Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»
Институт радиоэлектроники и информационных технологий – РТФ
Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

Директор ШПиАО
_____ Д.В. Денисов
(подпись) (Ф.И.О.)
« 03 » июня 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ПРОЕКТИРОВАНИЕ ЦИФРОВОГО СЕРВИСА ИЗВЛЕЧЕНИЯ ИЗ ТЕКСТОВ ВАКАНСИЙ СТРУКТУРИРОВАННОЙ ИНФОРМАЦИИ О ТРЕБОВАНИЯХ К СОИСКАТЕЛЮ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

Научный руководитель: Коломыцева Анна Олеговна
к.э.н., доцент
_____ подпись

Нормоконтролер: Огуренко Егор Владимирович
_____ подпись

Студент группы: РИМ-220962 Савоськина
Светлана Владимировна
_____ подпись

Екатеринбург
2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
**«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»**

Институт радиоэлектроники и информационных технологий – РТФ
Школа профессионального и академического образования
Направление подготовки 09.04.01 Информатика и вычислительная техника
Образовательная программа 09.04.01/33.03 Инженерия машинного обучения

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студента *Савоськиной Светланы Владимировны* **группы** РИМ-220962
(фамилия, имя, отчество)

1. Тема выпускной квалификационной работы
Проектирование цифрового сервиса извлечения из текстов вакансий структурированной информации о требованиях к соискателю с использованием технологий обработки естественного языка
Утверждена распоряжением по институту от «4» декабря 2023 г. № 33.02-05/298

2. Научный руководитель
Коломыцева Анна Олеговна, к.э.н. , доцент
(Ф.И.О., должность, ученая степень, ученое звание)

3. Исходные данные к работе
Материалы, полученные в ходе учебной и преддипломной практик; техническая и научная литература; наборы данных

4. Перечень демонстрационных материалов
Презентация, веб-приложение

5. Календарный план

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	*1 раздел*	до 23.03.2024 г.	выполнено
2.	*2 раздел*	до 29.04.2024 г.	выполнено
3.	*3 раздел*	до 20.05.2024 г.	выполнено
4.	*ВКР в целом*	до 24.05.2024 г.	выполнено

Научный руководитель *Коломыцева Анна Олеговна* _____ (подпись)
Ф.И.О.

Студент задание принял к исполнению *03.12.2023* _____ (подпись)
дата

6. Консультанты по проекту (работе) с указанием относящихся к ним разделов*

Раздел	Консультант	Подпись, дата	
		задание выдал	задание принял

7. Допустить *Савоськину Светлану Владимировну* к защите выпускной квалификационной работы в экзаменационной комиссии

Директор ШПиАО _____ (подпись) Д.В. Денисов Ф.И.О.

РЕФЕРАТ

Выпускная квалификационная работа магистра 106 стр., 39 рис., 4 табл., 56 источников

ТЕХНОЛОГИИ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА, МЕТОДЫ ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ, СТРУКТУРИЗАЦИЯ ТЕКСТА, ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА, ВАКАНСИИ, НАВЫКИ, РЫНОК ТРУДА, САЙТЫ ТРУДОУСТРОЙСТВА

Цель работы - исследование возможности структуризации текстов вакансий и извлечения из них информации о требуемых навыках методами машинного обучения с использованием технологий обработки естественного языка для создания веб-сервиса, реализующего функцию обработки текстов вакансий и предоставляющего возможность интеграции с ним посредством RESTful API интерфейса.

Объектом исследования являются методы и алгоритмы извлечения структурированной информации из текстов вакансий с использованием технологий обработки естественного языка.

Предметом является разработка веб-сервиса для извлечения структурированной информации о требованиях к соискателю из текстов вакансий.

При проведении исследований использовались методы и модели анализа данных, алгоритмы машинного обучения и технологии обработки естественного языка.

Результаты работы: реализован прототип веб-сервиса для извлечения структурированной информации из текстов вакансий, реализующий функцию обработки текстов вакансий и предоставляющий возможность интеграции с другими веб-сервисами посредством RESTful API интерфейса.

Значимость работы заключается в практической реализации веб-сервиса для извлечения структурированной информации из текстов вакансий.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современных условиях успешное развитие экономики любой страны тесно связано с механизмом функционирования рынка труда, важной частью которого является отрасль услуг по подбору персонала. Наиболее динамично развивающееся направление отрасли – онлайн-рекрутинг, в рамках которого в России и в мире в целом появилось множество онлайн-площадок для поиска работы. Эти площадки представляют собой огромные базы данных, эффективность подбора вакансий в которых во многом зависит от того, насколько качественно в них реализован поиск. Его результаты должны максимально соответствовать навыкам и профессиональным качествам соискателя, а сам поиск не должен требовать от последнего просмотра всей огромной базы данных нерелевантных объявлений.

Существующие на сегодняшний день онлайн-площадки трудоустройства по сути являются хранилищами огромных массивов текстовых документов. Главным способом поиска в них является полнотекстовый поиск. Его основным недостатком является избыточная или, наоборот, нерелевантная выдача, которая может объясняться отсутствием искомой информации в документальной базе данных в том виде, в котором ее ищет пользователь. Иногда это приводит к нахождению ненужных материалов, а иногда – к невыдаче нужных. Другим недостатком существующих систем подбора вакансий является отсутствие возможности удобной сортировки и фильтрации выдачи, что также является следствием применяемых алгоритмов при полнотекстовом поиске.

Очевидным способом решения этих проблем является использование фактографического поиска, который обеспечит не только отсутствие проблемы с нерелевантной выдачей, но и быстрый доступ к необходимой информации, ее эффективную сортировку и фильтрацию, а также автоматизацию рутинных задач по сбору и обработке данных. Для того, чтобы можно было использовать фактографический поиск по вакансиям, их

описания должны быть структурированы и представлены в виде атрибутов, по которым будет осуществляться поиск. И в первую очередь это касается информации о навыках, поскольку она является основной при принятии решения о приеме кандидата на заявленную в вакансии должность.

Актуальность работы заключается в необходимости структуризации текстов вакансий и извлечения из них информации о требуемых навыках для обеспечения возможности применения более эффективных алгоритмов поиска в коллекции документов. Такая задача может быть достаточно эффективно решена с помощью методов машинного обучения и технологий обработки естественного языка, однако большая часть из них реализована в рамках конкретных технологий и языков, не всегда подходящих для создания производительных веб-приложений. Поэтому является весьма перспективным выделение задач, связанных с использованием машинного обучения, в отдельные веб-сервисы, предоставляющие возможности обработки текстовых документов широкому кругу сторонних приложений посредством API-интерфейса.

Исходя из этого, целью данной работы является исследование возможности структуризации текстов вакансий и извлечения из них информации о требуемых навыках методами машинного обучения с использованием технологий обработки естественного языка для создания веб-сервиса, реализующего функцию обработки текстов вакансий и предоставляющего возможность интеграции с ним посредством RESTful API интерфейса.

Для достижения поставленной цели планируется решить такие задачи:

1. Оценить возможности онлайн-рекрутмента для решения актуальных задач рынка труда.

2. Изучить возможности повышения эффективности поиска в документальных базах данных за счет структуризации хранящихся в них документов.

3. Провести анализ существующих информационно-поисковых систем для подбора вакансий.

4. Выполнить сравнительный анализ существующих методов и алгоритмов извлечения структурированной информации из текстов.

5. Предложить и формализовать методы и алгоритмы получения и предварительной обработки текстов вакансий.

6. Разработать методы и модели для идентификации и извлечения основных структурных элементов из текстов вакансий.

7. Разработать методы и модели извлечения требований к навыкам соискателя из описаний вакансий.

8. Выполнить проектирование архитектуры приложения агрегатора вакансий, осуществляющего сбор данных с сайтов трудоустройства, их обработку, хранение и поиск.

9. Реализовать прототип сервиса для извлечения структурированной информации из текстов вакансий и оценить его способность выдерживать нагрузку, необходимую для обеспечения бесперебойной работы агрегатора вакансий.

Объектом исследования являются методы и алгоритмы извлечения структурированной информации из текстов вакансий с использованием технологий обработки естественного языка.

Предметом является разработка веб-сервиса для извлечения структурированной информации о требованиях к соискателю из текстов вакансий.

При проведении исследований использовались методы и модели анализа данных, алгоритмы машинного обучения и технологии обработки естественного языка.

Результатом работы будут архитектура веб-приложения для сбора вакансий с сайтов трудоустройства, их обработки, хранения и поиска, а также прототип веб-сервиса для извлечения структурированной информации из текстов вакансий.

1 Общая характеристика проблемы информационного поиска трудовых вакансий и структуризации их текстового описания

1.1 Рынок труда и онлайн-рекрутмент в России

В современных условиях успешное развитие экономики любой страны тесно связано с механизмом функционирования рынка труда. Определенная часть населения по тем или иным причинам меняет место работы, другая часть впервые приступает к трудовой деятельности. Создание условий, которые бы обеспечивали возможности для рационального размещения трудовых ресурсов, во многом обуславливает достижение устойчивого экономического роста и повышение благосостояния граждан.

Инфраструктура рынка труда представлена совокупностью государственных и негосударственных организаций по содействию в трудоустройстве, которые обеспечивают необходимое взаимодействие между спросом и предложением на рынке труда. В России управлением занятостью населения загимаются как государственные органы службы занятости, так и различные негосударственные службы занятости, называемые также «операторами рынка труда»: кадровые агентства, рекрутинговые компании, агентства по подбору персонала [1]. За последнее время их деятельность превратилась в бурно развивающуюся отрасль экономики. Объясняется это тем, что всегда существует потребность в квалифицированной рабочей силе.

В современном обществе наблюдается постоянный дефицит квалифицированных специалистов. Компании заинтересованы в привлечении работников наилучшей квалификации. В результате исследования [2] было выявлено, что главным требованием работодателей к работникам является высокий уровень квалификации специалиста (83,3% опрошенных). При этом собственные службы персонала предприятий не обладают достаточными возможностями для поиска кандидата необходимой квалификации. Для этого работодатели готовы обращаться в частные кадровые агентства и оплачивать

оказываемые ими услуги по поиску и подбору персонала. Новейшие технологии позволяют кадровым агентствам в максимально короткие сроки провести анализ сложившейся ситуации, как для отдельно взятого предприятия, так и для рынка труда в целом, и обеспечить быстрый подбор квалифицированных кандидатов на вакантные должности.

В настоящее время в России уже сформировалась отрасль услуг по подбору персонала. Деятельность, связанную с заполнением вакантных рабочих мест у сторонней организации (работодателя) кандидатами, максимально подходящими требованиям заказчика, часто называют «рекрутментом» или «рекрутингом».

Возможны два источника для поиска и подбора кандидатов – внутренний и внешний. К внутренним источникам относятся прямой поиск внутри организации (для службы занятости – в ее базе данных кандидатов, для нанимающей организации – из числа ее действующих или уволенных сотрудников). К внешним источникам относятся учебные заведения, другие службы занятости и публикация рекламных объявлений в прессе и специализированных источниках (СМИ и интернет) [1]. В последнем случае информация о вакансии может размещаться на собственном сайте агентства или на сторонних сайтах, ориентированных на потенциальных кандидатов. В работе [1] показано, что привлечение кандидатов извне организации, осуществляющей найм, увеличивает возможности выбора кандидатов и отбора наиболее квалифицированных из них, а также позволяет использовать предыдущий опыт, накопленный сторонними кандидатами в аналогичных организациях.

Однако внешнее привлечение кандидатов предполагает их самостоятельность и индивидуальную активность. Обращающиеся в организацию посторонние претенденты должны самостоятельно найти вакансию предприятия и прислать по почте или Интернету резюме и рекомендации. Залогом успешного функционирования такого метода привлечения является наличие условий для самостоятельного поиска работы

кандидатами и возможности качественно презентовать потенциальному работодателю свои профессиональные качества.

Поиск работы соискателями может происходить с помощью различных подходов. В работе [3] показано, что у кандидата для поиска вакансий есть несколько основных каналов: обращение в органы государственной службы занятости и частные агентства; помощь знакомых и родственников; просмотр объявлений о вакансиях и размещение объявлений о поиске работы в средствах массовой информации (в том числе в сети Интернет); непосредственное обращение к работодателю.

В настоящее время наиболее эффективный канал, обеспечивающий наибольший охват аудитории соискателей и работодателей – это размещение объявлений с описанием вакансий на специализированных ресурсах в сети интернет. Не так давно появилось понятие «онлайн-рекрутмент» (от англ. Online – на линии, на связи и англ. Recruitment – вербовка) – метод поиска работников с помощью интернет-ресурсов, таких как [3]:

- сайты трудоустройства,
- сайты типа «доска объявлений» и сайты-агрегаторы;
- профессиональные сайты;
- социальные сети.

Основная задача сайтов по трудоустройству – предоставить соискателю вакансии по разным специальностям и размещение его резюме. Считается, что этот метод поиска работы неэффективен для специалистов низкой квалификации и персонала высшего звена (топ-менеджеров): первые не ищут работу с помощью интернет-ресурсов, а вторые не используют подобный метод поиска работы. Сайты по трудоустройству могут быть узко специализированы. На таких сайтах размещается информация о вакансиях и кандидатах из какой-то одной сферы деятельности, например, только для информационных технологий [3].

Современные сайты трудоустройства, как правило, предоставляют весь спектр услуг профессиональных рекрутинговых агентств. При этом они часто

используются частными кадровыми агентствами для размещения своих объявлений. Основной вид бизнеса для таких сайтов – услуги по подбору персонала, т. е. они сами по сути являются кадровыми агентствами, и организации могут напрямую через них, не обращаясь в сторонние организации, осуществлять подбор персонала.

Сайты-доски объявлений являются упрощенным аналогом сайтов по трудоустройству с точки зрения размещения объявлений с описанием вакансий и резюме. Однако такие сайты не предоставляют дополнительные возможности поиска и подбора вакансий и кандидатов. На таких сайтах размещаются обычные списки объявлений с простыми функциями сортировки и фильтрации. Размещение объявлений может быть бесплатным или платным, но во втором случае стоимость, как правило, значительно меньше, чем на сайтах трудоустройства. Этот канал преимущественно используется для поиска низкоквалифицированной рабочей силы, а также организациями, у которых нет возможности выделить значительный бюджет на подбор персонала.

Кроме них, также существуют так называемые «сайты-агрегаторы», которые собирают объявления из различных источников и предоставляют информацию в более удобном для поиска виде. Они, как правило, не предоставляют услуг по рекрутингу, не имеют возможности добавлять свои объявления, их основная задача – информирование потенциальным кандидатов о вакансиях на других сайтах. Они предоставляют гораздо больший объем объявлений, сокращая таким образом время на подбор подходящих вакансий.

Профессиональные тематические сайты можно условно разделить на профессиональные тематические сайты и профессиональные форумы [4].

Профессиональные тематические сайты, так же как и социальные сети, позволяют квалифицированным специалистам заявить о себе, выступая в роли эксперта, комментатора или автора статей. Дополнительно профессиональные сайты могут включать разделы по трудоустройству и

форумы специалистов. Недостатком использования данного метода являются значительные затраты времени и усилий для достижения искомого результата.

Профессиональные форумы используются для общения и обмена опытом специалистов в какой-либо области. Особенностью их использования является возможность обратиться напрямую к профессионалам [4]. Недостатки такие же, как и в предыдущем случае – необходимость выполнения множества действий для поиска нужной вакансии и значительные затраты времени.

Из всех перечисленных наиболее удобным каналом для соискателей остаются сайты трудоустройства и электронные доски объявлений. Их неоспоримые преимущества: сайты обладают большим охватом аудитории; отражают актуальное состояние рынка труда; позволяют и кандидатам, и работодателям оценить свою конкурентоспособность. Крупные сайты трудоустройства, кроме прочего, предоставляют различные аналитические отчеты о развитии и изменениях рынка труда.

Все большее число работодателей используют интернет для размещения своих вакансий. А для большинства ИТ-компаний это основной метод поиска квалифицированных специалистов. Это связано с тем, что IT-сфера в настоящее время крайне динамично развивается. Для нее характерно ежегодное обновление технологий и инструментов, которыми должны владеть специалисты. В настоящее время спрос на квалифицированных специалистов в IT-сфере значительно превышает предложение из-за бурного роста потребности в ИТ-специалистах в различных областях, где происходят процессы автоматизации и цифровизации. Согласно ежегодным отчетам портала hh.ru, посвященным вакансиям в различных сферах, одной из наиболее популярных сфер на текущий момент являются информационные технологии [5].

В данный момент ИТ-направление для Российской Федерации – одно из самых важных, но при этом отмечается значительный кадровый дефицит

[6]. На начало 2013 года в России в сфере ИТ было задействовано около 300 000 специалистов высокой квалификацией [7]. А к 2018 году это число выросло уже до 700 000, практически в 2,5 раза, но работников по-прежнему не хватало. В США, например, планируется нанять как минимум 1 000 000 новых работников в сфере ИТ, в России – 400 000, а в европейских странах – 300 000.

Рекрутинговое агентство GMS провело опрос 100 рекрутеров в сфере ИТ, и оказалось, что на поиск IT-специалистов по всей России уходит до 30-40% их рабочего времени. Это связано с очень высокими требованиями IT-компаний к компетенциям своих будущих работников, из-за чего поиски кандидатов затягиваются. Несоответствие компетенций приводит к большому количеству отказов соискателям, найденным рекрутинговыми компаниями [4].

В августе 2023 года нехватка кадров в IT-отрасли в России оценивалась в 500–700 тыс. работников. Об этом в тот период сообщил глава Минцифры Максут Шадаев. На сегодняшний день дефицит специалистов в IT-области оценивается в цифру более 1 млн человек. И ближайшие годы этот дефицит покрыт не будет [8]. Это обусловлено тем, что компетенции, которые ожидают работодатели от кандидатов, зачастую не соответствуют компетенциям, которыми они обладают. Такое несоответствие навыков особенно свойственно наиболее интенсивно развивающимся областям, в которых, в том числе, активно используют информационные технологии. Дефицит квалифицированных сотрудников сегодня существует также и во многих других отраслях: производство, продажи, транспорт и строительство.

Поэтому современному работнику для того, чтобы быть востребованным на рынке труда, требуется наличие достаточно высокой квалификации. Профессионализм кандидата важен не только для принятия решения о найме, но и для дальнейшей трудовой деятельности кандидата. При этом эксперты сходятся во мнении, что для нахождения стабильной высокооплачиваемой работы IT-специалисту необходимо регулярно

совершенствовать свои навыки и изучать новые направления [8]. В случае отсутствия требуемых на рынке труда компетенций, соискатель должен приобретать их посредством самообразования, краткосрочных программ дополнительного образования, курсов повышения квалификации. Поэтому является актуальным наличие таких инструментов, которые бы с одной стороны, позволили соискателю иметь актуальную информацию о требованиях рынка труда к знаниям и навыкам в определенной сфере деятельности, а с другой стороны – оценить востребованность своих собственных знаний и умений в выбранной профессии.

Одной из проблем, которые приходиться решать соискателю при поиске работы, является выбор подходящей специализации в рамках его профессии. У него могут возникнуть сложности при выборе той или иной специализации из-за того, например, что не все требуемые для заявляемой позиции знания и умения явно перечислены в описании вакансии. С другой стороны, вакансии на одну и ту же должность, например, «бекенд разработчик», могут ожидать от соискателя совершенно разных навыков. Для решения этой проблемы специалист должен ясно понимать, какие именно навыки и знания необходимы на рынке труда для той или иной специализации.

1.2 Введение в проблему информационного поиска

Основой большинства сайтов трудоустройства является достаточно большая база описаний вакансий и резюме. Для поиска подходящих вакансий необходимо проанализировать большой объем существующей на текущий момент информации. Соискатель, как правило, стремится подобрать вакансию наиболее близкую к имеющимся у него навыкам и знаниям, однако, из-за того, что значительная часть объявлений существует в виде неструктурированных текстов, это требует высоких трудозатрат и времени.

Поиск (search) – это совокупность операций, которые связаны с определением местонахождения в произвольном хранилище информации одного или нескольких элементов, обладающих заданным свойством [9].

Предметом поиска является информационная потребность пользователя – это состояние отдельного лица или системы, которое характеризуется необходимостью получения информации [9]. Как правило, это некоторый запрос, состоящий из набора ключевых слов из искомой области.

Объект запроса – это единица информации, которая хранится в базе системы поиска. Объектом поиска могут быть структурированные, неструктурированные и слабоструктурированные данные [9].

Структурированные данные – это данные, имеющие определенный формат или определенную модель данных, например, в форме таблицы со строками и столбцами, четко определяющими атрибуты данных. Такие данные можно эффективно обрабатывать для получения аналитики благодаря их количественному характеру. Они обычно расположены в хранилище, например в реляционной базе данных или просто в файле .csv, и их можно легко извлечь и прочитать с помощью SQL.

Под термином «неструктурированные данные» обычно подразумеваются данные, которые не имеют ясной и легко реализуемой на компьютере структуры. Неструктурированные данные представлены в абсолютно необработанной форме. Это наиболее распространенный вид данных, к нему относятся изображения, видео, аудио и текст на естественном языке. Эти данные сложно обрабатывать из-за их сложной организации и форматирования. Несмотря на отсутствие структуры, неструктурированные данные могут содержать важную информацию, которую можно извлечь и преобразовать в структурированные или слабоструктурированные данные с помощью, например, методов обработки естественного языка (NLP, natural languageprocessing).

В случаях, когда структура информации не соответствует строгим требованиям форматирования, используется формат слабоструктурированных данных, объединяющий данные различной структуры в одном контейнере (таблице базы данных или документе). Как и неструктурированные данные, слабоструктурированные данные не привязаны к заданной схеме. Однако

они, в отличие от неструктурированных данных, имеют определенную структуру, которая обычно выражена в виде тегов или других маркеров. Например, текстовые данные имеют неявную структуру, характерную для естественных языков: в них есть заголовки, абзацы и сноски, которые обычно оформлены в тексте в виде какой-либо разметки (например, HTML) [10].

В зависимости от типа данных используют либо поиск данных, либо информационный поиск.

Поиск данных – процесс аналитического исследования больших массивов информации с целью выявления определенных закономерностей и взаимосвязей между переменными. Поиск данных происходит в массивах со структурированной информацией, и его результат всегда детерминированный [9]. Поскольку структурированные данные хранятся в реляционных СУБД, то и поиск таких данных осуществляется средствами этих СУБД с помощью специального языка запросов SQL.

Информационный поиск (IR) – это процесс поиска в большой коллекции некоего неструктурированного и слабоструктурированного материала (обычно – документов) [10]. В отличие от поиска данных, содержанием информационного поиска является нахождение объектов: фактов, информации в документах, самих документов, метаданных из документов, текста, изображений, содержащихся в хранилище неструктурированной информации. Как критерии информационного поиска, так и его результаты являются недетерминированными, т.е. неопределенными. Этими признаками информационный поиск отличается от поиска данных. Поэтому важным критерием качества информационного поиска является релевантность – степень соответствия результатов поиска запросу пользователя.

Для выполнения поиска используется поисковая система (ПС) – совокупность программных и лингвистических средств, которые, кроме собственно поиска, выполняют предварительный анализ информации, формирование запроса, представление результатов поиска.

По способу организации информационного массива среди поисковых систем выделяют документальные, фактографические и документально-фактографические (смешанные) [11].

Документальные системы служат для работы с данными, в которых единицей информации и объектом поиска является текстовый документ (document) или графический объект, соответствующий полученному запросу. Они снабжены тем или иным аппаратом поиска. Группу документов, по которой осуществляется поиск обычно называют коллекцией (collection) или корпусом (corpus) или массивом текстов (body of texts) [10].

Наиболее распространённым объектом запроса является текстовые документы, которые представляют собой неструктурированные и слабоструктурированные данные в виде текста на естественном языке, характерной особенностью которых является нечеткость. Сознание человека способно воспринимать нечеткие суждения и из контекста делать выводы о значениях. Машина может воспринимать только то, что явно задано в описании модели автоматической обработки текста. Многозначность языковых единиц значительно снижает качество работы систем автоматической обработки текстов, так как ставит проблему выбора из множества альтернатив, что не доступно «пониманию» машины. Все это делает задачу поиска информации в массивах неструктурированных данных на естественном языке сложной и нетривиальной.

