }
            }
        </div>
    </body>
</html>

```



```

    }
    };
    //Добавление курса
    if (isset($_GET['add_new'])) {
        //получаем переменные!
        $id_kurs_new = trim($_GET['id_kurs_new']);
        $name_kurs = trim($_GET['name_kurs']);
        $description_kurs = $_GET['description_kurs'];
        $all_ball = $_GET['all_ball'];
        //Делаем проверку логин если 0 то нечего нету если 1 то
такой пользователь есть!
        $kurs_prov=mysql_query("SELECT id_kurs FROM all_kurs
WHERE id_kurs = '$id_kurs_new' LIMIT 1");
        $kurs_pr = mysql_num_rows($kurs_prov);
        if ($kurs_pr == 1) {
            echo "<div class='error_regis'>Такой курс уже
существует!</div>";
        }
        else {
            //Делаем запрос для сохранение данных т.е
Регистрация курса:!)
            mysql_query("
                INSERT INTO all_kurs (id_kurs, name,
description, all_ball_kurs)
                VALUES
                ('$id_kurs_new','$name_kurs','$description_kurs','$all_ball');
            ");
            echo "<div class='error_regis'>Курс $name_kurs
успешно создан!</div>";
        }
    }
    //Удаление курса gogo
    if (isset($_GET['gogo'])) {
        //получаем переменные!
        $id_kurs_delete = $_GET['id_kurs_delete'];
        mysql_query(" DELETE FROM all_kurs WHERE id_kurs =
'$id_kurs_delete';
        ");
    }
    ?>
    <tr>
        <td><form method="get" action="">
            <input type="text" style="width:30px;"
name="id_kurs_new" placeholder="id"></td>
            <td><input type="text" name="name_kurs"
placeholder="Название курса"></td>
            <td><input type="text" style="width:500px;"
name="description_kurs" placeholder="Описание курса"></td>
            <td><input type="text" style="width:35px;"
name="all_ball" placeholder="баллы"></td>

```

```

                                <td><input      type="submit"      name="add_new"
value="Создать" ></td>
                                </form>
                                </tr>
                                </table>
                                <div>
                                    <h1>Удаление курса!</h1>
                                    <form method="get" action="">
                                        <input      type="text"      name="id_kurs_delete"
placeholder="Введите id курса">
                                        <input type="submit" name="gogo" value="Удалить">
                                    </form>
                                </div>
                            </div>
                            <style type="text/css">
                                .error_regis {
                                    background: rgba(255, 0, 0, 0.5);
                                    max-width: 500px;
                                    border-radius: 15px;
                                    margin: 10px auto;
                                }
                            </style>
                        </body>
                    </html>

```

Приложение 5 скрипт разработки модели

```
import shutil
```

```
from functools import reduce
```

```
from pathlib import Path
```

```
import numpy as np
```

```
import pytest
```

```
import tensorflow as tf
```

```
from deeppavlov.core.layers.tf_layers import cudnn_lstm, cudnn_compatible_lstm, cudnn_gru,
cudnn_compatible_gru
```

```

tests_dir = Path(__file__).parent

tf_layers_data_path = tests_dir / "tf_layers_data"


def setup_module():

    shutil.rmtree(str(tf_layers_data_path), ignore_errors=True)

    tf_layers_data_path.mkdir(parents=True)


def teardown_module():

    shutil.rmtree(str(tf_layers_data_path), ignore_errors=True)


class DPCudnnLSTMMModel:

    def __init__(self, num_layers, num_units):

        sess_config = tf.ConfigProto(allow_soft_placement=True)

        sess_config.gpu_options.allow_growth = True

        self.sess = tf.Session(config=sess_config)

        self.x = tf.placeholder(shape=(None, None, 50), dtype=tf.float32)

```

```

with tf.variable_scope('cudnn_model'):

    h, (h_last, c_last) = cudnn_lstm(self.x, num_units, num_layers, trainable_initial_states=True)


    self.h = h

    self.h_last = h_last


    self.sess.run(tf.global_variables_initializer())


    def __call__(self, x):

        feed_dict = {

            self.x: x,

        }

        return self.sess.run([self.h, self.h_last], feed_dict=feed_dict)