При документальном поиске пользователь сам извлекает из документа факты и данные. Относительно объекта запроса документальный поиск может иметь следующие цели [10]:

1) поиск сведений об объекте и установление его наличия в системе других объектов;

2) поиск самих объектов (документов), в которых есть или может содержаться нужная информация;

3) поиск фактических сведений, которые содержатся в информационных объектах.

В документальных ПС используют следующие методы поиска [9]:

– адресный поиск (формально-механический) – процесс поиска документов по чисто формальным признакам, указанным в запросе; для этого нужны точный адрес документа (например, адрес веб-страницы) и обеспечение определенного порядка расположения документов в хранилище;

– семантический поиск (тематический) – процесс поиска документов по их содержанию; для этого содержание документов и запроса должно быть переведено с естественного языка на информационно-поисковый язык и получены поисковые образы документа и запроса.

Основное отличие адресного и семантического поиска заключается в том, что при адресном поиске документ рассматривается как единый объект, а при семантическом поиске – как информация. Из методов семантического поиска чаще используются следующие [9]:

– полнотекстовый поиск – поиск по всему содержимому документа;

– поиск по атрибутам или метаданным документа - свойствам, характеризующим каким-либо образом документ: названию, дате создания, автору и т. п.

Пример поиска по метаданным – поиск файла по его названию или расширению.

Пример полнотекстового поиска – любая поисковая система в интернете. Как правило, для ускорения полнотекстового поиска применяют различные индексы, например, инвертированные индексы. Найденные в результате документы признаются системой формально релевантными.

Релевантность – это соответствие результатов поиска отправленному запросу. Формальная релевантность определяется используемыми математическими моделями в конкретной информационно-поисковой системе. Фактическая релевантность устанавливается человеком в процессе сопоставления документа и запроса [10].

Среди недостатков документальной системы можно выделить два:

– способность выдавать ненужные пользователю документы;

– способность не выдавать нужные документы (например, если автор употребил какой-то синоним или ошибся в написании).

Документальные поисковые системы на запрос пользователя могут выдать очень большой список документов, в котором, как правило, наблюдается огромная избыточность, что, в свою очередь, не позволяет такой поиск охарактеризовать как эффективный. Поэтому основной характеристикой таких систем является релевантность.

Другой характеристикой документальных информационно-поисковых систем (ИПС) является «пертинентность» – соответствие найденных информационно-поисковой системой документов информационным потребностям пользователя [10]. А они, как правило, характеризуются неопределенностью: когда пользователю не хватает имеющихся знаний, ему бывает крайне тяжело точно сформулировать запрос для поиска.

Оценка результатов работы системы пользователем субъективна по своей природе и зависит от его ожиданий. Поэтому на практике для оценки качества работы ИПС используются два критерия – релевантность и пертинентность, из которых выполнение требований первого считается обязательным, а второго – желательным. Для пользователя же наиболее важной оценкой является пертинентность полученной информации. Нерелевантные выдачи могут объясняться отсутствием искомой информации в документальном потоке в том виде, в котором ее ищет пользователь. Иногда это приводит к нахождению ненужных материалов, в других случаях – к выдаче «неожиданно полезных», т.е. сначала не запрашиваемых, но ценных с точки зрения пользователя документов или фактических данных.

Вопросу повышения релевантности выдачи в полнотекстовых ИПС посвящено множество исследований ([12] ,[13] , [11] , [14] , [15] , [16] , [17] и др.), однако универсального решения, одинаково хорошо работающего в различных ситуациях, все еще не найдено. Одним из подходов, например, является усовершенствование моделей извлечения информации, учитывающие особенности естественного языка [11].

Неоспоримым преимуществом документальных систем является относительная дешевизна и простота их создания: не требуется предварительно выявлять структуру данных, проектировать схему хранилища, документы помещаются в базу как есть, без предварительной обработки, при этом не предъявляется особых требований к типу и формату этих документов. Простота создания хранилища оборачивается сложностью методов и алгоритмов, используемых в информационно-поисковой системе для работы с этими документами. Среди недостатков таких систем также выделяют:

– задача формулировки правильного (с точки зрения ИПС) поискового запроса часто оказывается сложной, если пользователь не является экспертом в области поиска, в результате пертинентность поиска часто оказывается весьма низкой;

– избыточность, т. е. нахождение ненужных материалов, так называемый «шум» информационного поиска, что приводит к необходимости оценивать результаты поиска с помощью критерия релевантности;

– необходимо активное участие пользователя в процессе поиска;

– у большинства программных систем поиска присутствуют лишь простейшие средства навигации в найденном множестве документов, позволяющие только перебирать документы [18].

Фактографические информационно-поисковые системы лишены главного недостатка документальных ИПС – избыточности и нерелевантности поисковой выдачи. В них единицей информации выступает факт, событие или данные, которые могут быть описаны конкретными значениями или свойствами, а также сведения, полученные из документов.

Фактографические информационные системы (ФИС) хранят структурированные данные, имеющие четкую структуру, позволяющую машине отличать одно данное от другого. В основе работы ФИС лежит использование баз данных для хранения и организации фактографической информации, а также применение различных алгоритмов и методов анализа

данных для обработки и анализа фактов [9].

В фактографических ИС объектами поиска являются факты – конкретные значения данных (атрибутов) об объектах реального мира в каком-то заранее обусловленном формате. Фактографический поиск представляет собой поиск фактов, непосредственно отвечающих на запрос. Поэтому фактографическая система способна давать однозначные ответы на поставленные вопросы.

Важным элементом ФИС являются системы управления базами данных (СУБД). Современные СУБД оперируют огромными массивами информации, объемы которых достигают десятков терабайт. Выполняя запрос пользователя они должны обеспечит время отклика порядка нескольких секунд. Для этого во всех СУБД организован метод ускоренного доступа к данным с помощью таких высокоэффективных методов организации прямого доступа, как индексирование и хэширование.

Для получения информации из БД пользователи направляют СУБД запросы. СУБД отвечает за их обработку и формирование результата поиска. Для запросов используется специальный язык запросов. Фактическим стандартом такого языка для современных реляционных СУБД стал SQL (Structured Query Language – структурный язык запросов).

Фактографические информационные системы имеют определенные преимущества перед документальными:

– быстрый доступ к необходимой информации, эффективная сортировка и фильтрация поисковой выдачи;

– автоматизация рутинных задач по сбору и обработке фактов;

– отсутствие проблемы с нерелевантной выдачей, результаты поиска всегда точны и соответствуют поисковому запросу.

Есть и недостатки ФИС:

– сложность внедрения и настройки системы: информация перед помещением ее в базу данных должна быть структурирована;

– необходимость наличия у пользователей соответствующих навыков по работе с системой, например, умения составлять запросы на специальном языке (например, SQL);

– возможность ошибок при вводе и обработке фактов.

Основное отличие документальных и фактографических поисковых систем с точки зрения пользователя состоит в том, что в документальных системах чаще всего используется полнотекстовый поиск, который позволяет вводить строку запроса на естественном языке, и пользователю не требуется дополнительных знаний для работы с ней. В ФПС требуется составлять запрос с помощью специального языка запросов, на изучение которого может потребоваться какое-то время. Цена за такое удобство – качество поиска: в целом, все методы поиска в документальных ИПС дают избыточную и не релевантную выдачу. Как способ компенсации предлагается ранжирование выдачи по релевантности, что немного смягчает ситуацию, но не решает проблему. Лучший выход в этом случае – структуризация информации, содержащейся в документах, и переход на фактографические системы.

Возможно также компромиссное решение – использование смешанных систем, в которых описание каждого факта соотносится с документами, в которых имеется информация о нем.

1.3 Обзор существующих ИПС для подбора вакансий

Объявление о вакансии представляет собой обычный документ, содержащий неструктурированную тестовую информацию. Некоторые ИПС предлагают возможности для частичной структуризации информации в объявлении и снабжают представляющий его документ дополнительными атрибутами (или метаданными). Поэтому ИПС для подбора специалистов чаще являются смешанными документально-фактографическими системами, в которых присутствуют оба вида поиска.

Основная задача потенциального кандидата – найти такие объявления, в которых перечень требуемых навыков и профессиональных качеств

соответствовал бы имеющимся у него навыкам и качествам. Отсутствие хотя бы одного навыка из требуемых делает вакансию неподходящей. Поэтому очевидно, что поиск в базе вакансий должен осуществляться в первую очередь по критерию наличия всех требуемых навыков у соискателя.

Другая задача, которая стоит перед соискателем, – составить свое резюме таким образом, чтобы его нашли рекрутеры среди тысяч других, таких же резюме. Существующие подходы к поиску кандидатов приводят к тому, что рекрутеры выбирают не лучшего специалиста, а того, кто лучше всех других подходит под их критерии. Поисковые системы ищут высокое совпадение ключевых слов в резюме, обычно от 70% до 80%, между тем, что нужно работодателю в вакансии, и навыками, указанными в чьем-то резюме. Если робот не найдет эти ключевые слова, претендент не пройдет в следующий раунд, даже если во всём остальном он идеально подходит на роль [19].

Для успешного прохождения фильтров поисковых систем, резюме необходимо оптимизировать для поисковых систем с ИИ, указав релевантные специальности ключевые слова и навыки. Найти их можно в вакансиях для этой специальности. А для этого важно найти список вакансий, которые наиболее точно соответствуют качествам соискателя.

Далее рассмотрены несколько платформ для трудоустройства с точки зрения возможности поиска вакансий по требуемому набору навыков и знаний кандидата.

Наиболее крупными и известными сайтами онлайн-рекрутмента в России являются «HH.ru», «SuperJob.ru», «Rabota.ru» [20]. Среди тематических сайтов в сфере информационных технологий можно выделить такие, как «Хабр Карьера» [21], «Geekjob» [22], «Типичный программист» [23]. Среди международных агрегаторов наиболее крупные «LinkydIn», «Monster.com».

Ресурс «HeadHunter.ru» («HH.ru») – крупнейшая онлайн-рекрутинг платформа в России, лидер в разработке HR-tech решений и сервисов,

занимает лидирующие позиции в сфере поиска работы. По состоянию на конец февраля 2024 года в системе онлайн-рекрутмента hh.ru опубликовано более 1,4 миллиона объявлений о вакансиях, в том числе более 85000 – для IT-сферы [24].

«HeadHunter» (hh.ru) был создан в 2000 году, правда, первые три года носил другое имя. Изначально он ориентировался на поиск работы только для высококвалифицированных сотрудников. Однако потом политика сайта изменилась и его аудитория возросла. На площадке появилось много вакансий для студентов и неквалифицированных специалистов. Вакансии предлагаются в разных сферах деятельности по всей России. В настоящее время на сайте представлено 68 млн. резюме и около 1,5 млн вакансий [24].

Сервис бесплатен для соискателей. Они могут создать резюме во встроенном редакторе и настроить параметры его отображения. Для работодателей предлагается бесплатное размещение трех вакансий. Дополнительные объявления, доступ к контактным данным претендентов, а также использование инструментов продвижения предоставляются платно.

На сайте используются передовые технологии для того, чтобы работодатели могли быстро найти подходящего сотрудника, а соискатели – хорошую работу. Заявлено, что поиск на сайте использует искусственный интеллект, и сайт обрабатывает до 3000 запросов в секунду [24].

Платформа «HH.ru» представляет собой классическую документальную информационно-поисковую систему. Единицей поиска здесь служит объявление с описанием вакансии или резюме, а основным методом поиска является полнотекстовый. Для оптимизации работы сайта разработчики платформы используют самые современные методы и технологии, такие как: для поиска и рекомендаций – обработка естественного языка с помощью нейросетей-трансформеров, для рекомендательной системы – методы машинного обучения. Рекомендательная система, используя данные из резюме соискателя, подбирает соискателям подходящие вакансии .

Упрощённое представление рекомендательно-поисковой системы «HH.ru» показано на рисунке 1.1 [25].

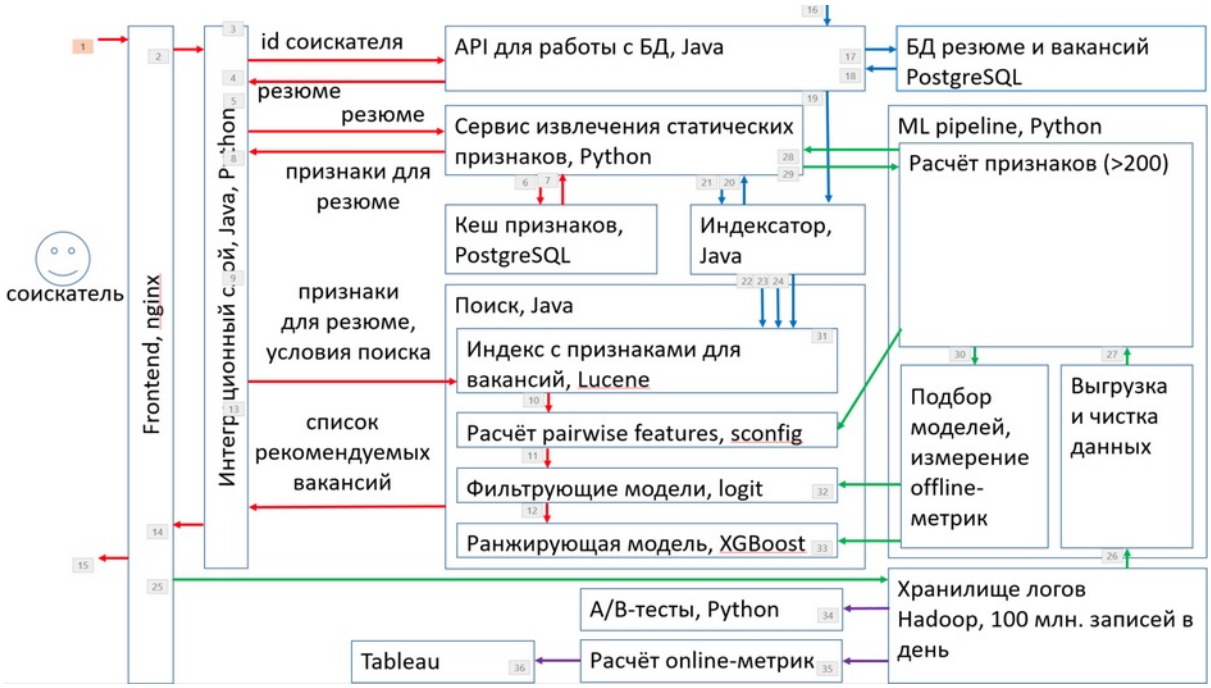


Рисунок 1.1 – Рекомендательно-поисковая система «HH.ru»

Поиск осуществляется по ключевым словам из поискового запроса с использованием фильтров и с последующим ранжированием от рекомендательной системы (рисунок 1.2) [25].

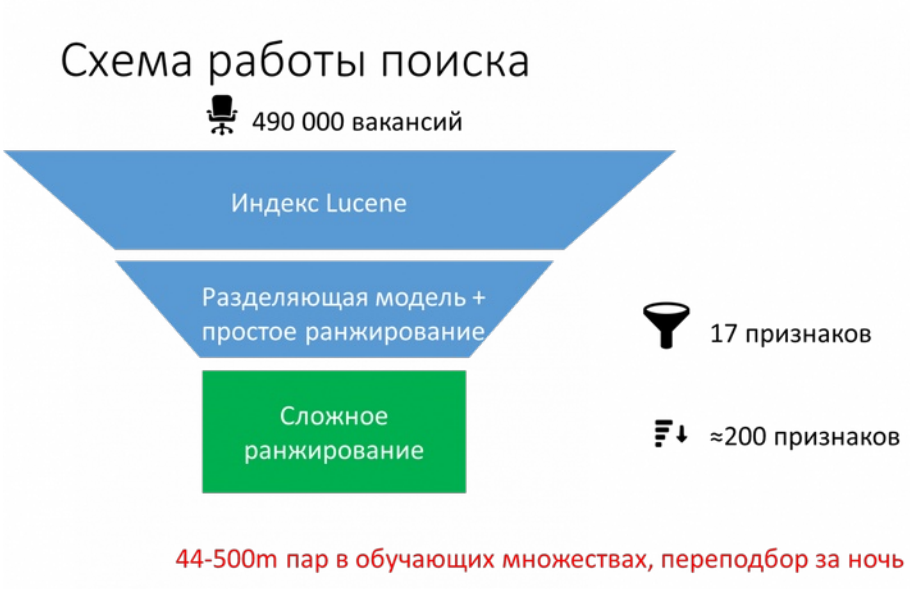


Рисунок 1.2 – Схема работы поиска «HH.ru»

В поиске используются признаки из рекомендательной системы: статические (вычисляемые до выполнения запроса), текстовые, числовые и категориальные, а также динамические, которые считаются при обработке запроса. Для построения системы ранжирования используются логи пользователей. Для повышения эффективности поиска используются такие инструменты как подсказки, синонимы, опечаточник для исправления ошибок, саджест для подсказки более подходящего запроса в строке поиска [25].

Помимо полнотекстового поиска, платформа предлагает фильтрацию по таким атрибутам, как: уровень дохода, регион, специализация, отрасль компании, образование, тип занятости, график работы. Однако возможности сортировки весьма ограничены: только по релевантности (для полнотекстового поиска) и по метаданным объявления (дате и зарплате).

Необходимо отметить, что несмотря на активное использование методов машинного обучения и продвинутых помощников, в целом качество поисковой выдачи, предоставляемой поисковым роботом «HH.ru», весьма посредственное: много дублирующихся и нерелевантных объявлений, нет возможности поиска отдельно по навыкам.

Однако команда «HH.ru» активно работает в этом направлении. Так, в статье [26] указывается , что «Skill-based подход» – это общемировая тенденция, и сервис «HH.ru» «оказался одним из тех крупных игроков рынка, кто разглядел в нем большое будущее для своих продуктов и решил внедрить его на своей платформе». Этот подход подразумевает подбор персонала с опорой на навыки, то есть на то, что кандидат действительно знает и умеет. «Это конкретные требования к соискателям, выверенный список профессиональных компетенций и личных качеств, которыми должен обладать представитель каждой специальности, – от софт– до хард-скилов». Философия «skill-based» уже реализована в продуктах и сервисах «LinkedIn», «Indeed», «SEEK». Менеджмент «HH.ru» 2−3 года назад включил этот подход в обновлённую стратегию сервиса, посчитав это важным.

Внедрение навыков в систему поиска «HH.ru» основано на понятии «модели компетенций» – «это формализованное описание профессии, отвечающее на вопрос «Что должен уметь профессионал в рамках этой профессии?». Модель компетенций собирается в рамках конкретной компании. Однако разработчики указывают на сложность в ее реализации, связанную с тем, что в одной вакансии зачастую может быть сразу несколько профессий. В объявлениях с одинаковым названием работодатели часто подразумевают сотрудников с принципиально разным набором навыков. К примеру, объявления с «дизайнером» в заголовке могут содержать абсолютно отличающиеся друг от друга ожидания от компетенций кандидата, ведь категорий и профилей в этой профессии очень много [26].

Для разработанной системы была создана упорядоченная и согласованная таксономия, которая легла в основу карты компетенций для более чем 150 профессиональных ролей на российском рынке труда. Для таксономии разработчики решили использовать имеющийся в каждом объявлении раздел (метаданные объявлений) «Ключевые навыки» [27]. Команда проекта полагает, что ключевые навыки из описания вакансии – это валидный и достаточный источник данных для формирования моделей компетенций. Соискателям модели компетенций дают подсказку, чего именно им не хватает, чтобы получить должность или продвинуться по карьерной лестнице. Это фактически подробный чек-лист навыков, на который следует опираться и развиваться, если в каких-то навыках есть пробел [26].

Таким образом, на платформе «HH.ru» сейчас активно идут исследования и разработки в направлении использования навыков в процессе рекрутинга. Результат их деятельности представлен в новом проекте «Путеводитель по профессиям» (career.hh.ru/professions), позволяющий соискателям оценить свой набор навыков для той или иной профессии и рассчитать зарплату, на которую они могут претендовать (рисунок 1.6).

Использование навыков в сервисах «career.hh.ru» – это, несомненно, положительная тенденция. Однако есть отдельные проблемы.

1. Команда проекта для создания различных аналитических сервисов полагается на уже имеющиеся в вакансиях атрибуты «Ключевые навыки». Однако анализ вакансий на сайте показывает, что этот раздел либо вообще не заполнен (рисунок 1.4), либо содержание этого раздела не соответствует содержанию самой вакансии (рисунок 1.5). Как правило, этот раздел заполняют специалисты по подбору персонала, и часто они не обладают достаточными компетенциями для его заполнения. Следовательно, его нельзя использовать ни в качестве исходных данных для формирования профессиональных групп, ни для построения какой-либо аналитики. Например, на рисунке 1.5 список требуемых навыков для профессии «Data Scientist» слишком мал и не соответствует реальным требованиям работодателей.

2. Модели компетенций на данный момент используются только в малоизвестном сервисе, а для поиска не используются. Поэтому все разработки «HH.ru» в этом направлении не решают главной задачи – повышение эффективности работы поиска по базе вакансий.

Страница поиска представлена на рисунке 1.3, страницы с описанием вакансии – на рисунках 1.4 и 1.5, сервис «Путеводитель по профессиям» – на рисунке 1.6.

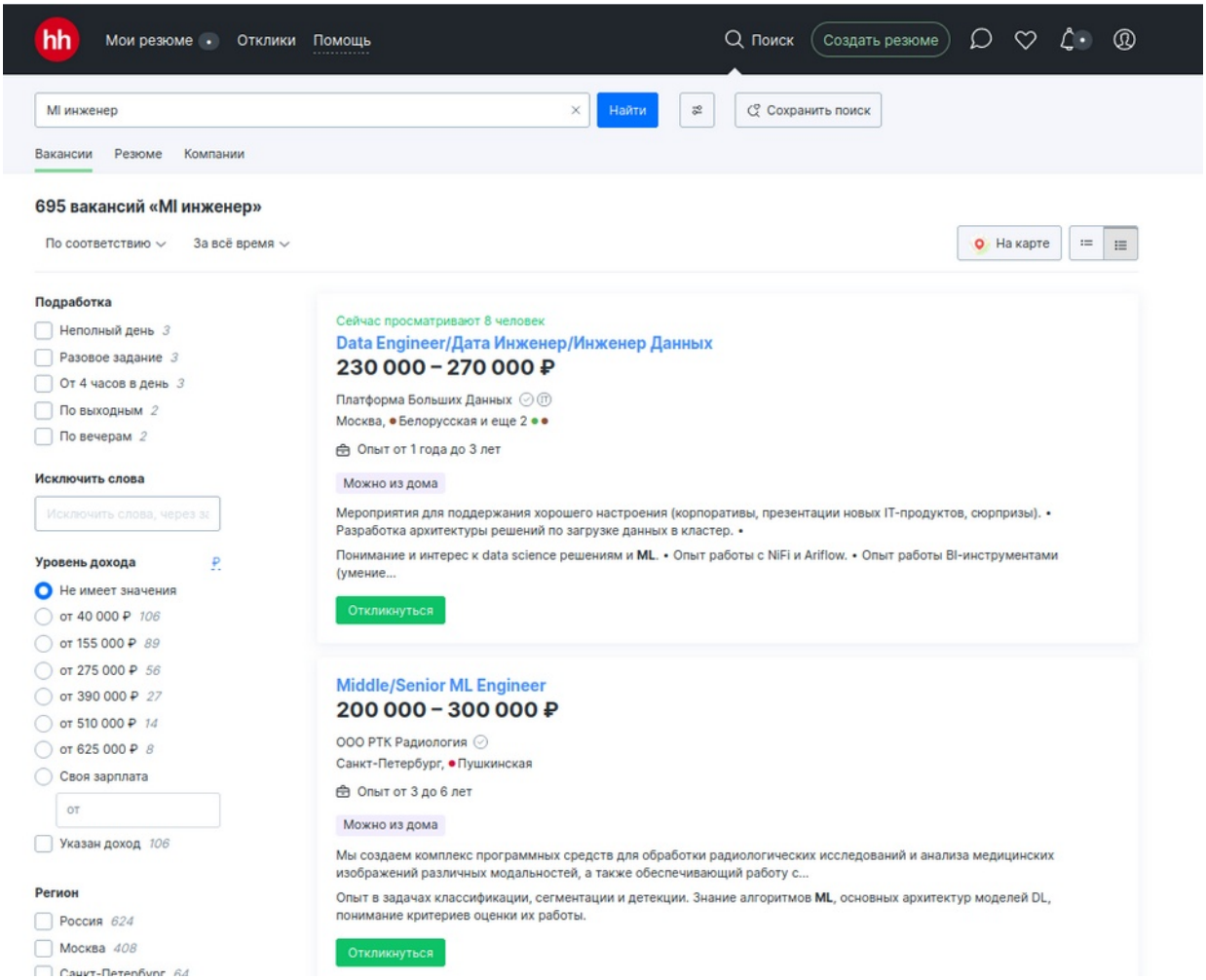
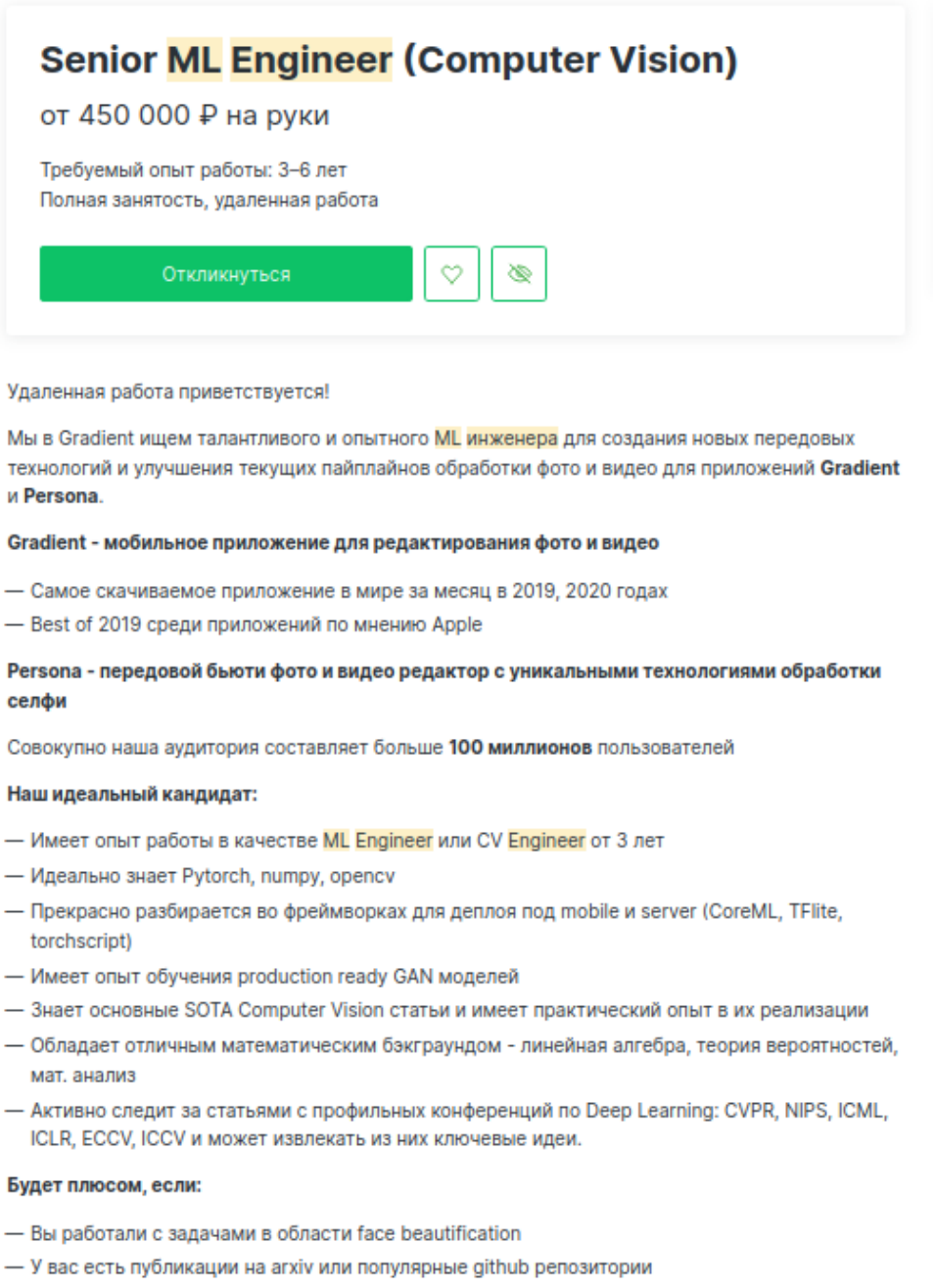


Рисунок 1.3 – Страница поиска на сайте «HH.ru»



Senior ML Engineer (Computer Vision)

от 450 000 ₽ на руки

Требуемый опыт работы: 3–6 лет
Полная занятость, удаленная работа

Откликнуться

Удаленная работа приветствуется!

Мы в Gradient ищем талантливого и опытного ML инженера для создания новых передовых технологий и улучшения текущих пайплайнов обработки фото и видео для приложений **Gradient** и **Persona**.

Gradient - мобильное приложение для редактирования фото и видео

— Самое скачиваемое приложение в мире за месяц в 2019, 2020 годах
— Best of 2019 среди приложений по мнению Apple

Persona - передовой бьюти фото и видео редактор с уникальными технологиями обработки селфи

Совокупно наша аудитория составляет больше **100 миллионов** пользователей

Наш идеальный кандидат:

— Имеет опыт работы в качестве ML Engineer или CV Engineer от 3 лет
— Идеально знает Pytorch, numpy, opencv
— Прекрасно разбирается во фреймворках для деплоя под mobile и server (CoreML, TFlite, torchscript)
— Имеет опыт обучения production ready GAN моделей
— Знает основные SOTA Computer Vision статьи и имеет практический опыт в их реализации
— Обладает отличным математическим бэкграундом - линейная алгебра, теория вероятностей, мат. анализ
— Активно следит за статьями с профильных конференций по Deep Learning: CVPR, NIPS, ICML, ICLR, ECCV, ICCV и может извлекать из них ключевые идеи.

Будет плюсом, если:

— Вы работали с задачами в области face beautification
— У вас есть публикации на arxiv или популярные github репозитории

Рисунок 1.4 – Пример объявления о вакансии на сайте «HH.ru» без раздела «Ключевые навыки»

— Автоматизация процессов переобучения и развертывания моделей с помощью Apache Airflow для поддержания точности моделей.

— Управление облачными хранилищами данных и ресурсами с использованием Yandex Cloud для обеспечения эффективной обработки данных и масштабируемости.

— Разработка и поддержание системы визуализации данных с помощью Yandex DataLens.

Требования:

— Подтвержденный опыт решения задач NLP, таких как классификация текстов, NER и сентимент анализ.

— Уверенное понимание процесса развертывания моделей в производственных средах, практический опыт работы с FastAPI, Docker, GitLab CI/CD и k8s.

— Опыт автоматизации обучения моделей с помощью Apache Airflow.

— Знакомство с облачными сервисами (Yandex Cloud, S3) для хранения и обработки данных. Владение навыками визуализации данных и создания информационных панелей, предпочтительно с помощью Yandex DataLens или аналогичных инструментов.

— Будет преимуществом инженерное образование.

— Сильные аналитические навыки с умением решать сложные задачи.

— Отличные коммуникативные навыки и умение работать в команде.

Условия:

— Режим работы: гибридный. Офис в Москве (м.Дмитровкая / м.Савеловская)
График: с 10:00 до 19:00 мск.

— Проектная работа. После испытательного срока повышение зарплаты и возможно зачисление в штат.

— Зарплата по результатам собеседования.

— Дружный амбициозный коллектив, перспективы роста.

— Компания - резидент Сколково, аккредитована в Минцифры. Лидер в своем сегменте.

Ключевые навыки

NLP | GPT 2 | Машинное обучение | SpeechSense | Речевая аналитика

Рисунок 1.5 – Пример объявления о вакансии на сайте «HH.ru» с разделом «Ключевые навыки», не соответствующим разделу «Требования»

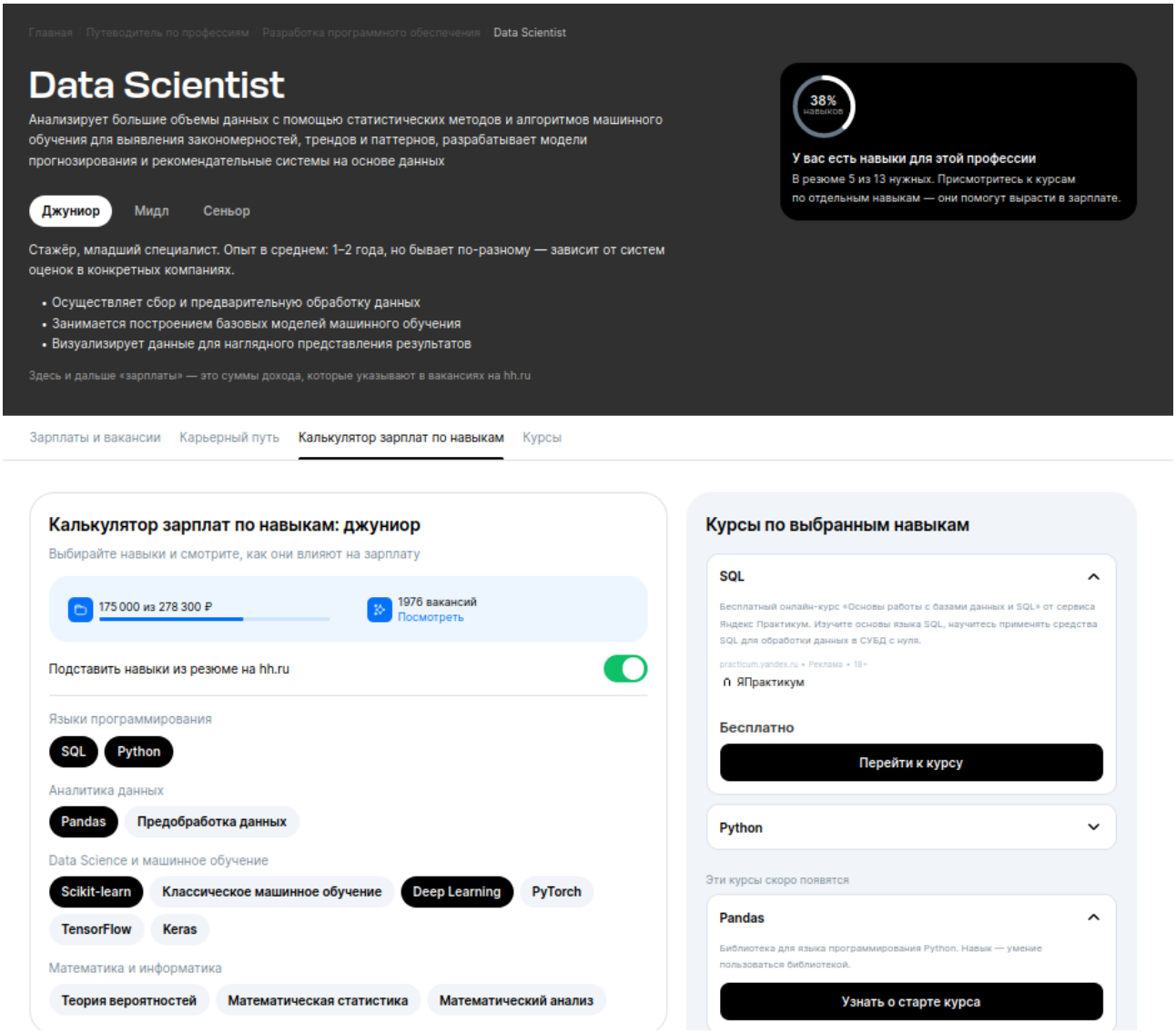


Рисунок 1.6 – Сервис «Путеводитель по профессиям» от «career.hh.ru»

Заслуженное второе место среди сайтов по поиску работы у «SuperJob.ru». «SuperJob.ru» также появился в 2000 году. Сам портал именует себя «лидером онлайн-рекрутмента в России», однако по статистическим показателям серьезно отстает от «HeadHunter». Однако 10 миллионов посещений в месяц, 400 тысяч вакансий и 25 миллионов резюме также серьезный показатель. Помимо этого, владельцы портала утверждают, что благодаря «SuperJob.ru» работодатели приглашают на собеседования по два миллиона человек в месяц. По их же данным, сайт помог найти сотрудников уже полутора миллионам работодателей. С 2005 года на базе «SuperJob.ru» работает собственный исследовательский центр, который занимается

мониторингом рынка труда в России [28].

Сайт также использует документальную ИПС с полнотекстовым поиском [29]. База вакансий существенно меньше, чем у его конкурента. Для фильтрации вакансий также используются дополнительные атрибуты: зарплата, регион, тип ставки, специализация, тип вакансии, тип занятости. Возможности сортировки также ограничены: только по релевантности (для полнотекстового поиска) и по метаданным объявления (дате и зарплате).

Для улучшение результатов поиска разработчики сайта «SuperJob.ru», так же, как и «HH.ru», создали справочник синонимов профессий для обогащения результатов, средства автоматического исправления опечатков в запросах. В отличие от «HH.ru», на сайте «SuperJob.ru» с помощью кластеризации и объединения решена проблема дублей и близких по описанию вакансий. Так же, как и в «HH.ru», в «SuperJob.ru» реализовано ранжирование результатов поисковой выдачи в соответствии с текстовым запросом на основе логов пользователей. Есть также возможность указания в объявлении профессиональных навыков, однако, также как и для «HH.ru», этот раздел заполнен не для всех объявлений, а если заполнен, то не соответствует требованиям в тексте вакансии, и самое главное – не используется в поиске.

В статье [30] сравнивалась поисковая выдача «Superjob.ru» и «HH.ru». В работе было проведено исследование и проверка на практике, насколько точно работает поисковая выдача при одинаковых запросах на «HH.ru» и «Superjob.ru». В результате оказалось, что релевантность выдачи у «Superjob.ru» выше.

Анализ различных систем онлайн-рекрутмента в целом показал, что качество поисковой выдачи для таких сайтов можно оценить как весьма посредственное, что связано с природой обрабатываемых данных. Большинство сайтов трудоустройства используют базу неструктурированных текстовых документов, и реализовать полноценный поиск для таких данных не представляется возможным.

Однако большинство сайтов предлагают возможности для частичной структуризации информации о требованиях работодателя к соискателю: при публикации вакансии работодатель может указать, например, ключевые навыки, зарплату, тип занятости и т. п, которые являются отдельными полями (или атрибутами, или метаданными), относящимися непосредственно к платформе онлайн-рекрутинга.

Используемые атрибуты, как правило, дополняют информацию о вакансии, но не являются репрезентацией всех требований, которые работодатель предъявляет к соискателю и при этом часто не соответствуют полному тексту в самом объявлении. Эти поля не извлекаются из текстов объявлений, а заполняются теми, кто их публикует, например кадровыми агентствами или службами персонала предприятий. Иногда они не могут корректно их заполнить в силу отсутствия необходимой информации или компетенции. Поэтому часто поле ключевых навыков вообще не заполнено, или указываются не все необходимые для успешной работы навыки. Другой проблемой этого атрибута является произвольный формат его заполнения.

Все это связано с тем, что требования к соискателю основаны на экспертных знаниях публикующего вакансию сотрудника работодателя, который отражает их в требованиях в самом тексте вакансии.

Поэтому основным источником информации для соискателя по-прежнему остается текст самой вакансии, представленный на естественном языке в неструктурированном виде.

Анализ систем онлайн-рекрутмента также показал, что в настоящее время отсутствуют информационно-поисковые системы для подбора вакансий на основе компетенций кандидатов. Большинство HR-систем основаны на поиске ключевых слов и фильтрации стандартных запросов к базам данных, отсутствует возможность определения соответствия компетенций кандидата его обязанностям в компании [4]

Большинство информационно-поисковых систем при первичном отборе объявлений ориентируются в первую очередь на текст заголовка, а во вторую

– на содержание текста объявления. В обоих случаях используются алгоритмы полнотекстового поиска по принципу вхождения каждого слова из запроса в искомую область (заголовок или текст). Как следствие отсутствует возможность отсеивать объявления, которые содержат требования, не соответствующие качествам кандидата. Это означает, что в поисковую выдачу попадают объявления, в которых требуется навык, которого нет у соискателя. Они определенно не подходят соискателю, но из-за такого подхода к реализации поиска он вынужден тратить значительное время на просмотр неподходящих объявлений. Такие лишние объявления обычно называют нерелевантными, и нерелевантность поисковой выдачи является неотъемлемой частью полнотекстового поиска.

Очевидным способом повышения эффективности поиска является структурирование данных и переход на фактографические информационно-поисковые системы. Структуризация осуществляется через извлечение информационных объектов (фактов) определенного типа, информация о которых имеется в документе.

Наиболее очевидным кандидатом для извлечения из текста объявления является перечень требований к кандидату. Попытки реализовать такую систему уже были, но в силу ряда причин они не имели коммерческого успеха. Примером является проект «Emply.ru» [31] . Сервис основан на технологии извлечения фактов из вакансий и резюме. Он отличается от существующих на рынке решений тем, что позволяют оперировать не категориями полнотекстового поиска (текст содержит «программист» или «developer»), а категориями предметной области (вакансия программиста). Принципиальным отличием от наиболее популярных средств поиска работы является то, что сервис индексирует объявления о вакансиях подобно классическим поисковикам, используя пауков. Но при этом найденные вакансии классифицируются по множеству признаков, позволяя их тонко фильтровать [32]. Для анализа текстов была создана система, которая позволяет анализировать тексты вакансий и резюме и не просто разбивать их

на блоки, а извлекать из них факты (должности, навыки, отрасли и т.п.). Чтобы это всё работало, также была создана база знаний предметной области, содержащая справочники этих самых фактов с их возможные названиями (то как этот объект может называться в резюме/вакансии).

К сожалению, данный сервис долго не просуществовал. Однако идея, заложенная при его создании, должна послужить основой для других систем для поиска вакансий.

1.4 Сравнительный анализ существующих методов извлечения информации из слабоструктурированных текстов описания вакансий

В предыдущем разделе установлено, что для организации эффективного поиска по вакансиям, необходима их предварительная структуризация с целью извлечения новых признаков – требований к навыкам, знаниям и умениям.

Описания вакансий с точки зрения возможности их автоматизированной обработки обладают рядом специфических особенностей [33]:

– в большинстве случаев небольшой размер, что затрудняет их статистический анализ;

– отсутствие структуризации, что усложняет процедуры извлечения информации;

– наличие определенного количества грамматических и синтаксических ошибок, что приводит к необходимости введения дополнительных этапов обработки;

– нестационарность тезауруса (состава и важности слов), который зависит от выхода новых нормативных документов, новых технологий, англицизмов, что приводит к необходимости использования процедур динамической кластеризации рубрик;

– название вакансии часто не соответствует и не отражает ее содержание или не соответствует должности и/или специализации кандидата,

что не позволяет выполнять группировку или фильтрацию объявлений только по их заголовкам.

Основная сложность при анализе требований вакансий – это большое количество их различных формулировок, что значительно осложняет их обработку и накладывает определенные ограничения на возможности применения методов машинного обучения, морфологического, синтаксического и семантического анализов [34].

Вопросу извлечения навыков из текстов вакансий посвящено множество работ ([35] , [36] , [37] , [38], [39]).

Общий алгоритм извлечения информации из текстовых документов представлен на рисунке 1.7 [34].

Для задач, связанных с извлечением информации из текстов на естественном языке, часто используют подходы, основанные на правилах, и методы машинного обучения с учителем на размеченныхэкспертами корпусах текстов. Эти подходы показывают лучшее качество и для русского языка [40].

В работе [37] демонстрируются выделения из резюме и вакансий ключевых навыков с помощью кластеризации текстов. Первой задачей, которая была решена, была кластеризация текстовых документов, поскольку набор ключевых понятий часто отличается для разных областей документов, даже входящих в один корпус Большинство резюме и вакансий принадлежат разным областям и сферам. Например, они могут быть разделены по профобластям (продажи, маркетинг, телеком, строительство и др.).

В работе [37] проверялись следующие алгоритмы кластеризации: LSA (Latent semantic analysis, Латентно-семантический анализ), STC (Suffix Tree Clustering) и K-means. В качестве проверочного корпуса использовались 50 тыс. вакансий, распределенных по профобластям, что составляет 1% от имеющегося корпуса вакансий – 5 млн.

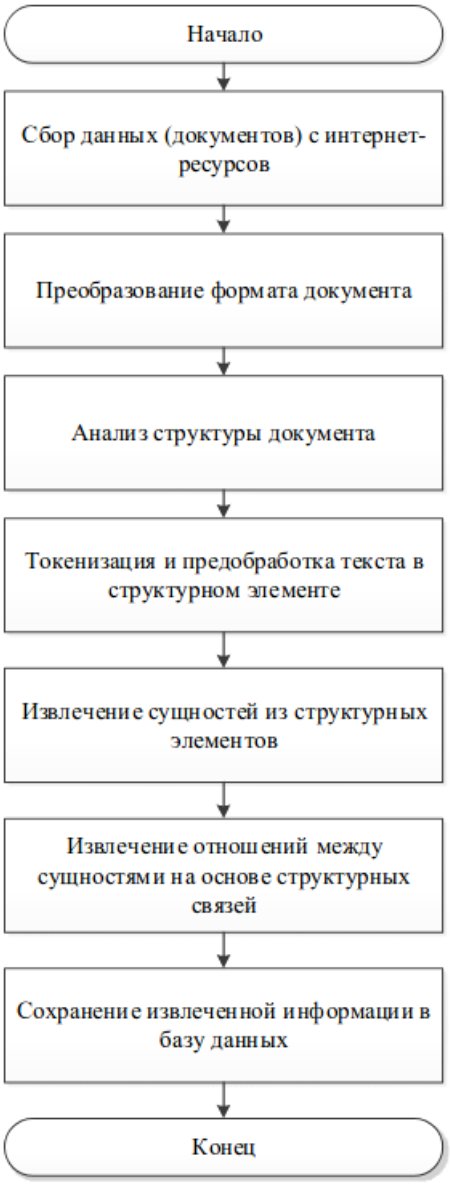


Рисунок 1.7 – Алгоритм извлечения информации из текстов вакансий [34]

Анализ показал, что описание вакансий хорошо поддается кластеризации и все выбранные алгоритмы справляются с поставленной им задачей. Немного лучше показал себя алгоритм латентно-семантического анализа (LSA). В достоинства этого алгоритма стоит так же записать, что он использует матрицу TF-IDF и не требует обучения. Для кластеризации резюме был применен этот же алгоритм (LSA), корпус резюме при этом состоял из 2,5 млн. документов.

Следующей задачей в работе [37] было разбиение текста вакансии на разделы. В качестве алгоритма классификации текста в таких «неразмеченных» вакансиях был использован алгоритм «наивной» Байесовой классификации. Так как обучающая выборка была достаточно большой (50% от 5 млн. вакансий), выбор алгоритма был не критичен в отношении к результатам, а его простота реализации и прозрачность оказались ключевыми факторами при выборе алгоритма. Заключительным этапом работы было выделение ключевых понятий из текста. Для этого использовалась матрица TF-IDF, которая позволяет получить распределение всех термов на каждый документ. Отбросив стоп-слова и общеупотребительные слова был получен рейтинг термов для каждой профобласти, после чего был определен минимальный порог вхождения терма в профобласть. Но так как все термы состоят из одного слова, а ключевые навыки в резюме и требования в вакансиях могут состоять из 2-х – 3-х слов, дополнительно были выделены биграммы и триграммы на документах каждой профобласти. В ходе экспериментов и анализа результатов выяснилось, что для кластеризации таких документов как резюме и вакансии требуется алгоритм, который формирует пересекающиеся кластеры, так как число резюме и вакансий, которые могли бы попасть с одинаковым успехом в разные профобласти, очень велико.