    def save(self, path='model'):

        print('[saving model to {}]'.format(path))

        saver = tf.train.Saver()

        saver.save(self.sess, path)


    def load(self, path):

```

```

saver = tf.train.Saver()

saver.restore(self.sess, path)


class DPLSTMModel:

    def __init__(self, num_layers, num_units):

        sess_config = tf.ConfigProto(allow_soft_placement=True)

        sess_config.gpu_options.allow_growth = True

        self.sess = tf.Session(config=sess_config)


        self.x = tf.placeholder(shape=(None, None, 50), dtype=tf.float32)

        with tf.variable_scope('cudnn_model'):

            h, (h_last, c_last) = cudnn_compatible_lstm(self.x, num_units, num_layers,
            trainable_initial_states=True)


        self.h = h

        self.h_last = h_last


        self.sess.run(tf.global_variables_initializer())


    def __call__(self, x):

```

```

feed_dict = {

self.x: x,

}

return self.sess.run([self.h, self.h_last], feed_dict=feed_dict)

```

```

def save(self, path='model'):

print('[saving model to {}]'.format(path))

saver = tf.train.Saver()

saver.save(self.sess, path)

```

```

def load(self, path):

saver = tf.train.Saver()

saver.restore(self.sess, path)

```

```

class DPCudnnGRUModel:

def __init__(self, num_layers, num_units):

sess_config = tf.ConfigProto(allow_soft_placement=True)

sess_config.gpu_options.allow_growth = True

self.sess = tf.Session(config=sess_config)

```

```

self.x = tf.placeholder(shape=(None, None, 50), dtype=tf.float32)

with tf.variable_scope('cudnn_model'):

    h, h_last = cudnn_gru(self.x, num_units, num_layers, trainable_initial_states=True)


self.h = h

self.h_last = h_last


self.sess.run(tf.global_variables_initializer())


def __call__(self, x):

    feed_dict = {

        self.x: x,

    }

    return self.sess.run([self.h, self.h_last], feed_dict=feed_dict)


def save(self, path='model'):

    print('[saving model to {}]'.format(path))

    saver = tf.train.Saver()

    saver.save(self.sess, path)

```

```

def load(self, path):

    saver = tf.train.Saver()

    saver.restore(self.sess, path)


class DPGRUModel:

    def __init__(self, num_layers, num_units):

        sess_config = tf.ConfigProto(allow_soft_placement=True)

        sess_config.gpu_options.allow_growth = True

        self.sess = tf.Session(config=sess_config)


        self.x = tf.placeholder(shape=(None, None, 50), dtype=tf.float32)

        with tf.variable_scope('cudnn_model'):

            h, h_last = cudnn_compatible_gru(self.x, num_units, num_layers, trainable_initial_states=True)


        self.h = h

        self.h_last = h_last


        self.sess.run(tf.global_variables_initializer())


    def __call__(self, x):

```



```
feed_dict = {  
  
self.x: x,  
  
}  
  
return self.sess.run([self.h, self.h_last], feed_dict=feed_dict)
```

```
def save(self, path='model'):  
  
print('saving model to {}'.format(path))  
  
saver = tf.train.Saver()  
  
saver.save(self.sess, path)
```

```
def load(self, path):  
  
saver = tf.train.Saver()  
  
saver.restore(self.sess, path)
```

```
class TestTFLayers:
```

```
allowed_error_lvl = 0.01
```

```
@staticmethod
```

```

def equal_values(a, b, round=5):

    a, b = np.round(a, round), np.round(b, round)

    return np.sum(a == b) / reduce(lambda x, y: x * y, a.shape)