Аналогичный подход к выделению навыков представлен в работе [41] .

В работе Яруллина Д.В. [42] также использовалась кластеризация для выявления обобщенных требований работодателей к компетенциям ИТ-специалистов. Для векторного представления вакансий использовалась сглаженная метрика TF-IDF. Направления профессиональной деятельности выявлялись путем кластерного анализа вакансий с помощью алгоритма спектральной кластеризации, а число кластеров определялось кластеризацией множества ключевых понятий. Качество кластеризации оценивалось при помощи интегральной оценки на основе коэффициента силуэта, индекса Калински-Харабаша и индекса Дэвиса-Болдина.

В другой работе [43] Яруллина Д.В. попытался выполнить сравнение двух методов извлечения требований работодателей из текстов вакансий: онтологическая модель и кластерный анализ. Онтологическая модель - это иерархическая структура понятий предметной области, преимуществом которой является возможность машинной обработки и гибкость при масштабировании. Онтология представляет собой как целостную модель знаний. В качестве ключевого понятия для формирования структуры онтологии был выбран навык, позволяющий решать производственные задачи. На основе знаний о предметной области в работе были выделены девять групп навыков.

В качестве альтернативы в работе [43] рассматривался алгоритм кластеризации affinity propagation с предварительной векторизацией частотным методом. Среди достоинств этого подхода автор указал, что алгоритм самостоятельно формирует структуру и количество кластеров. Для оценки близости элементов использовалось косинусное сходство. С помощью кластерного анализа была выполнена группировка навыков для различных рабочих нагрузок. Для именования группы использовался центральный элемент кластера. В работе были получены кластеры, которые достаточно сильно пересекаются, и относительно низкий коэффициент силуэта (0,363). Алгоритм сформировал 11 групп по их центральным навыкам.

Автор отмечает, что в подходах используются разные принципы, определяющие принадлежность навыков к определенному классу: в кластерном анализе группу объединяет навык, который более схож со всем элементами кластера, в онтологии классом является абстрактное понятие, к которому относятся выделенные навыки. Онтологическая модель более сложна, группировка в онтологии выполняется по категориям, но отдельная категория не является совокупностью навыков, необходимых для определенного направления деятельности. Кластерный подход, напротив, формирует более динамичную модель, группируя навыки в соответствии с каким-либо направлением деятельности [43].

Среди преимуществ использования онтологической модели автором были отмечены возможность построить многоуровневую иерархию классов с жесткой системой организации, возможность соотнесения одного элемента с несколькими классам; среди недостатков: трудоемкость - составление онтологической модели требует значительных ресурсов и привлечения экспертов. Среди достоинств кластерный анализа автор указал возможность автоматизации процесса выделения групп, что позволяет снизить трудоемкость структуризации данных, среди недостатков — невозможность выделения всех навыков предметной области и неявных зависимостей между ними. Автор отметил аналогичную эффективность обоих подходов [43].

Другой подход с использованием нейронных сетей предложен в работе [44]. Автор предлагает выполнять дообучать на текстах онлайн-вакансий нейросетевую модель BERT для перехода от сложного текста к набору простых сочетаний слов. Использовался процесс дообучения нейросетевых моделей BERT на текстах онлайн-вакансий. Также были реализованы два механизма добавления новых связей между словами требований с учѐтом знаний из предметной области: линейный и через дополнение дерева синтаксического разбора. После проведенного сравнительного анализа различных инструментов, была выбрана лучшая комбинация: дообученная модель BERT, deeppavlov_syntax_parser и линейный способ дополнения связей. Данный метод показал более высокую эффективность, по сравнению с подходом, основанным на правилах.

В работе [40] изучаются возможности решения задач извлечения информации в отдельных предметных областях с применением нейросетевых моделей, таких как word2vec, fasttext, paragraph2vec, обучаемых без учителя на больших текстовых корпусах. Указывается, что данные подходы наиболее эффективны при определении семантической близости, в том числе и для русского языка. В работе описан подход к извлечению информации на основе определения семантической близости векторов предложений и сущностей базы знаний с помощью нейросетевых моделей языка. Предложенный в

работеметод позволил без выполнения разметки текстового корпуса и без использования методов, основанных на правилах, достичь приемлемого качества в решении задачи выявления актуальных требований рынка труда. В роли базы знаний предметной области с ограниченной лексикой использовались профессиональные стандарты. В работе определялась семантическая близость между векторными представлениями текстов, полученных спомощью различных нейросетевых моделей языка: усредненный word2vec, взвешенный по SIF усредненный word2vec, взвешенный по TF-IDF усредненный word2vec, paragraph2vec. В результате проведенных экспериментов лучшее качество работы показала модель усредненного word2vec (CBOW).

В работе [45] представлен подход, в основе которого лежит использование ориентированных графов для классификации компетенций, а также алгоритмы сопоставления навыков с требованиями вакансии. Предложенный подход позволил учитывать сложные иерархические отношения между навыками. В основе алгоритма, основанного на ориентированном графе, лежит структура, где каждая компетенция или навык представлены узлом, а связи между ними формируют ориентированные рёбра. При этом одно и то же понятие может быть выражено несколькими фразами, и каждая фраза может быть связана с другими понятиями. Этот подход подразумевает предварительное построение графа, в котором каждая фраза из резюме представляет узел, а связи между ними устанавливаются на основе семантической близости. Путем поиска на этом графе можно преобразовывать выражения, представленные в резюме или вакансиях, в конкретные навыки.

Таким образом, существующие подходы обработки текстов вакансий подразумевают решение двух основных задач:

– векторизация текста;
– извлечение ключевых слов.

Для векторизации текста вакансий в большинстве работ используются следующие подходы [46]:

– частотные методы: «мешок слов» и TF-IDF матрица; это методы, основанные на получении математической матрицы, описывающей частоту встречающихся терминов;

– представление в векторной форме на основе Word2Vec, doc2vec, Paragraph2Vec (эмбедингов); каждый абзац и каждое слово представляется в виде уникальных векторов, и векторы абзаца и слова усредняются или объединяются, чтобы предсказать следующее слово в контексте;

– представление в векторной форме на основе трансформеров - языковых нейросетевых моделей, обучаемые на большом количестве текстовых данных; существует возможность использования предобученных моделей на разных языках, которые можно найти на Hugging Face Hub.

Векторные представления слов при использовании TF-IDF являются one-hot векторами, содержащими лишь одно значение, равное TF-IDF весу этого слова. Размерность такого вектора равна количеству уникальных слов в корпусе. Главным недостатком таких векторов является их чрезвычайная разреженность. Преимуществами же TF-IDF являются простота ее расчета ипонятность в интерпретации значений векторов [40].

Положительными характеристиками модели Word2Vec являются низкая разреженность итоговых векторов, возможность задания их размерности, а также скорость работы. Основным недостатком является невозможность интерпретации значений координат некоторого вектора. Для получения векторного представления целого текста необходимо объединить векторные представления отдельных слов. Используя paragraph2vec, можно получать векторные представления текстов без каких-либо дополнительных действий [40].

Сегодня одним из наиболее перспективных и популярных подходов к анализу естественного языка являются нейросетевые модели, использующие механизм внимания – способность поиска взаимосвязей между различными

частями текста, и построенные на архитектуре трансформеров. Особенность BERT заключается в том, что он может генерировать векторное представление, учитывает контекст для всех слов и способен лучше справляться с долгосрочными зависимостями в тексте. В настоящее время BERT превосходит нейросетевые модели предыдущего поколения, такие как word2vec, LSTM и др. Последние достижения в компьютерной лингви стике позволили перейти к эффективным векторным представлениям для целых предложений и абзацев текста. Примером является проект SentenceTransformer (SBERT), который представляет собой технологию модификации предварительно обученной сети BERT. В модели используются сиамские и триплетные сетевые структуры для получения семантически значимых векторов предложений. Это позволяет дообучать модель на задаче определения семантически близких текстов. В настоящее время для SentenceTransformer на сайте разработчиков опубликованы ссылки на множество предобученных моделей, в том числе и для русского языка [47].

Таким образом, в настоящее время подход на основе использования глубоких нейронных сетей и трансформеров способен решать больше задач, чем классические методы, а также чаще показывает лучшие результаты на бенчмарках, и является наиболее предпочтительным.

Для извлечения ключевых слов существуют следующие подходы:

– кластерный анализ;
– построение онтологии и ориентированных графов;
– построение таксономии;
– использование нейронных сетей и профессиональных стандартов для определения семантической близости.

Из всех перечисленных методов для решения задачи построения веб-сервиса, ежедневно обрабатывающего тысячи объявлений из разных источников, наиболее подходящим видится кластерный анализ как наиболее динамичный подход для отслеживания всех изменений рынка труда.

1.5 Выводы по главе 1

В результате проведенного анализа было установлено, что:

1. Для соискателей наиболее перспективным методом поиска вакансий являются специализированные сайты трудоустройства, а также различные сайты-агрегаторы, собирающие объявления из различных источников.

2. Особенно эффективно использование сайтов трудоустройства в IT-сфере в силу ее постоянной изменчивости и высоким требованиям к квалификации кандидатов.

3. Для успешного трудоустройства необходимо наличие инструментов, позволяющих соискателю получать актуальную информацию о требованиях рынка труда к квалификации кандидата в определенной сфере деятельности с учетом возможной специализации, а также оценивать востребованность своих знаний и умений в выбранной профессии и специализации.

4. Полнотекстовый поиск, используемый на сайтах по подбору персонала, не способен покрыть информационную потребность соискателя – подбор вакансий, которые ему действительно подходят, и отсев объявлений, для которых у него не хватает навыков и квалификации.

5. Использование строки запроса на естественном языке в свободной форме не удобно для поиска вакансий по конкретному набору навыков.

6. Лучшим решением для сайта по подбору персонала является смешанная документально-фактографическая информационно-поисковая система, в которой объявление о вакансии состоит из значений отдельных атрибутов (фактов) и собственно самого документа, содержащего текстовое неструктурированное описание вакансии. Для поиска вакансий должны использоваться их атрибуты, а не текстовое описание.

7. Информационно-поисковая система, используемая на сайте по подбору персонала, должна предлагать возможные требуемые навыки из заранее заданного списка при поиске вакансий на определенную должность или специализацию. Поиск вакансий должен осуществляться по указанному

пользователем перечню требуемых навыков.

8. Перечень наименований навыков должен формироваться автоматически и следовать тенденциям изменений рынка труда.

9. Для реализации поиска по навыкам требуется выполнить структуризацию текста вакансий с извлечением навыков и записью их в соответствующие атрибуты документа. Для этого существуют разные подходы, и лучшим методом векторизации являются трансформеры, а для извлечения информации в условиях постоянно меняющегося рынка труда является кластеризация.

2 Методы и модели извлечения структурированной информации из текстов вакансий

2.1 Характеристика исходных данных и алгоритм их обработки

Для выполнения работы была собрана коллекция текстов объявлений из архива портала HeadHunter – крупнейшего российского онлайн-сервиса для поиска работы.

Информация о вакансии, возвращаемая открытым API проекта «HeadHunter» позволяет получать доступ к вакансиям, размещенным на портале в формате json. При этом он поддерживает поиск по ключевым словам и фильтрацию по огромному числу параметров.

API содержит достаточно большое число параметров, и при этом имеет существенное ограничение: API разбивает ответ на страницы, на один запрос выдается не более 500 вакансий, а общая глубина поиска не может превышать 2000 вакансий. Поэтому для получения полного архива текстов нужны специальные стратегии разбиения запросов на подзапросы по отдельным областям. Для решения данной проблемы сбора данных в этой работе использовался архив из работы [48], полученный авторами скачиванием с помощью официального API «HeadHunter» (https://api.hh.ru/), с последующей переработкой и дополнением данными с сайта «HH.ru».

В работе использовался набор датасетов, содержащий архивные данные с сайта «HH.ru» с 2015 по 2022 гг. Изначальный объем данных – около 3 млн. строк, что очень много для используемой технологии анализа данных. Размеры наборов данных представлены в таблице 2.1.

Каждое объявление о вакансии содержит достаточно большой набор характеристик. В таблице 2.2 перечислены те из них, которые вошли в исследуемые датасеты.

Таблица 2.1 – Наборы данных для обработки

Датасет	Строк	Столбцов
hh_data_it_2022	40398	40
hh_data_it_2021	287915	19
hh_data_it_2020	587637	19
hh_data_it_2019	535956	19
hh_data_it_2018	517670	19
hh_data_it_2017	391464	19
hh_data_it_2016	332460	19
hh_data_it_2015	284763	19

Таблица 2.2 – Признаки, полученные при парсинге данных с сайта-агрегатора api.hh.ru

№	Имя и тип признака в датасете	Название поля
1	id (int)	Уникальный ID вакансии
2	published_at (string)	Дата первоначальной публикации объявления на сайте
3	name (string)	Название вакансии
4	employer_id (string) и employer_name	Идентификатор и название компании
5	salary_from (string) и salary_to (string)	Нижняя и верхняя границы предлагаемой зарплаты
6	salary_currency (string)	Код валюты
7	spec_id_0, spec_id_1, spec_id_2 и spec_0, spec_1, spec_2	коды и названия специализаций в классификаторе «HH.ru»
8	experience_id (string)	Требуемый опыт работы (нет опыта, от 1 года до 3 лет, от 3 лет до 6 лет, более 6 лет)
9	key_skills (Array of string)	Ключевые навыки
10	area_id (string) и rea_name (string)	Идентификатор и название региона
11	description (string)	Описание в HTML

В настоящий момент каждый день на портале публикуется свыше 100 тыс. новых объявлений, а сам архив содержит несколько миллионов

объявлений. Такой объем вакансий и охват практически всех видов деятельности и профессий дает основания считать эту информацию достаточно представительной. Однако для задач исследования возможностей структуризации текста вакансий такой объем информации избыточен. Кроме того, обработка такого большого массива информации требует определенных инфраструктурных решений, выходящих за рамки этой работы. Поэтому для дальнейшего анализа объем набора данных был сокращен следующим образом.

1. Фильтрация по специальности

Несколько столбцов в исходных датасетах содержали информацию о специализации в соответствии с внутренней классификацией портала «HH.ru». Согласно этой классификации, каждая вакансия может относиться к нескольким специализациям. Для данного исследования достаточно одной специализации. Она будет использоваться только для предварительной фильтрации данных. Всего в датасете было представлено 482 специализации. Для выполнения данной работы были оставлены только вакансии, относящиеся к наиболее многочисленной группе: профессиональной области «1. Информационные технологии, интернет, телеком» и специализации «1.221. Программирование, Разработка». За счет этого удалось значительно уменьшить объем датасета до 1,15 млн строк

2. Удаление строк с ошибками

Ввиду наличия HTML-разметки в одном из столбцов и некорректного ее обрамления кавычками в некоторых строках в csv файлах, они не были правильно распознаны библиотекой Pandas, и в процессе импорта данных появились новые безымянные столбцы. Такие строки были удалены.

3. Удаление дубликатов

В ходе анализа данных было установлено, что в столбцах с текстовым содержимым много одинаковых данных. Это особенно характерно для признака с описанием вакансии. Для нас это означает смещение выборки, поэтому такие данные должны быть удалены.

Анализ данных показал, что дублирование объявлений происходит в следующих ситуациях:

– работодатели одно и то же объявление публикуют для разных должностей, поскольку слабо представляют различия в должностных обязанностях и требованиях к соискателям для разных позиций (например, «Инженер технической поддержки» и «Главный инженер»);

– одну и ту же должность по-разному именуют для большего охвата аудитории (например, «Специалист контакт-центра» и «Менеджер по работе с входящими обращениями»), т.е. это по сути синонимы названия должности;

– одну и ту же позицию продвигают для разных населенных пунктов, отличия в названии заключаются в добавлении к заголовку вакансии названия населенного пункта или адреса (например, «Инженер технической поддержки (Обь)» и «Инженер технической поддержки (Новосибирск)»).

В ходе анализа были выявлены объявления с одинаковыми названием, описанием, работодателем, навыками и опытом. При этом такие объявления отличаются только полями, не создающими отличительных признаков вакансии, таких как дата публикации, регион, идентификатор и т.п. Было установлено, что для повторной публикации работодатели часто идут на хитрость – немного изменяют значения некоторых полей, таких как заголовок, требования к опыту, зарплата, список ключевых навыков. Неизменным всегда остается в таком случае только сам текст с описанием вакансии. Поэтому, чтобы избежать удаления объявлений, которые действительно представляют разные должности от одного работодателя, удалялись объявления, в которых кроме описания и работодателя, совпадает по крайней мере еще одно поле из перечисленных выше.

Анализ также показал, что у одного и того же работодателя могут быть объявления с очень близкими названиями, при совпадающих других параметрах. Можно предположить, что такие объявления также являются дублями, но перед проверкой на совпадение они были приведены к некой общей базе. Для этого названия были очищены: удалены специальные

символы, текст в скобках, обрезаны ненужные «хвосты» (например, название региона, в котором публикуется объявление) и сохранены в новом столбце с названием 'short_title'. После этого удалось обнаружить много новых дублей.

В ходе многоступенчатой чистки датасета от дублей была удалена почти его треть, после чего в датасете осталось 735850 обявлений.

4. Удаление строк с ненужными или отсутствующими значениями

Были удалены объявления с незаполненными ключевыми навыками, и оставлены объявления только на русском языке, поскольку работа с многоязыковыми корпусами текстов не входила в планы этой работы.

В результате после нескольких этапов чисток и удаления в датасете осталось 334380 объявления.

Кроме сокращения объема выборки с датасетом также были выполнены следующие действия:

1. Отбор необходимых для анализа признаков

Были удалены 6 столбцов со специализацией, поскольку после фильтрации по специальности они стали больше не нужны. Также были удалены ненужные для обучения другие поля. В результате после предварительной обработки в датасете осталось 6 полей:

– id (int): идентификатор вакансии;

– name (string): оригинальное название;

– short_name (string): укороченное название, очищенное от лишней информации

– description (string): описание в HTML;

– key_skills (Array of string): ключевые навыки, список названий ключевых навыков;

– employer_name: название компании.

2. Заполненние пропусков в столбцах 'key_skills', 'salary_from', 'salary_to', 'salary_currency', 'employer_id' значениями по умолчанию.

Часть работодателей не заполнили эти поля. Для решения поставленной задачи эти столбцы существенной роли не играют, поскольку нужны только

для дополнительной информации. Однако для корректной работы некоторых алгоритмов фильтрации они должны быть заполнены.

3. Исправление неверного типа данных в некоторых столбцах

На этом предварительная обработка датасета была закончена. Общая последовательность предварительной обработки представлена на рисунке 2.1.

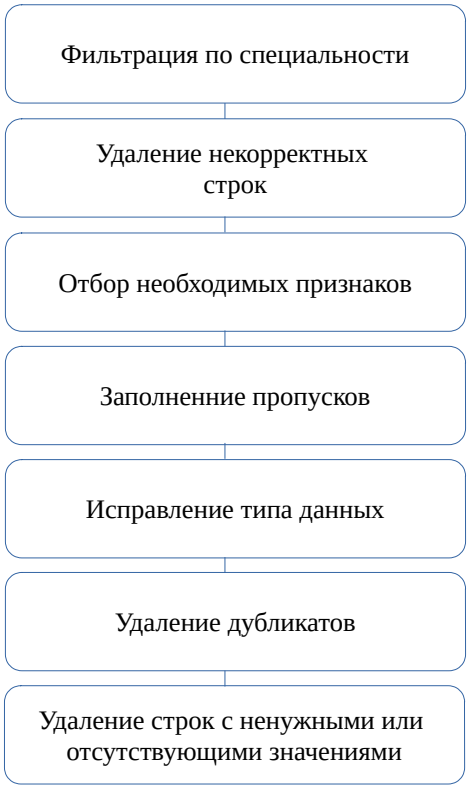


Рисунок 2.1 – Алгоритм предварительной обработки наборов данных

Важной частью подготовки данных является предварительная обработка текстов. Под предварительной обработкой текста обычно понимают процесс преобразования последовательности слов, встречающихся в документе, в n-мерное векторное пространство.

Предобработка текста является обязательным этапом при решении задач обработки естественного языка, и она обычно включает в себя:

1. Графематический анализ

Графематический анализ – это выделение элементов структуры текста (графем): абзацев, предложений, слов и иногда символов. Чаще всего ограничиваются разделением по разделителям (пробелы, знаки препинания,

перевод на следующую строку) на слова – токены, и процесс называется токенизацией. Также может выполняться извлечение наиболее значимых слов (термов).

2. Морфологический анализ

Морфологический анализ – анализ словоформ и приведение их к какому-то нормализованному виду. Русский язык, кроме того, что он сам по себе не структурированный, имеет также такие неприятные особенности, как неоднозначность, флективность – изменяемость формы слова, свободный порядок слов предложений. Формы слова в русском языке меняются, причем меняются достаточно сильно. Эти особенности затрудняют его автоматическую обработку.

Однако несмотря на такую вариативность и неоднозначность русский язык, как и английский, все-таки имеет достаточно жесткие правила, позволяющие людям интерпретировать текст правильно. Для того, чтобы облегчить понимание текста компьютерами, требуется его дополнительная обработка, смягчающая эффект неоднозначности и флективности. К числу таких методов относят:

– лемматизация и стемминг – приведение к какой-либо стандартной форме;

– определение частей речи и добавление к лемматизированным токенам частеречной разметки (POS-теги);

– постановка слова в заданную форму.

У каждого слова есть основа – это его начальная, неизменяемая часть. Также у каждого слова есть нормальная форма. Лексема – это набор всех форм одного слова. Словоформа – это слово в том виде, в котором она встречается в тексте. Лемматизация – это приведение словоформы к нормальной (словарной) форме, стемминг – к его основе, когда от слов отбрасываются окончания.

3. Очистка текста

Производится удаление символов, которые не являются буквенно-цифровыми, пробелами или выбранными специальными символами; здесь также часто выполняется удаление и HTML-тегов и других символов разметки, стоп-слов и служебных частей речи (союзы, предлоги и местоимения). Может выполняться как после графематического, так и после морфологического анализов.

4. Синтаксический анализ

Выполняется анализ сочетаний слов внутри предложения, а также построение древа зависимостей для всех слов в предложении.

5. Векторизация текста

Векторизация текста – формирование цифрового представления для каждого символа.

Для каждого этапа данной работы выполнялась отдельная обработка текстов вакансий в следующей последовательности:

1) токенизация: разделение на разных этапах работы на абзацы, выражения или слова, в зависимости от задачи этапа исследования;

2) очистка текста: удаление тегов, специальных символов и знаков пунктуации; на каждом этапе была своя очистка;

3) лемматизация: приведение слов к нормальной форме;

4) повторная очистка текста: удаление стоп-слов и лишних пробелов;

5) векторизация текста: на разных этапах использовались разные модели векторизации.

Лемматизация выполнялась с помощью библиотеки pymorphy3 – морфологического анализатора для русского и украинского языков [49]. Она распространяется бесплатно и имеет открытые исходные коды. При работе используется словарь OpenCorpora; для незнакомых слов строятся гипотезы. Библиотека достаточно быстрая – до 100тыс слов/сек, и с исследуемой коллекцией текстов она справилась успешно.

Векторизация – это подход к преобразованию входных данных из исходного формата (например, текста) в векторы действительных чисел,

которые понятны моделям машинного обучения. Идея заключается в получении некоторых отличительных признаков из текста для обучения модели путем преобразования текста в числовые векторы.

Для векторизации были использованы так называемые «плотные векторные представления» – word embeddings. Для создания эмбеддингов используют нейронные сети. Суть этого метода заключается в том, что нейронная сеть обрабатывает большое количество текста и на основании него создает векторы для каждого слова. Эти векторы являются на самом деле весами обученной нейронной модели, которая либо берет на вход какое-то слово и пытается предсказать ближайшие окружающие его слова, либо, наоборот, берет на вход окружающие слова, а на выходе предсказывает загаданное слово.

Первым представителем такого подхода был метод Word2Vec, разработанный компанией Google в 2013 году. На выходе у Word2Vec каждое слово имеет какой-то длинный вектор чисел. Одним из недостатков такой модели является то, что вектор чисел каждого слова не зависит от контекста этого слова.

Указанный недостаток был исправлен в следующем поколении моделей – BERT, также разработанном Google. По сравнению с Word2Vec модели BERT намного сложнее и более гибкие относительно контекста слова. В отличие от Word2Vec с одним слоем нейронная сеть BERT состоит из множества слоёв, поэтому BERT долго и сложно обучается на очень большом количестве данных. Основные отличия BERT от Word2Vec:

– BERT обучается на более длинных текстах;

– BERT учитывает расположение слова в предложении;

– BERT дает разные векторы для одного и того же слова в зависимости от контекта.

Впервые появившись в 2018 году, модель BERT совершила переворот в компьютерной лингвистике. Базовая версия модели долго предобучается, читая миллионы текстов и постепенно осваивая язык, а потом её можно

дообучить на собственной прикладной задаче.

Все вместе эти модели (BERT, Word2Vec и их производные) называются трансформерами.

В данной работе использовалась библиотека на языке Python под названием Sentence-transformers [50]. Эта библиотека предоставляет простые методы вычисления эмбедингов для предложений, абзацев и изображений. Эта библиотека включает как готовые предварительно обученные модели для векторизации текста, так и соответствующие им средства токенизации. Есть многоязыковые модели с поддержкой русского языка, однако токенизация производится без учета флективности русского языка.

Библиотека Sentence-transformers содержит множество предобученных моделей для различных задач, в том числе для классификации текстов. Для сравнительного тестирования были отобраны несколько моделей, обученных на данных разного размера, общим свойством которых является то, что эти модели могут находить семантически похожие предложения как на одном языке, так и на разных языках (что важно для вакансий, поскольку их описания согут быть на разных языках), и их основное предназначение – кластеризация или семантический поиск:

– «stsb-xlm-r-multilingual» – версия моделей-кодировщиков «XLM-RoBERTa», признанных надёжными мультиязычными решениями; стандартный BERT embedding имеет размерность 768 [51];

– «distiluse-base-multilingual-cased-v2» – многоязыковая дистиллированная версия модели «multilingual Universal Sentence Encoder» [52]; поддерживает более 50 языков, но работает немного хуже, чем модель v1; отображает предложения и абзацы в 512-мерное плотное векторное пространство [53];

– «paraphrase-multilingual-MiniLM-L12-v2» – многоязычная версия «paraphrase-MiniLM-L12-v2», обученная на параллельных данных для более чем 50 языков; отображает предложения и абзацы в 384-мерное плотное векторное пространство [54];

– «paraphrase-multilingual-mpnet-base-v2» – многоязычная версия «paraphrase-mpnet-base-v2», обученная на параллельных данных для более чем 50 языков; отображает предложения и абзацы в 768-мерное плотное векторное пространство [55].

Указанные модели генерируют выровненные векторные пространства, т.е. аналогичные входные данные на разных языках отображаются близко в векторном пространстве. Язык ввода для них указывать не нужно.

На всех этапах этой работы исследовались эти 4 модели, и на основе сравнительного анализа выбиралась одна, обеспечивающая лучшие показатели качества используемого метода.

2.2 Методы и модели извлечения основных структурных элементов из текстов вакансий и идентификации их типа

Основным элементом объявления о вакансии является поле «description» – её описание, оформленное с использованием HTML разметки. Оно содержит всю ту информацию, которую работодатель хотел донести до соискателя. Для эффективного поиска по базе, содержащей миллионы таких вакансий, необходима структуризация этого описания и выделение полезной информации в виде отельных полей. В рамках данной работы решается вопрос выделения только требований к навыкам как основы такого поиска.

Для автоматического извлечения требований к кандидату сперва необходимо определиться, в каком разделе описания их можно найти. Большинство текстов вакансий «HH.ru» имеют неявно выраженную структуру, т.е. имеют визуально различимые блоки различного семантического назначения, оформленные с помощью тегов HTML.

Текст большинства вакансий содержит такие необязательные части, как (рисунок 2.2):

1) описание вакансии;
2) обязанности;
3) обязательные требования;

4) необязательные требования;
5) условия работы и дополнительные выгоды, предлагаемые работодателем;
6) ключевые навыки.

Разметка текста на блоки обычно выполняется с помощью блочных элементов языка HTML.

Границы блоков в идеальном случае определяются границами списка его элементов. Однако в вакансии может вообще не быть списка.

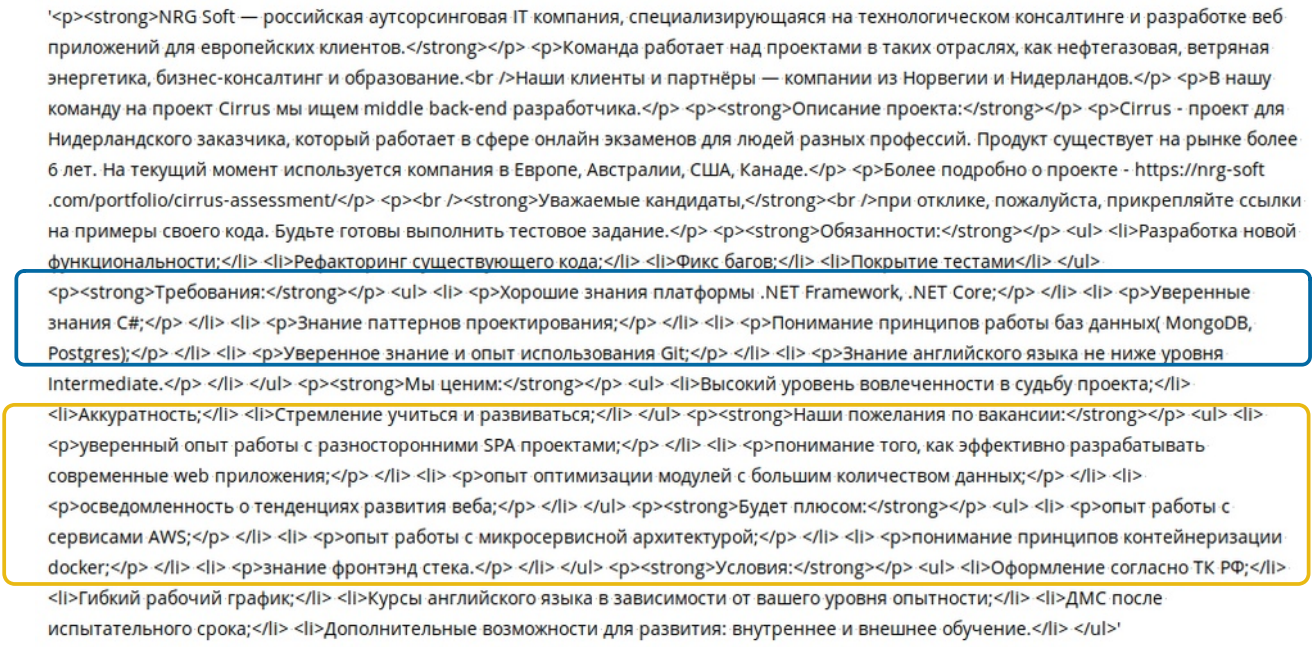
Рисунок 2.2 – Пример структуры объявления

Дополнительно в блоке могут присутствовать еще два типа элементов: просто текст (блок №1) и строка перед списком или текстом (заголовок блока, подчеркнуто в блоках 2, 3, 5 и 6).

Однако на практике далеко не все авторы объявлений придерживаются этих правил: списки могут или отсутствовать вовсе, или могут оформлены строчным тегом, или вообще с помощью знаков пунктуации. Заголовок также может никак не размечаться дополнительно, т.е. с точки зрения кода – это обычный абзац. Некоторые блочные элементы размечаются строчными тегами. А иногда все объявление помещается в один-единственный блочный тег. Часто наблюдаются ошибки разметки, неверное использование тегов HTML. Семантическая разметка нигде не используется.

Пример хорошо идентифицируемой HTML-разметки приведен на рисунке 2.3, пример описания с нестандартной структурой – на рисунке 2.4.

Способ разбиения на блочные элементы определяет автор объявления, поэтому в целом все объявления имеют совершенно разную структуру даже на уровне HTML-разметки.



' NRG Soft — российская аутсорсинговая IT компания, специализирующаяся на технологическом консалтинге и разработке веб приложений для европейских клиентов. Команда работает над проектами в таких отраслях, как нефтегазовая, ветряная энергетика, бизнес-консалтинг и образование.
Наши клиенты и партнёры — компании из Норвегии и Нидерландов. В нашу команду на проект Cirrus мы ищем middle back-end разработчика. Описание проекта: Cirrus - проект для Нидерландского заказчика, который работает в сфере онлайн экзаменов для людей разных профессий. Продукт существует на рынке более 6 лет. На текущий момент используется компания в Европе, Австралии, США, Канаде. Более подробно о проекте - https://nrg-soft .com/portfolio/cirrus-assessment/
Уважаемые кандидаты,
при отклике, пожалуйста, прикрепляйте ссылки на примеры своего кода. Будьте готовы выполнить тестовое задание. Обязанности: <ul> Разработка новой функциональности; Рефакторинг существующего кода; Фикс багов; Покрытие тестами Требования: <ul> Хорошие знания платформы .NET Framework, .NET Core;   Уверенные знания C#;   Знание паттернов проектирования;   Понимание принципов работы баз данных(MongoDB, Postgres);   Уверенное знание и опыт использования Git;   Знание английского языка не ниже уровня Intermediate.   Мы ценим: <ul> Высокий уровень вовлеченности в судьбу проекта; Аккуратность; Стремление учиться и развиваться; Наши пожелания по вакансии: <ul> уверенный опыт работы с разносторонними SPA проектами;   понимание того, как эффективно разрабатывать современные web приложения;   опыт оптимизации модулей с большим количеством данных;   осведомленность о тенденциях развития веба;   Будет плюсом: <ul> опыт работы с сервисами AWS;   опыт работы с микросервисной архитектурой;   понимание принципов контейнеризации docker;   знание фронтэнд стека.   Условия: <ul> Оформление согласно ТК РФ; Гибкий рабочий график; Курсы английского языка в зависимости от вашего уровня опытности; ДМС после испытательного срока; Дополнительные возможности для развития: внутреннее и внешнее обучение. '

Рисунок 2.3 – Пример хорошо размеченного описания (обязательные требования – в синем прямоугольнике, необязательные – в желтом)

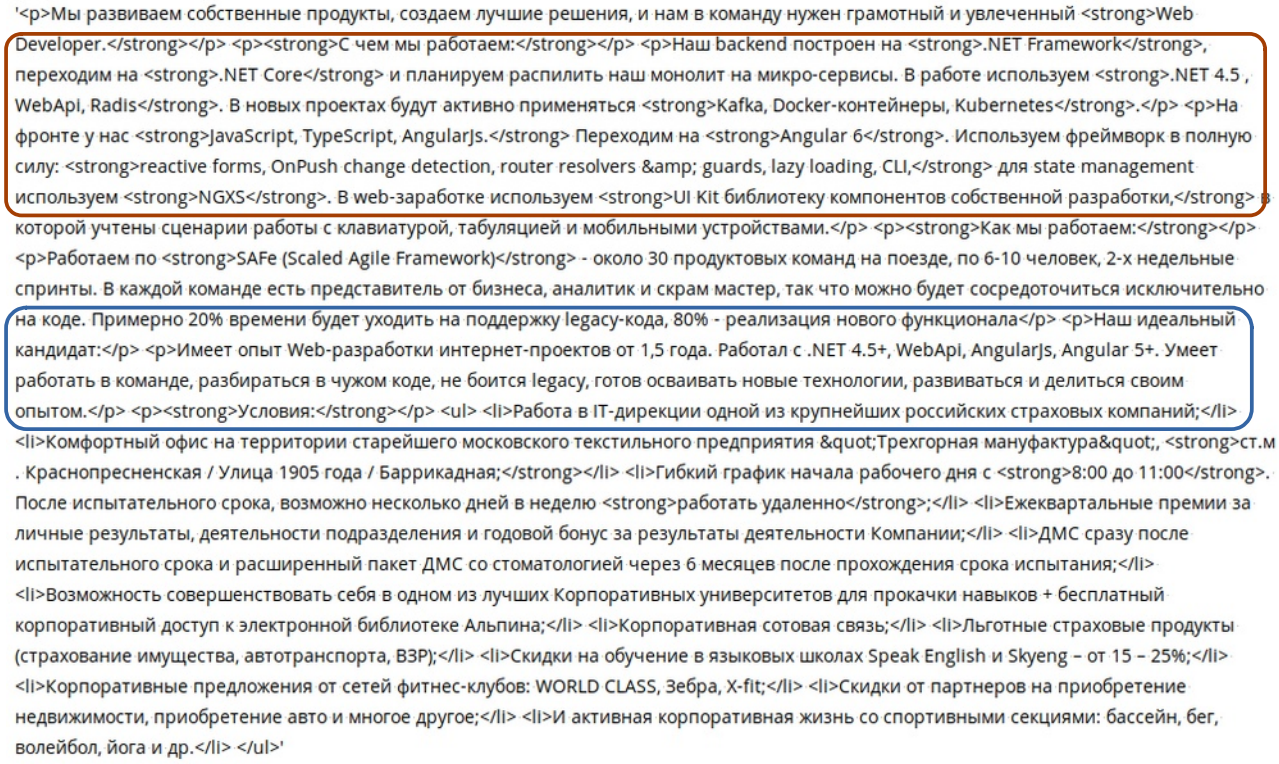


Рисунок 2.4 – Пример описания с нестандартной структурой: все требования к навыкам в одном предложении (в синей рамке), списка нет. Часть требований находится в описании вакансии (в красной рамке)

Количество неправильно оформленных объявлений в общей массе достаточно велико (не менее 20%), поэтому написать какой-то один универсальный скрипт, который мог бы эту структуру распознать, не представляется возможным.

Учитывая указанные выше особенности, использовать описание вакансии в том виде, в каком оно предоставляется сайтом «HH.ru», для анализа требований становится весьма проблематичным. Поэтому для выявления структуры документа в данной работе использовался смешанный подход: сначала производилась разбивка документа на текстовые блоки скриптом с помощью правил, а далее выполнялась классификация каждого блока на основе его семантического анализа.

Обработка текста вакансии выполнялось в следующей последовательности:

1) предварительное исправление ошибок разметки и некорректного текста: удаление специальных символов и знаков пунктуации, лишних пробелов и разрывов строк, пустых тегов и т. п.;

2) выявление общих закономерностей в структуре документов и формирование логических правил разделения на блоки и выделения заголовков;

3) разбивка текста описания на блоки и выделение заголовков блоков с помощью методов, основанных на правилах;

4) удаление тегов HTML-разметки из блоков; окончательная чистка текста: удаление специальных символов и знаков пунктуации, повторов символов, лишних пробелов и разрывов строк;

5) удаление дублей;

6) ручная разметка семантических типов блоков;

7) обучение алгоритма классификации по размеченным данным и использование его в дальнейшем для идентификации семантической роли любого блока текста вакансии, независимо от правильности его оформления автором.

Перед разбиением на блоки выполнялась корректировка HTML-разметки. При этом удалялись все пустые парные теги и теги с ошибками.

Для исправления ошибок и очистки текста было составлено более 30 регулярных выражений.

Разбивка на блоки выполнялась на основе анализа объектной модели документа (DOM), полученной с помощью библиотеки BeautifulSoup [56]. Предположительно, требования должны размещаться в структурных элементах, оформленных в виде списков. Списками считались элементы, размеченные тегами <ul>, <ol>,
. Все остальное рассматривалось как просто текст. Текст, оформленный с помощью тегов <h> и <p> интерпретировался как заголовок.

На рисунке 2.5 представлен результат разбиения на блоки.

11 rows × 6 columns pd.DataFrame

	id	title	content	content_type	semantic_type	block_id
3	42153070	Ждем от вас	опыт взаимодействия с командой разработки;\non...	1	3	1
7	42323563	Требования	Профильное образование не ниже средне-специаль...	1	3	
10	42324314	Требования	Опыт на аналогичной должности – более 2 лет;\n...	1	3	
11	42324314	Будет плюсом, но не обязатель...	Liquibase\nTestContainers\nJUnit	1	4	
14	42352885	Что мы ожидаем от вас	Вы обладаете опытом коммерческой разработки пр...	1	3	
...	...	...	...	...	...	
1365062	15568958	Требования	креативность, работа в команде	1	3	
1365066	15569541	Что мы ожидаем от кандидата	Знание HTML5, CSS3, JavaScript;\nЗнание хотя б...	1	3	
1365070	15569605	Наши пожелания к кандидатам	Иметь опыт работы по пунктам выше\nУметь и не ...	1	3	
1365075	13279864	Нам важно	Опыт разработки на Java от 2-х;\nОпыт работы с...	1	3	
1365076	13279864	Дополнительным плюсом будет	Практические навыки работы Spring Core (знания...	1	4	

Рисунок 2.5 – Результат разбиения описаний на блоки (столбец «title» – заголовок блока, столбец «content» – его содержимое)

В результате выполненного преобразования было выделено свыше 1,3 млн. блоков, из которых был сформирован новый датасет со следующими признаками:

– id (int) – идентификатор объявления на «HH.ru»;

– title (str) – извлеченный заголовок блока;

– content (str) – содержимое текстового блока;

– content_type (int) – тип содержимого блока, возможны 4 варианта: текстовый (text=0), список (list = 1), возможно список (list_br = 2), только заголовок (block_title = 3);

– semantic_type (int) – семантический тип, возможны 6 вариантов: неизвестное назначение (unknown = 0), описание вакансии (description = 1), обязанности (responsibilities = 2), требования (requirements = 3), желательные требования (desirable = 4), условия работы (conditions = 5).

Следующим этапом была разметка датасета, в ходе которой были заполнены значения признака 'structure_type' (семантический тип блока). Для ускорения этого процесса использовались тексты заголовков блоков из признака title, которые были выделены в отдельный датасет (рисунок 2.6).

Всего было найдено свыше 50 тыс. текстов заголовков, и оказалось, преобладают: "требования", "условия" и "обязанности". Остальные заголовки являются их возможными синонимами, а также различными ошибками парсинга и плодом безумного полета фантазии их авторов.

11 rows 56062 rows × 1 columns pc

title	count
Требования	180006
Обязанности	155542
Условия	87483
Будет плюсом	26774
Мы предлагаем	26396
...	...

Рисунок 2.6 – Набор данных с заголовками блоков

Для дальнейшей обработки были выбраны только те заголовки, для которых количество блоков будет более 50. Созданный датасет был сохранен в файл и отредактирован вручную После этого специальным скриптом разметка из этого файла была перенесена в основной датасет с блоками. Таким образом было размечено свыше 800 тыс. блоков. На рисунке 2.7 представлено распределение записей по семантическому типу.

Из графика 2.7 видно, что набор данных не сбалансирован – в каждой категории представлено разное число объектов с сильным разбросом по количеству блоков. Разные классы оказались представлены не в равной степени, и при распознавании блоков, относящихся к категориям с малым количеством данных, могут возникать трудности.

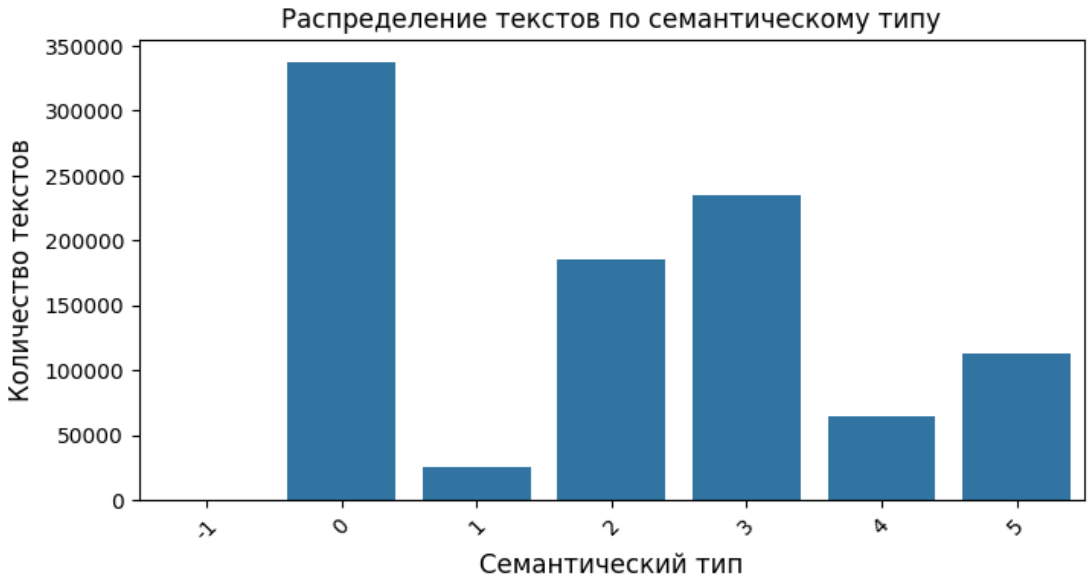


Рисунок 2.7 – Распределение размеченных блоков по семантическому типу

Наименее представлен семантический тип 1 – описание. Это следствие применявшегося метода разметки по заголовкам, т.к. разделы описаний не имеют заголовков. Такие блоки без заголовка чаще всего относятся к типу 0, который не был размечен. Размечать такое количество блоков без заголовков вручную, чтобы уравновесить группы, в рамках данной работы не представляется возможным.

Кроме того, анализ данных показал, что содержимое блоков типа 2, 3 и 4 трудно различимо, поскольку авторы объявлений используют для них одни и те же слова и выражения. Например, перечень требований к навыкам может быть в любом из этих блоков.

Для решения этих двух проблем было выполнено дополнительное преобразование.

– введена новая система классов с уменьшенным количеством категорий:

0 – описание вакансии (semantictype = 0 или 1);

1 – требования (semantictype = 2, 3 или 4);

2 – условия работы (semantictype = 5);

– для каждого класса было взято одинаковое количество объектов – 30 000 блоков.

Таким образом был получен новый датасет, содержащий 90000 строк.

Следующим этапом была предобработка текста блоков и векторизация, порядок выполнения которых был описан в главе 2.1. Предобработка блоков имела свои особенности и включала такие этапы:

1) чистка содержимого блоков;

2) соединение заголовка блока с его содержимым;

3) разбивка содержимого блока на токены по пробельным разделителям; в качестве токенов рассматривались отдельные слова;

4) приведение к нижнему регистру и лемматизация токенов;

5) обратная сборка предложений и содержимого блоков с учетом имевшихся в них знаков пунктуации и разделителей.

В результате получился следующий датасет (рисунок 2.8):

11 rows 89467 rows × 3 columns pd.DataFrame

	target_type	tokenized_block	raw_block
1	2	условие.полный рабочий день.как работа в офис ...	условия. полный рабочий день;\nкак работа в оф...
2	1	требование.знакомство с мобильный разработка п...	требования. знакомство с мобильной разработкой...
3	2	условие.комфортный офис в пешеходный доступнос...	условия. комфортный офис в пешеходной доступно...
4	0	о мы.g soft это развивающийся софтверный компа...	о нас. g-soft – это развивающаяся софтверная к...
...	...	...	...
89462	0	csssr крупный цех по производство фронтенд в р...	. csssr — крупнейший цех по производству фронт...
89463	2	мы предлагать.высокий заработный плата 210 000...	мы предлагаем. высокая заработная плата 210 00...
89464	1	задача.разработка серверный часть платёжный си...	задачи. разработка серверной части платежной с...
89465	0	мы расти и развиваться поэтому мы в команда тр...	. мы растем и развиваемся, поэтому нам в коман...
89466	2	компания предлагать.бонус за релокация.компенс...	компания предлагает. бонус за релокацию;\nкомп...