@pytest.mark.parametrize("num_layers", [1, 3])

def test_cudnn_lstm_save_load(self, num_layers):

    x = np.random.normal(size=(10, 10, 50))

    tf.reset_default_graph()

    cdnnlstmmodel = DPCudnnLSTMMModel(num_layers=num_layers, num_units=100)

    before_load_hidden, before_load_state = cdnnlstmmodel(x)[0], cdnnlstmmodel(x)[1]

    cdnnlstmmodel.save(str(tf.layers_data_path / 'dpcudnnlstmmodel' / 'model'))

    tf.reset_default_graph()

    cdnnlstmmodel = DPCudnnLSTMMModel(num_layers=num_layers, num_units=100)

    cdnnlstmmodel.load(str(tf.layers_data_path / 'dpcudnnlstmmodel' / 'model'))

    after_load_hidden, after_load_state = cdnnlstmmodel(x)[0], cdnnlstmmodel(x)[1]

    equal_hidden = self.equal_values(after_load_hidden, before_load_hidden)

    equal_state = self.equal_values(after_load_state, before_load_state)

    assert equal_hidden > 1 - self.allowed_error_lvl

```

```

assert equal_state > 1 - self.allowed_error_lvl

@pytest.mark.parametrize("num_layers", [1, 3])

def test_cudnn_lstm_save_and_cudnn_compatible_load(self, num_layers):

    x = np.random.normal(size=(10, 10, 50))

    tf.reset_default_graph()

    cdnnlstmmodel = DPCudnnLSTMMModel(num_layers=num_layers, num_units=100)

    before_load_hidden, before_load_state = cdnnlstmmodel(x)[0], cdnnlstmmodel(x)[1]

    cdnnlstmmodel.save(str(tf.layers_data_path / 'dpcudnnlstmmodel' / 'model'))

    tf.reset_default_graph()

    cdnnlstmmodel = DPLSTMMModel(num_layers=num_layers, num_units=100)

    cdnnlstmmodel.load(str(tf.layers_data_path / 'dpcudnnlstmmodel' / 'model'))

    after_load_hidden, after_load_state = cdnnlstmmodel(x)[0], cdnnlstmmodel(x)[1]

    equal_hidden = self.equal_values(after_load_hidden, before_load_hidden)

    equal_state = self.equal_values(after_load_state, before_load_state)

    assert equal_hidden > 1 - self.allowed_error_lvl

    assert equal_state > 1 - self.allowed_error_lvl

```

```

@pytest.mark.parametrize("num_layers", [1, 3])

def test_cudnn_gru_save_load(self, num_layers):

    x = np.random.normal(size=(10, 10, 50))

    tf.reset_default_graph()

    cdnngrumodel = DPCudnnGRUModel(num_layers=num_layers, num_units=100)

    before_load_hidden, before_load_state = cdnngrumodel(x)[0], cdnngrumodel(x)[1]

    cdnngrumodel.save(str(tf.layers_data_path / 'cdnngrumodel' / 'model'))

    tf.reset_default_graph()

    cdnngrumodel = DPCudnnGRUModel(num_layers=num_layers, num_units=100)

    cdnngrumodel.load(str(tf.layers_data_path / 'cdnngrumodel' / 'model'))

    after_load_hidden, after_load_state = cdnngrumodel(x)[0], cdnngrumodel(x)[1]

    equal_hidden = self.equal_values(after_load_hidden, before_load_hidden)

    equal_state = self.equal_values(after_load_state, before_load_state)

    assert equal_hidden > 1 - self.allowed_error_lvl

    assert equal_state > 1 - self.allowed_error_lvl

@pytest.mark.parametrize("num_layers", [1, 3])

```

```

def test_cudnn_gru_save_and_cudnn_compatible_load(self, num_layers):

    x = np.random.normal(size=(10, 10, 50))

    tf.reset_default_graph()

    cdnngrumodel = DPCudnnGRUModel(num_layers=num_layers, num_units=100)

    before_load_hidden, before_load_state = cdnngrumodel(x)[0], cdnngrumodel(x)[1]

    cdnngrumodel.save(str(tf.layers_data_path / 'cdnngrumodel' / 'model'))

    tf.reset_default_graph()

    cdnngrumodel = DPGRUModel(num_layers=num_layers, num_units=100)

    cdnngrumodel.load(str(tf.layers_data_path / 'cdnngrumodel' / 'model'))

    after_load_hidden, after_load_state = cdnngrumodel(x)[0], cdnngrumodel(x)[1]

    equal_hidden = self.equal_values(after_load_hidden, before_load_hidden)

    equal_state = self.equal_values(after_load_state, before_load_state)

    assert equal_hidden > 1 - self.allowed_error_lvl

    assert equal_state > 1 - self.allowed_error_lvl

```