Рисунок 2.8 – Датасет, подготовленный к классификации

В полученных блоков текстов оказалось от 1 до 936 слов, в среднем 33 слова, и от 0 до 6731 символов, в среднем 274. Гистограммы распределения длин текстов блоков показаны на рисунке 2.9 (в символах) и 2.10 (в словах).

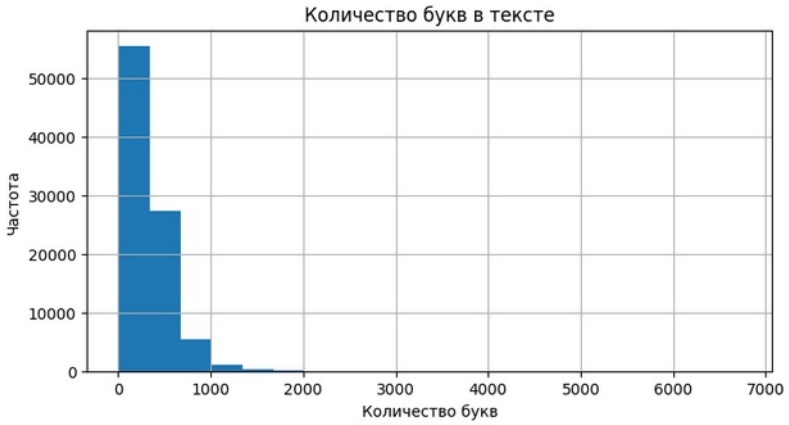


Рисунок 2.9 – Длина текстов в символах

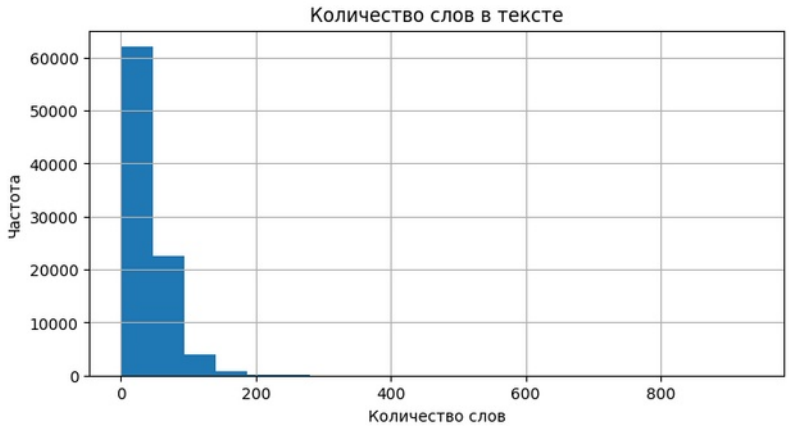


Рисунок 2.10 – Длина текстов блоков в словах

Тексты блоков достаточно длинные, поэтому для их векторизации использовались плотные векторные представления (эмбединги) для предложений, полученные с помощью библиотеки Sentence-transformers, как описано в гл. 2.1. Пример плотного векторного представления для первых двух измерений для модели «paraphrase-multilingual-mpnet-base-v2» представлен на рисунке 2.11.

Поскольку в процессе предварительной подготовки данных были получены сбалансированые классы, то в качестве основной метрики для оценки качества классификации использовалась доля правильных ответов accuracy – показатель, который описывает общую точность предсказания модели по всем классам. Он рассчитывается как отношение количества правильных прогнозов к их общему количеству. Также дополнительно рассчитывалась F1 – мера – более сложная метрика, которая учитывает баланс между точностью (precision) и полнотой (recall).

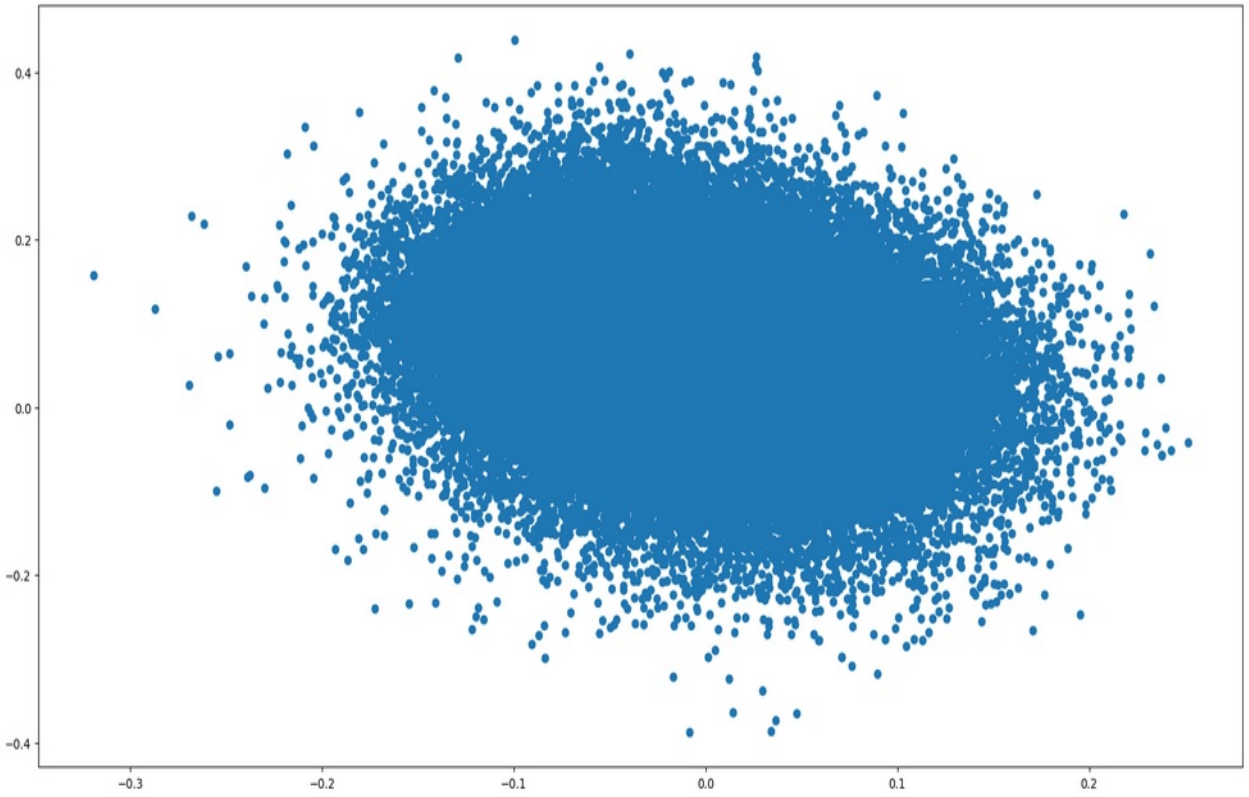


Рисунок 2.11 – Визуализация плотного векторного представления для первых двух измерений для модели «paraphrase-multilingual-mpnet-base-v2»

Для классификации использовалась логистическая регрессия. Для подбора ее гиперпараметров использовалась функция «GridSearchCV» из пакета «sklearn.model_selection». В результате были получены следующие метрики: accuracy= 0.93, f1_score= 0.93. Метрики для каждого из классов представлены в таблице 2.3.

Таблица 2.3 – Метрики качества классификации по методу логистической регрессии для классов 0, 1 и 2

Класс	Precision	Recall	F1-score
0	0.92	0.87	0.90
1	0.93	0.96	0.95
2	0.94	0.96	0.95

Лучше всего распознается блок с классом 2 – это условия работы, и это предсказуемо, поскольку в нем, как правило содержится информация, специфичная только для него.

Интересующий в данном исследовании класс 1 также хорошо распознается, и это дает основание полагать, что данную модель можно использовать для классификации блоков.

2.3 Методы и модели извлечения требований к навыкам соискателя из описаний вакансий

Требования, обязательные и необязательные, обычно содержат перечень навыков, знаний и умений, а также так называемых «софт-скиллов». Требования в вакансиях обычно задаются работодателями в свободной форме. Это значит, что в общем случае из строки, описывающей объект, идентифицировать этот объект нельзя. Поэтому при определении набора требований возникает проблема – неоднозначность трактовки, всякие непонятные формулировки, что в свою очередь осложняет проблему поиска вакансии соискателем среди тысяч других.

Для облегчения поиска подходящей вакансии соискателем предназначено поле «Ключевые навыки», которое, казалось бы, должно решить эту проблему. Однако и здесь есть сложности:

– название технологии или навыка в разных объявлениях может быть указано по разному, например: node js, node.js или даже NodeJS;

– раздел «Ключевые навыки» заполнен работодателем далеко не всегда, и чаще всего, не соответствуют тексту вакансии.

Раздел «Ключевые навыки» не является частью описания вакансии, это отдельное поле, возвращаемое API проекта HeadHunter. Поэтому в анализе текста вакансии оно не участвует, но его можно использовать, например, для составления таксономии навыков.

К указанным проблемам стоит также отнести и то, что описание требуемых навыков может быть в произвольном месте объявления: в заголовке, в описании вакансии, в перечне обязанностей или собственно в предназначенном для этого блоке требований.

Для организации эффективного поиска каждой вакансии должен быть сопоставлен однозначный список навыков, которые она требует. Соответственно, каждый навык в этом сопоставлении должен иметь одно и только одно название, даже если в тексте описания упоминаются его синонимы.

Подходы к извлечению навыков как к задаче извлечения ключевых слов с применением частотных методов и сравнения «мешка слов» (bag of words), в данном случае дают посредственные результаты. Для объектов, которым соответствует множество возможных синонимичных наименований, сначала нужно нормализовать наименования.

Для этого придётся составить базу знаний, таксономию объектов, а можно создать и целую онтологию виде графового дерева, отражающего все взаимосвязи между объектами, включая различные их трактовки. Но специфика этой области такова, что сами навыки не являются константой, по своей природе они динамичны и меняются вместе с требованиями рынка

труда. Поэтому любую таксономию или онтологию, построенную с учетом всех взаимосвязей знаний-умений-навыков, необходимо постоянно корректировать, чтобы она соответствовала запросам ее основных потребителей – участников процесса онлайн-рекрутмента.

Учитывая огромное количество объявлений, размещаемых на онлайн-площадках ежедневно, обрабатывать всю эту информацию и отслеживать тенденции рынка труда вручную – задача неподъемная. Для ее решения нужна автоматизация.

В данной работе для автоматического выделения и группирования навыков использовалась кластеризация – это объединение в группы объектов с близкими значениями их отдельных признаков. Эти группы называют кластерами. Преимуществом кластеризации является то, что обучение модели происходит без «учителя», т. е. только на входных данных. В отличие от алгоритмов с учителем, методы без учителя могут выявлять сущности на основе поиска схожих слов в документе с учетом контекста.

Перед разбиением на кластеры входные данные были подготовлены и векторизованы. В общем виде весь процесс извлечения навыков можно представить в следующем виде:

1) извлечение блочных элементов и их заголовков с помощью методов, основанных на правилах;

2) идентификация типа блочного элемента – заголовок или содержание блока, для блока дополнительно определяется наличие списка;

3) идентификация семантического типа блока с помощью классификации, формирование нового датасета, содержащего только блоки с требованиями;

4) извлечение требований из блоков с требованиями;

5) токенизация требований;

6) векторизация требований – вычисление их эмбедингов;

7) уменьшение размерности эмбедингов;

8) удаление выбросов;

9) выполнение кластеризации;

10) вычисление центров кластеров и присвоение им имен;

11) извлечение навыков, характерных для каждого кластера;

12) формирование таблицы, содержащей номер кластера, его имя, координаты центра, перечень основных навыков, относящихся к кластеру, и дополнительных навыков из поля «Ключевые навыки» всех вакансий, вошедших в кластер.

Этапы 1-3 были ранее описаны в гл. 2.2.

Извлечение требований из блоков с требованиями выполнялось в следующей последовательности:

1) разделение блока на предложения;

2) очистка предложения, удаление специальных символов, знаков препинания и т. п.

Результат выделения требований представлен на рисунке 2.12.

Всего было извлечено свыше 800 тыс. уникальных выражений, содержащих требования к знаниям и навыкам соискателей. На рисунке 2.13 показаны наиболее популярные требования.

Анализ полученных данных показал, что в текстах требований от 1 до 55 слов, в среднем 3 слова, верхнему квартилю соответствует значение 4. Аналогично, от 2 до 417 символов, в среднем 23, верхнему квартилю соответствует значение 33.

11 rows 3926029 rows × 4 columns pd.DataFrame

	id	block_id	semantic_type	list_item
4	42153070	11	3	операционные инструкции
...	...	...	...	...
3926025	13279864	7	4	Integration
3926026	13279864	7	4	Опыт работы с SOAP WEB Service
3926027	13279864	7	4	Представление о современных методологиях разра...
3926028	13279864	7	4	непрерывная интеграция
3926029	13279864	7	4	итеративная разработка

Рисунок 2.12 – Новый датасет с извлеченными требованиями

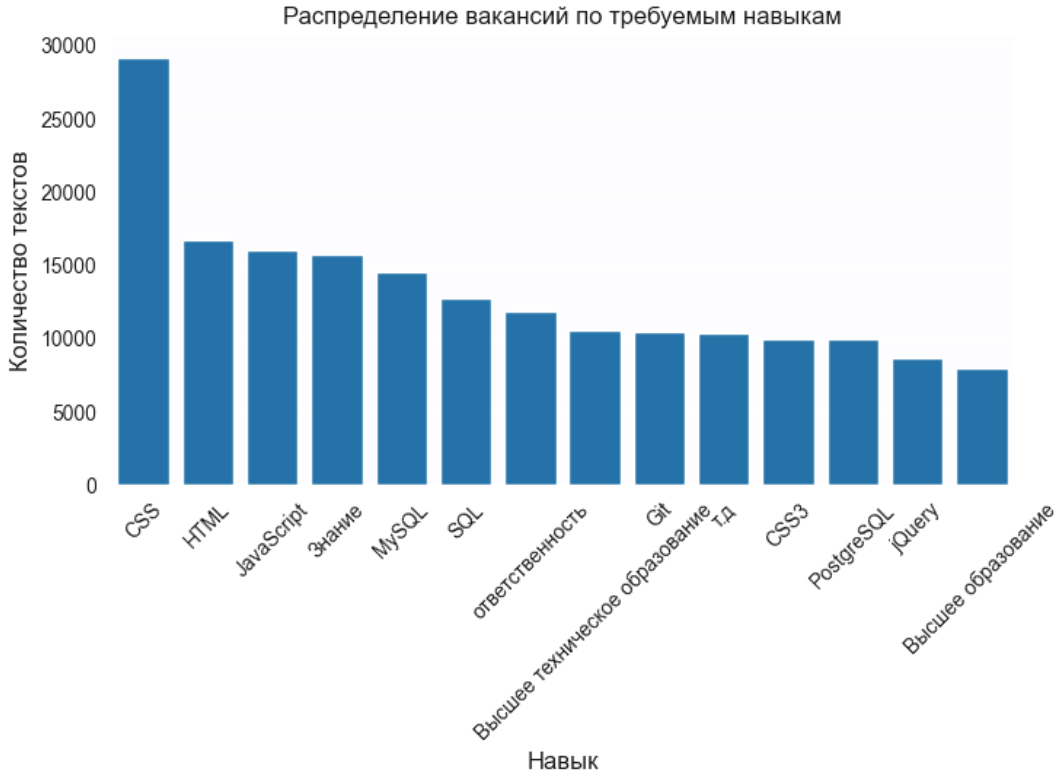


Рисунок 2.13 – Первые 14 по популярности требования к навыкам

Таким образом, среди найденных фраз больше всего униграмм, однако биграмм и триграмм также много, и в сумме их количество примерно равно количеству униграмм. Поэтому для дальнейшего анализа будут использоваться униграммы, биграммы и триграммы.

Токенизация выполнялась в следующей последовательности:

1) разделение на токены – слова с помощью метода word_tokenize из библиотеки nltk.tokenize;

2) очистка и удаление знаков пунктуации и стоп-слов;

3) приведение к нижнему регистру и лемматизация токенов;

4) создание биграмм из лемм двух соседних слов в тексте требования; триграмм – из лемм трех соседних слов.

На рисунке 2.14 представлен новый датасет с извлеченными токенами, а на рисунках 2.15– 2.17 — распределение 14 наиболее популярных из них.

10 rows × 5 columns pd.DataFrame

	count	canonical	unigrams	bigrams
опыт_взаимодействие_команда_разраб...	42	опыт взаимодействия с командой разрабо...	[опыт, взаимодействие, команда, ...	[(опыт, взаимодействие), (взаимоде
опыт_работа_реляционный_бд_sql_фор...	1	опыт работы с реляционными БД SQL для ...	[опыт, работа, реляционный, бд, s...	[(опыт, работа), (работа, реляционны
опыт_работа_функциональный_требов...	1	опыт работы с функциональными требова...	[опыт, работа, функциональный, т...	[(опыт, работа), (работа, функционал
опыт_подготовка_проектный_документ...	1	опыт подготовки проектной документаци...	[опыт, подготовка, проектный, док...	[(опыт, подготовка), (подготовка, пр
операционный_инструкция	15	операционные инструкции	[операционный, инструкция]	[(операционный, инструкция)]
знание_принцип_разработка	81	знание принципов разработки ПО	[знание, принцип, разработка]	[(знание, принцип), (принцип, разра
проектирование_приложение_часть_бд...	1	проектирования приложений в части БД п...	[проектирование, приложение, ча...	[(проектирование, приложение), (пр

Рисунок 2.14 – Результат разбивки на токены

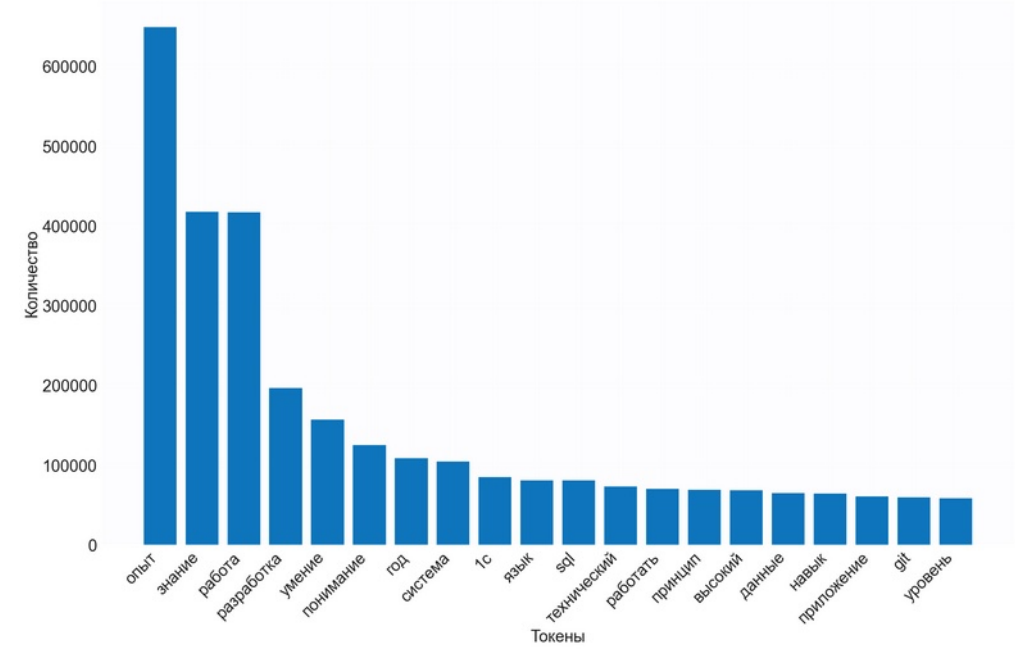


Рисунок 2.15 – Популярность униграмм

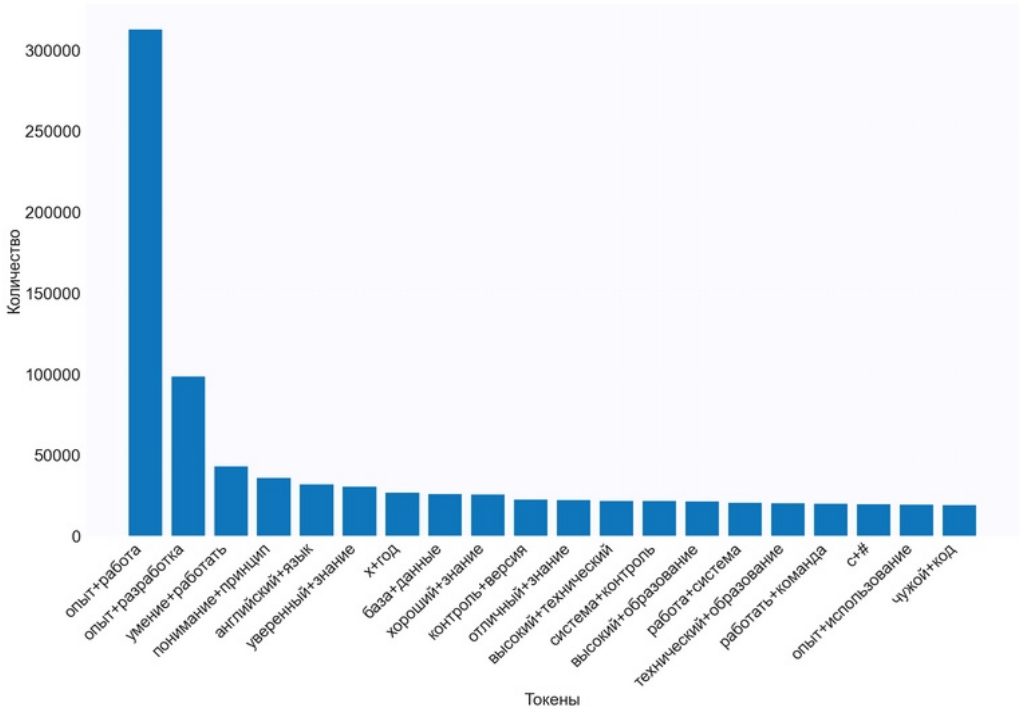


Рисунок 2.16 – Популярность биграмм

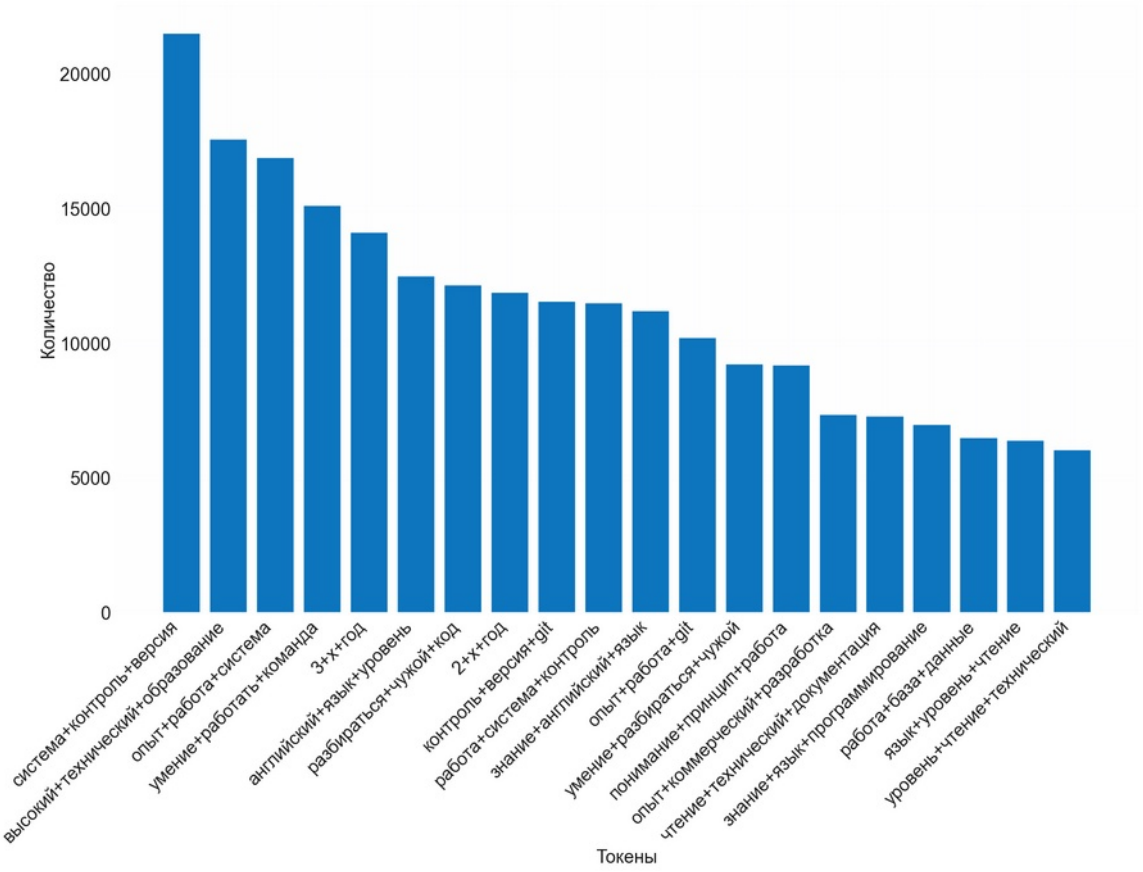


Рисунок 2.17 – Популярность триграмм

Векторизация токенов выполнялась с помощью библиотеки Sentence-transformers так, как это было ранее описано в гл.2.1. Для каждой из четырех моделей было получено свое векторное представление, и все они имеют высокую размерность (от 384 до 768 измерений).

На следующем шаге размерность векторных представлений была уменьшена с помощью метода UMAP библиотеки umap. Было взято три значения размерности: 2, 3 и 4 для возможности сравнить и выбрать лучшее.

При извлечении токенов оказалось, что различных терминов, описывающих требования к соискателю, очень много. Однако часть из них является шумом и выбросами, вследствие того, что иногда работодатели пишут в своих объявлениях пожелания, не связанные напрямую с требованиями к соискателю (выбросы), а также из-за наличия различных стоп-слов (шум). Далее, на следующем этапе, этот список был очищен и от того, и от другого.

Для удаления выбросов была использована кластеризация с помощью алгоритма DBSCAN, среди преимуществ которого:

– DBSCAN не требует спецификации числа кластеров в данных;

– DBSCAN может найти кластеры произвольной формы;

– DBSCAN имеет понятие шума и может выделять выбросы.

Однако DBSCAN хорошо работает только в пространстве объектов малой размерности. Для уменьшения размерности использовался метод UMAP.

Для DBSCAN основными гиперпараметрами на входе являются:

– радиус окрестности – eps;

– минимальное количество точек, которое может быть в одном кластере (соседей) – minpt.

Количество соседей принимается из расчета $minpt>n+1$, где n – размерность данных. Однако, если принять во внимание большой объем исходного набора данных, наличие данных с шумом и дубликатов, кластер с количеством точек меньше 20 можно считать шумом (выбрано эмпирически).

Для того, чтобы понять, какую окрестность выбрать, необходимо понять, как близко расположены точки в наборе данных. Для этого сначала рассчитаем среднее расстояние по minpt=20 ближайшим соседям для каждой точки (рисунок 2.18).

Для определения количества кластеров использовались «метод локтя».

На основе анализа графиков на рисунке 2.18 были выбраны следующие значения: для n= 2: eps=0.1, для n= 3: eps=0.2, для n= 4: eps=0.3, где n – количество измерений векторного представления.

С указанными параметрами была выполнена кластеризация алгоритмом DBSCAN. Количество полученных кластеров для каждой модели показано в таблице 2.4.

Визуализации кластеров для модели distiluse-base-multilingual-cased-v2 показаны на рисунках 2.19, 2.20 и 2.21.

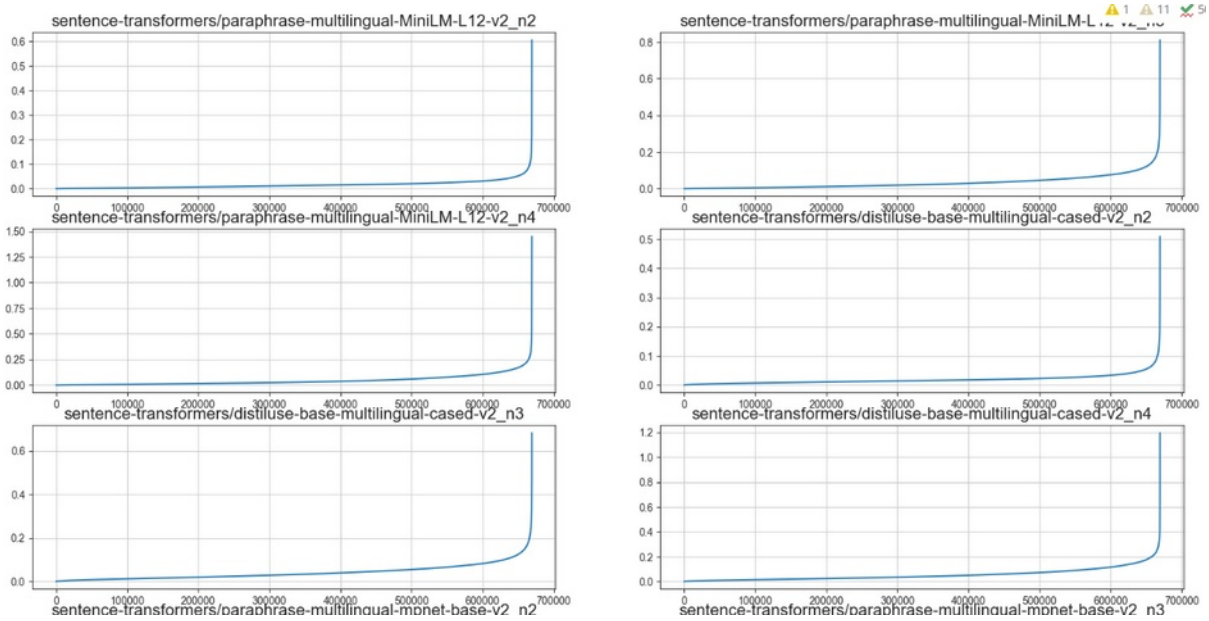


Рисунок 2.18 – Графики среднего расстояния по 20 ближайшим соседям для разных моделей векторизации и количества измерений n={2, 3, 4}

Таблица 2.4 – Результаты кластеризации алгоритмом DBSCAN

Количество измерений	Всего кластеров	Точек, попавших в шум
Модель: sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2		
2	1077	4073
3	934	3843
4	857	4533
Модель: sentence-transformers/distiluse-base-multilingual-cased-v2		
2	308	3066
3	299	2817
4	241	1444
Модель: sentence-transformers/paraphrase-multilingual-mpnet-base-v2		
2	984	4080
3	854	3548
4	689	2253
Модель: sentence-transformers/stsb-xlm-r-multilingual		
2	588	2921
3	496	2287
4	404	1635

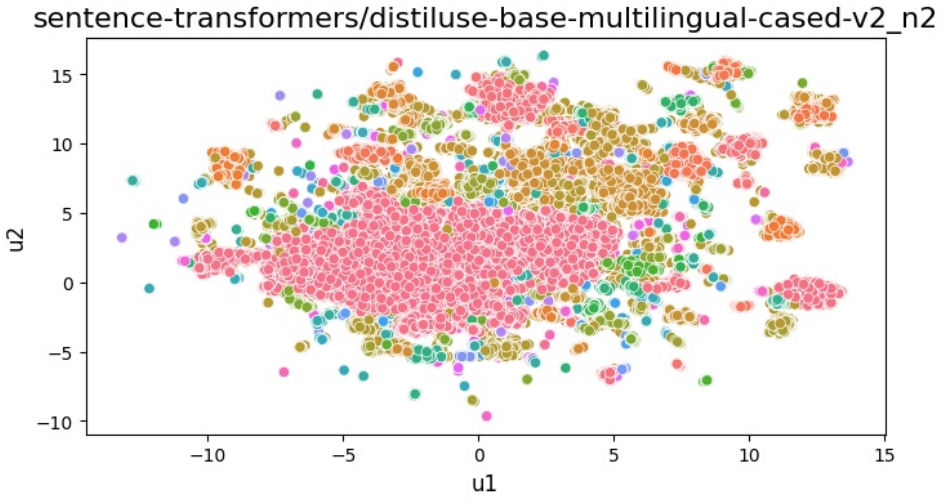


Рисунок 2.19 – Визуализация распределения кластеров для модели distiluse-base-multilingual-cased-v2, n=2

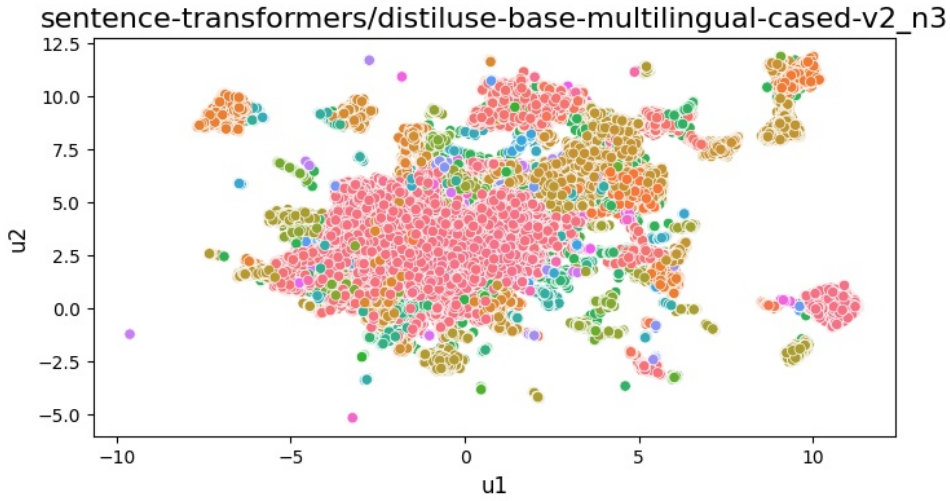


Рисунок 2.20 – Визуализация распределения кластеров для модели distiluse-base-multilingual-cased-v2, n=3

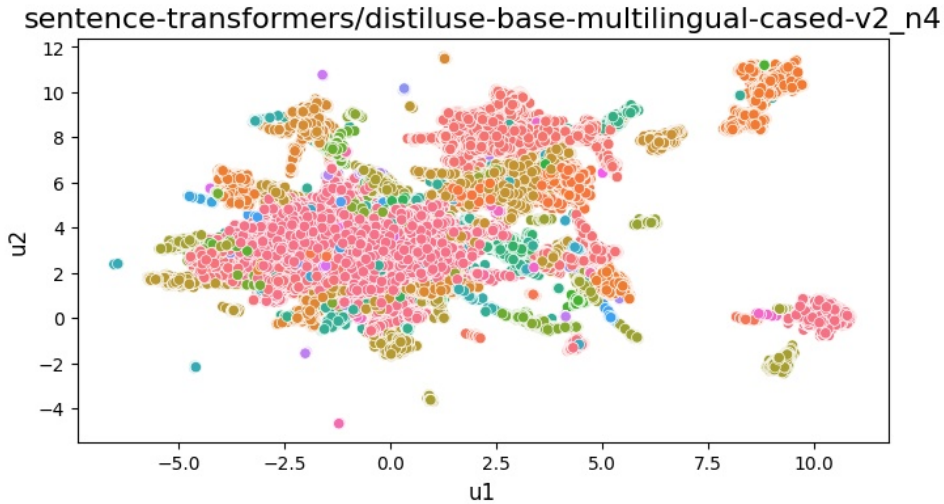


Рисунок 2.21 – Визуализация распределения кластеров для модели distiluse-base-multilingual-cased-v2, n=4

На основе анализа данных из таблицы 2.4 и графиков визуализации кластеров для очистки от выбросов выбрана модель distiluse-base-multilingual-cased-v2 с количеством измерений n=4.

Кластер с меткой «-1» содержит все точки, которые DBSCAN посчитал шумовыми – это точки, находящиеся за пределами окрестности какой-либо основной точки. Точки, попавшие в кластер «-1» удалялись полностью и автоматически.

Кластер «0» является основным кластером, содержащим большинство значений без выбросов. Нет смысла оценивать их на основе их среднего значения.

Для всех остальных кластеров использовался метод визуальной оценки содержимое каждого кластера. Чтобы оценить результаты кластеризации, были сопоставлены 4 первых и 4 последних требования в отсортированном списке для каждого кластера. На основании результатов оценки было принято решение для каждого кластера, сохранить его или нет.

Оценка выполнялась на основе евклидова расстояния относительно среднего арифметического всех точек в кластере (рисунок 2.22).

В результате было удалено еще 15 кластеров.

Таким образом, использование алгоритмов уменьшения размерности UMAP и кластеризации DBSCAN позволили выполнить очистку данных и удалить большинство выбросов.

203	20	Знание методологии тестирования с опытом работы тестировщиком г Опыт работы тестировщиком программного обеспечения от 2х лет (0. опыт тестирования программных продуктов от 2 х лет (0.0009) Опыт работы в сфере тестирования программного обеспечения от 2 х	Опыт ручного тестирования программного обеспечения от 2 лет (0.0017) Опыт работы в тестировании программных продуктов от 2х лет (0.0020) опыт работы в области тестирования или контроля качества программног опыт тестирования от 2 лет или программиста от 1 года (0.0133)
204	83	Опыт работы с Facebook (0.1243) Опыт общения с Facebook (0.1245) Опыт закупки трафика Facebook (0.1307) Опыт создания плееблов для Facebook (0.1333)	Опыт интеграции SDK событий с Фейсбук (0.6456) Опыт работы с SDK социальных сетей FB (0.6546) Опыт интеграции различных SDK facebook (0.6961) Опыт интеграции различных SDK FB SDK (0.9690)
205	113	вывода показателей на портал (0.0122) Наличие в портфолио порталов (0.0190) используемые для изготовления дверей (0.0209) агентский портал (0.0223)	понимание работы с портами (0.2499) работа с соответствующими порталами Минздрава (0.2578) разработки высоконагруженных корпоративных порталов (0.3173) работы с коммуникационными портами (0.3523)
206	143	Отличное понимание js DOM (0.0093) Понимание работы DOM (0.0097) понимание DOM (0.0098) глубокое понимание DOM management (0.0101)	Опыт работы с DOM API (0.2189) Опыт работы с DOM при помощи js нативный js (0.2308) DOMXPath (0.2452) DOM Level 0 (0.3031)
207	47	Умение применять Dependency Injection Dagger (0.0206) Использование любого dependency injection framework (0.0223) Умение работать с dependency injection (0.0240) знакомство с dependency injection (0.0256)	использование на практике концепции Dependency Injection (0.1285) deprndency injection di (0.1383) dll injection (0.2064) Zeniect или любой другой Dependency Injection method (0.3832)

Рисунок 2.22 – ручное оценивание кластеров по первым и последним элементам; в скобках – евклидово расстояние до центра кластера

Следующим этапом была основная кластеризация, которая позволит сгруппировать требования к навыкам по их семантическим признакам.

Для кластеризации было опробовано несколько методов из Python библиотеки Scikit-learn: KMeans, Agglomerative Clustering, OPTICS.

По критериям качества и быстроте кластеризации в качестве алгоритма кластеризации был выбран метод k-средних (также известный как k-means). Работа алгоритма основана на минимизации отклонения точек от центроидов (центров кластеров). Метод k-средних хорошо работает в случае векторных представлений, поскольку он основан на кластеризации точек по расстоянию между ними. Однако особенностью данного алгоритма является то, что число кластеров k является входным параметром, которое необходимо определить заранее.

Для выбора оптимального числа кластеров использовалась метрика – индекс силуэта (Silhouette score). Анализ силуэта относится к методу проверки согласованности внутри кластеров данных. Значение силуэта является мерой того, насколько объект похож на свой собственный кластер (сплоченность) по сравнению с другиими кластерами (разделение). Его можно использовать для изучения расстояния разделения между полученными кластерами. График силуэта отображает степень близости каждой точки в одном кластере к точкам в соседних кластерах и, таким образом, предоставляет способ визуальной оценки таких параметров, как количество кластеров.

Метод проверки силуэта вычисляет индекс силуэта для каждой выборки, средний индекс силуэта для каждого кластера и общий средний индекс силуэта для набора данных.

Если значение индекса силуэта велико, объект хорошо соответствует своему собственному кластеру и плохо соответствует соседним кластерам.

Если значение силуэта близко к 1, выборка хорошо кластеризована.

Если значение силуэта примерно равно 0, выборка может быть назначена другому ближайшему к нему кластеру, и выборка находится

одинаково далеко от обоих кластеров. Это означает, что она указывает на перекрывающиеся кластеры.

Если значение силуэта близко к -1, образец неправильно классифицирован и просто помещен где-то между кластерами.

В данной работе индекс силуэта вычислялся для каждой из моделей векторизации с целью выбрать наиболее подходящую из них, а также определить оптимальное число кластеров (рисунки 2.23 и 2.24).

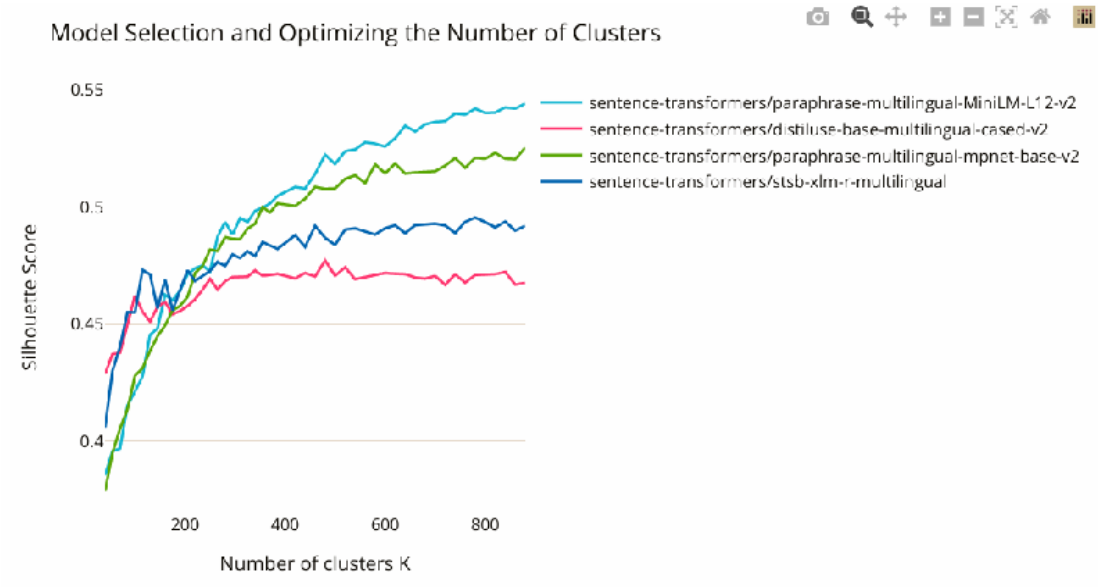


Рисунок 2.23 – Индекс силуэта для кластеризации по методу Kmeans для разных моделей векторизации

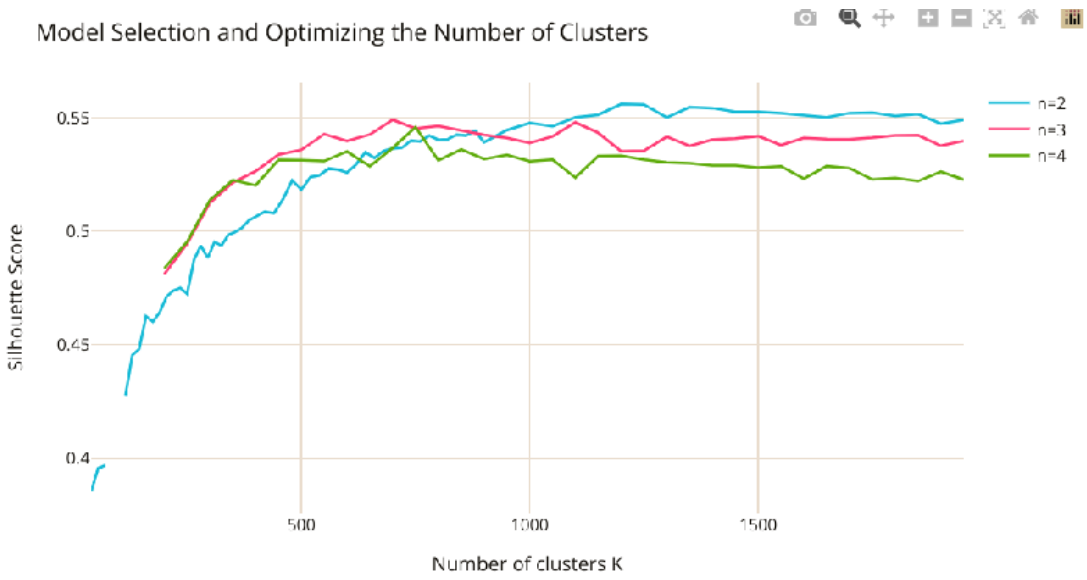


Рисунок 2.24 – Индекс силуэта для кластеризации по методу Kmeans для модели paraphrase-multilingual-MiniLM-L12-v2 при разном числе измерений n

Для оценки силуэта использовались данные с уменьшенной размерностью эмбедингов.

Анализ графика 2.23 показывает, что с увеличением числа кластеров индекс силуэта растет, и качество кластеризации также возрастает. Лучше всего показала себя модель «sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2», кроме того, в ней используется самый коротки вектор длиной 384. Поэтому выбор очевиден в пользу модели №1 – «sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2».

Для определения оптимального числа измерений был построен еще один график (рисунок 2.24), из которого видно, что оптимальные параметры кластеризации: количество измерений n=2, количество кластеров K =1200.

На рисунке 2.25 показана визуализация кластеризации, выполненной с данными параметрами, а на рис. 2.26 – распределение текстов по кластерам.

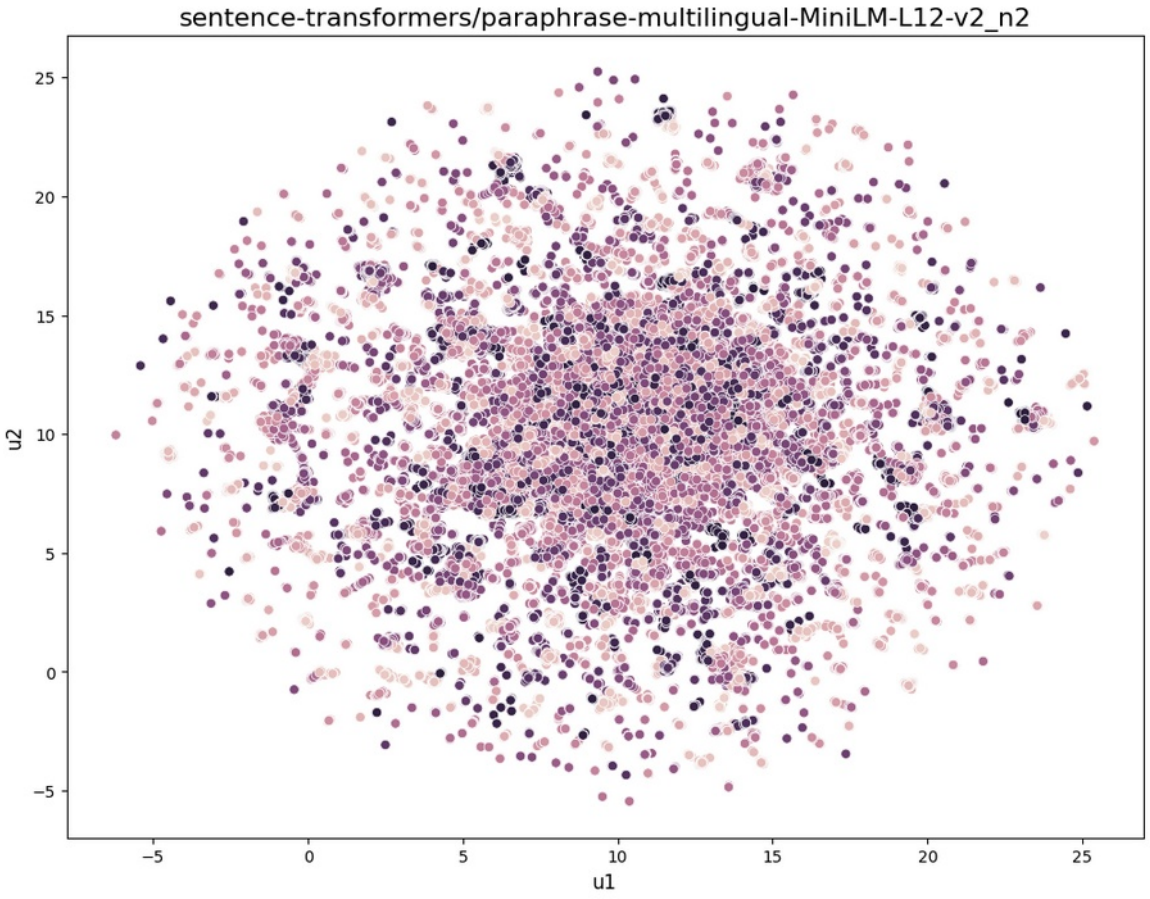


Рисунок 2.25 – Визуализация кластеризации по методу Kmeans для числа кластеров K=1200 и числом измерений векторного представления текста n=2

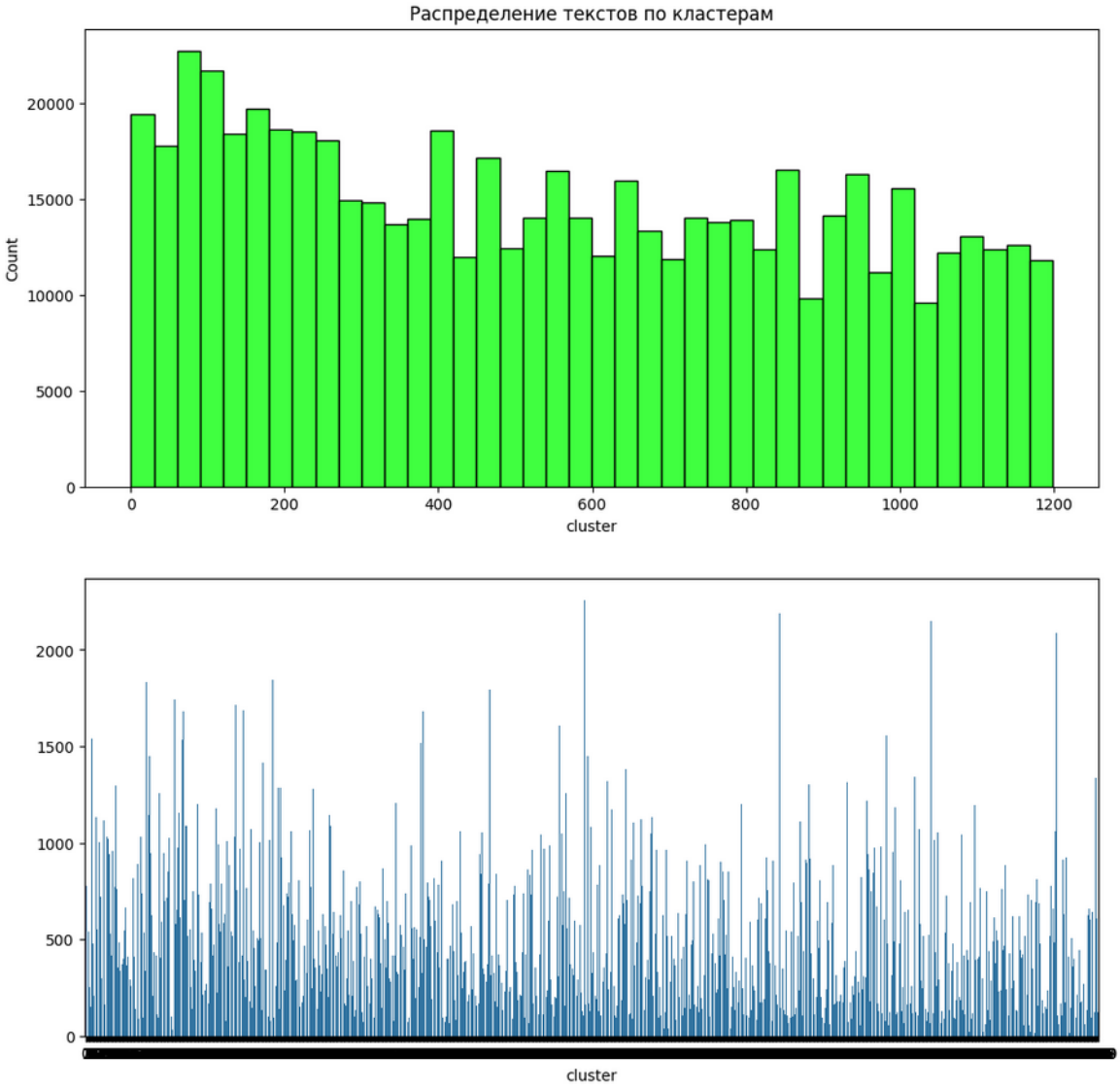


Рисунок 2.26 – Распределение текстов по кластерам

Вычисление центров кластеров выполнялось как среднее арифметическое координат входящих в него элементов.

Заключительным этапом работы было получение списка наиболее популярных в кластере униграмм, биграмм и триграмм. Самая популярная униграмма или биграмма становится именем кластера.

Результатом всей работы стало формирование таблицы, содержащей номер кластера, его имя, координаты центра, перечень основных навыков, относящихся к кластеру, и дополнительных навыков из поля «Ключевые навыки» всех вакансий, вошедших в кластер.

2.4. Выводы по главе 2

В результате проведенных исследований было установлено:

1. Несмотря на огромную коллекцию текстовых документов в архиве объявлений портала по трудойстройству, не менее трети из них являются дублями. Поэтому перед любой обработкой таких данных необходима чистка датасета от повторных объявлений.

2. Выделение структуры в текстовом описании вакансий весьма сложная задача из-за того, что авторы объявлений не придерживаются общих правил при их составлении. Методы, основанные на правилах, не могут полностью решить эту проблему.

3. Классификация с помощью алгоритма логистической регрессии является эффективным методом определения семантического типа разделов описания вакансии.

4. Кластеризация отдельных выражений, представляющих собой формулировки требований, является более эффективным методом извлечения требований к навыкам, чем кластеризация объявлений целиком, и позволяет группировать навыки по смыслу выполняемой функции, а не по виду профессиональной деятельности.

5. Применение трансформеров является эффективным методом векторизации текстов, обеспечивая более высокие показатели качества при кластеризации.

3 Проектирование цифрового сервиса извлечения структурированной информации из текстов вакансий

3.1 Архитектура приложения агрегатора вакансий

Веб-приложение агрегатора вакансий выполняет функции автоматизированного сбора данных с сайтов трудоустройства, их обработку, хранение и поиск. Обработанные данные становятся основой для комплекса математических моделей для автоматической структуризации текстов объявлений.

На основе анализа предметной области в главе 1 и разработанной модели извлечения структурированной информации из текста вакансии в главе 2 была разработана диаграмма потоков данных для приложения агрегатора вакансий (рисунок 3.1).

Основная внешняя сущность системы – это соискатель, который отправляет запрос в систему и запускает процедуру поиска объявлений в соответствии с указанными им параметрами (рисунок 3.1). Выходными данными являются список найденных вакансий.

Всю деятельность системы можно разбить на три основных процесса (см. рисунок 3.1)

1. Процесс подбора вакансий под запрос соискателя

Инициируется пользователем системы.

2. Процесс парсинга сайтов-агрегаторов объявлений и последующей обработки новых вакансий

Процесс инициируется модулем планирования задач операционной системы (cron) или любым другим сервисом, предназначенным для этого. В процессе обработки вакансий происходит извлечение структурированной информации из их описания. Результатом этой деятельности является обновление базы данных вакансий.

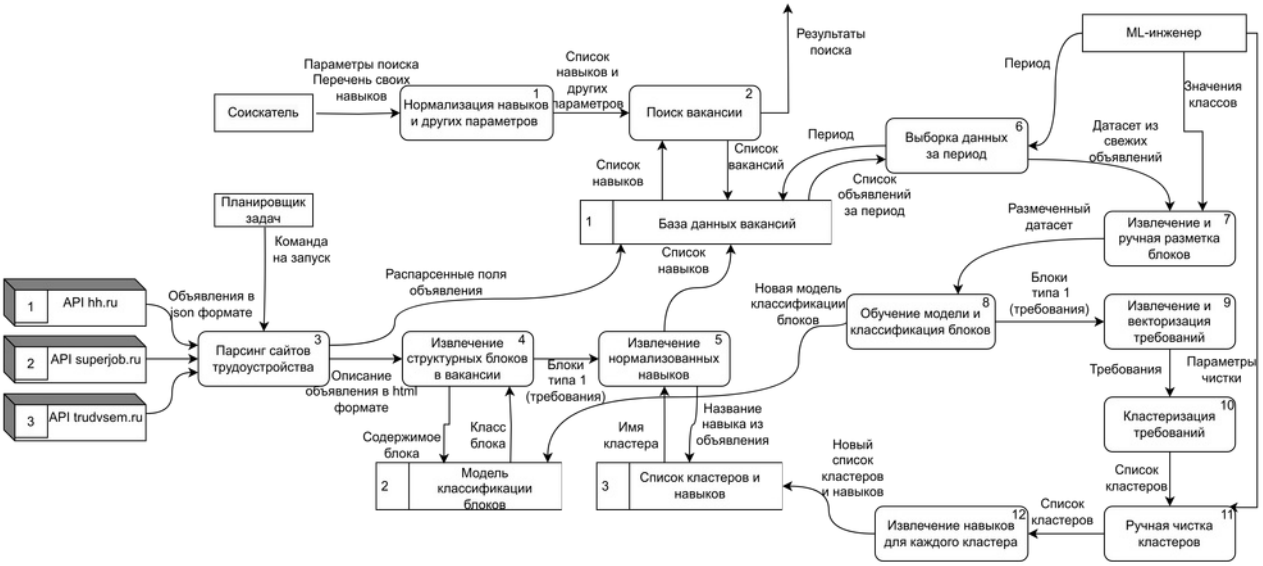


Рисунок 3.1 – Диаграмма потоков данных информационно-поисковой системы

3. Обучение и обновление моделей

Инициируется ML-инженером. Начинается с выборки из базы данных объявлений за определенный период и преобразования ее в удобный для анализа формат в виде датасета. В распоряжении ML-инженера готовый набор программ и сервисов для обработки этого датасета, от него требуется активное участие только на двух этапах : разметка типов блоков и чистка кластеров. Результатом этой деятельности является обновление модели классификации и списка кластеров с ключевыми словами.

Пример реализации представленной диаграммы потоков показан на рисунке 3.2. Программный продукт должен иметь микросервисную архитектуру.

Сервера на рисунке 3.2 показаны условно, поскольку все компоненты будут разворачиваться как веб-сервисы в контейнерах на облачной платформе (кроме компьютеров пользователей и внешних сервисов типа api.hh.ru).

Для организации серверов баз данных подходят любые реляционные или NoSQL серверные СУБД, а для хранилищ данных и артефактов – любые файловые сервера, доступные на большинстве облачных платформ.

Основная полезная нагрузка приложения на рисунке 3.2 сосредоточена внутри Product сервера. Он отвечает за функционирование трех основных сервисов приложения:

1. Каталог вакансий

Предоставляет методы работы с объявлениями для соискателей; принимает на вход список параметров, на выходе формирует список вакансий; также отвечает за предоставление подробной информации по любой из вакансий.

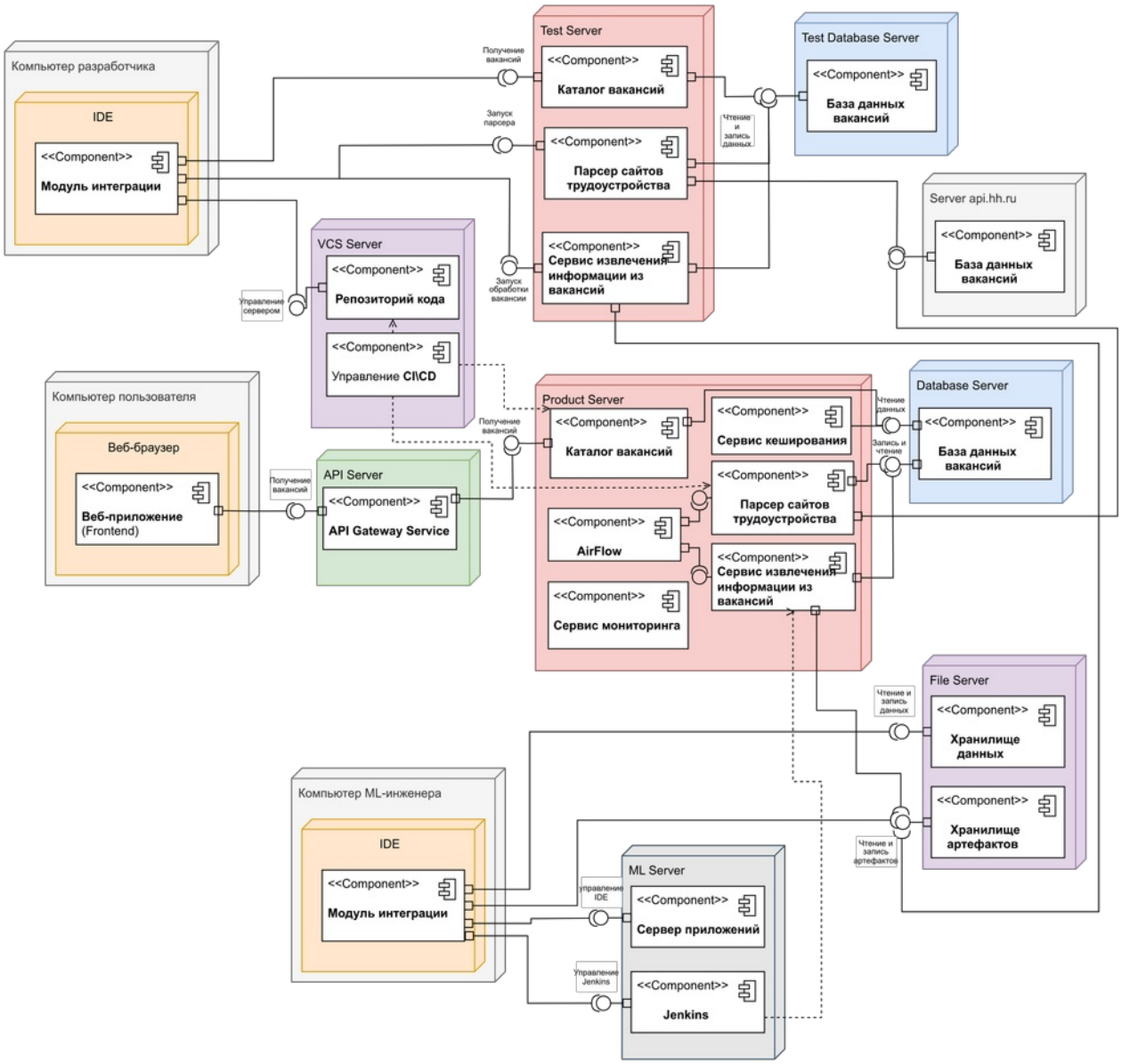


Рисунок 3.2 – Диаграмма развертывания информационно-поисковой системы

2. Парсер сайтов трудоустройства

Отвечает за получение и первичную обработку данных из внешних источников; предоставляет 2 метода: получение списка обновлений и загрузка объявлений с удаленного внешнего сервера.

3. Сервис извлечения информации из вакансий

Отвечает за обработку текстов вакансий и извлечение полезной информации из них, а также сохранение ее в базе данных. Предоставляет всего один метод, запускающий процесс обработки.

Каждый сервис в системе имеет свой HTTP REST JSON API, с помощью которого он взаимодействует с другими сервисами и с пользователями.

Успешное функционирование системы невозможно без подсистем мониторинга, кэширования, проксирования, а также централизованного управления кодом и развертыванием. Для этих задач предполагается использование готовых решений.

В рамках данной работы был реализован только сервис извлечения информации из вакансий.

3.2 Описание и принцип работы сервиса извлечения структурированной информации о требованиях к соискателю

На основе модели информационной системы, представленной на рисунке 3.1, и ее диаграммы на рисунке 3.2, можно определить следующие основные требования к функциям прототипа веб-сервиса извлечения структурированной информации о требованиях к соискателю:

1. Веб-сервис должен обеспечивать извлечение информации о структуре объявления в виде списка текстовых блоков, их типов и заголовков.

2. Веб-сервис должен обеспечивать извлечение информации о требуемых навыках, знаниях и умениях в виде списка.

3. Веб-сервис должен принимать на вход либо файл с описанием вакансии с HTML разметкой, либо сообщение с описанием вакансии с HTML

разметкой в формате JSON.

4. На выходе веб-сервис должен формировать сообщение в формате JSON, содержащую структурированную информацию, извлеченную из текста вакансии.

5. Веб-сервис должен выдерживать нагрузку, необходимую для обеспечения бесперебойной работы агрегатора вакансий, из расчета 300 000 запросов/сут или 3,48 RPS.

6. Для взаимодействия с веб-сервисом должен быть реализован REST API интерфейс поверх HTTP/HTTPS.

В соответствии с этими требованиями был разработан веб-сервис, API которого содержит два метода, запускающие процесс извлечения информации из текста вакансии. Методы отличаются только способом передачи текста вакансии на вход алгоритма.

В общем виде алгоритм извлечения информации представлен на рисунке 3.3.

На вход алгоритм принимает:

– описание вакансии с HTML разметкой;

– предобученные модели векторизации и кластеризации;

– список кластеров, для каждого из которых указано имя кластера, координаты центра кластера, список наиболее значимых требований в нормальной форме, максимальный радиус кластера;

– список навыков и требований, собранных по всем объявлениям.

На выходе алгоритма – структурированная информация о вакансии, включающая в себя:

– список разделов с указанием названия и типа раздела;

– список основных требований;

– список дополнительных требований.

Рисунок 3.3. – Алгоритм извлечения структурированной информации о требованиях к соискателю из описания вакансии

На основе описанного подхода был разработан прототип сервиса. Реализация серверной части осуществлена на языке программирования Python. Она включает в себя модули, которые обеспечивают:

– запуск сервера веб-приложения;

– обработку HTTP-запросов со стороны клиента;

– формирование и вывод JSON-ответа.

Сервис запускается командой:

```
uvicorn api:app
```

Вид мини UI для вебсервиса представлен на рисунках 3.4 – 3.6.

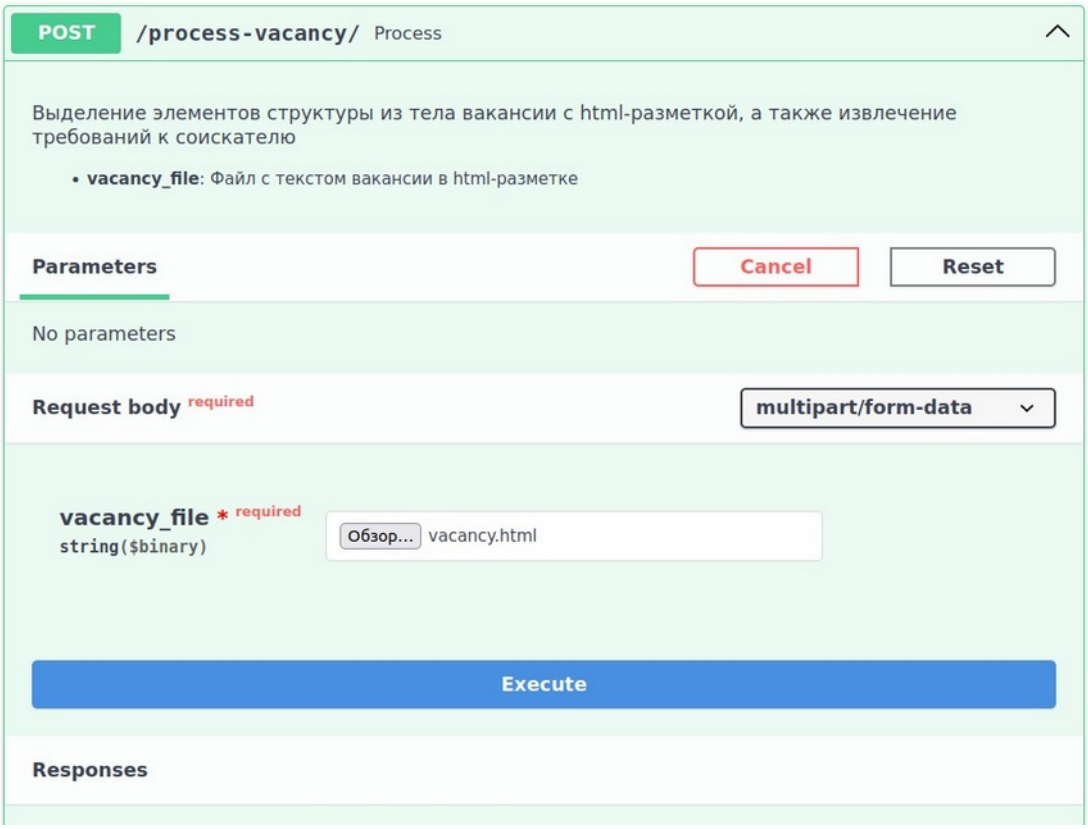


Рисунок 3.4 – Отправка запроса сервису

Server response

Code | Details

200

Response body


```
{
  "is_success": true,
  "message": null,
  "key_skills": [
    "rest api",
    "компания",
    "управление",
    "программный обеспечение",
    "проектный документация",
    "grpc",
    "docker compose",
    "код",
    "frameworks",
    "язык программирование",
    "rxjava",
    "kotlin",
    "k8s",
    "redux",
    "модель данные",
    "apache kafka",
    "плюс",
    "rust",
    "hibernate",
    "пк",
    "тестирование",
    "mvvm",
    "clickhouse",
```


Download

Рисунок 3.5 – Ответ со списком ключевых навыков

Server response

Code | Details

200

Response body


```
  "content": [
    {
      "block_id": 1,
      "id_hh": 0,
      "title": "",
      "content": "Наша компания разрабатывает аналитические решения для различных инд
устрий: системы предсказания спроса на товары для оффлайн-ритейлов, рекомендательные
системы в банках, поисковые системы по товарам для онлайн-ритейлеров и многое другое.
Мы активно развиваемся как в России (среди клиентов Сбербанк, Пятёрочка, KFC, Перекрё
сток, Альфа Банк, МВидео и др.), так и за рубежом. У нас молодой, дружный и опытный к
оллектив, многие сотрудники - выпускники МГУ, МФТИ, ВШЭ, Сколтеха, ШАД Яндекса, бывши
е сотрудники Яндекса, Гугла, Самсунга, Тинькова, Рамблера (однако всё это, разумеетс
я, не является ограничителем для кандидатов – мы смотрим на навыки, а не на регалии",
      "content_type": 0,
      "semantic_type": 0
    },
    {
      "block_id": 2,
      "id_hh": 0,
      "title": "",
      "content": "Мы ищем Java Developer уровня middle и выше для усиления нашего нап
равления разработки внутреннего фреймворка, позволяющего сокращать затраты на разрабо
тку проектов и вести их более стандартизировано.\nЭто работа, которая потребует макси
мального совмещения технической экспертизы и творческого мышления, надо постоянно дум
ать о том, как сделать работу других разработчиков проще и приятнее.\nПомимо участия
в разработке ядра фреймворка, мы предлагаем периодическое привлечени
угих проектов по принципу ротации - важно непосредственно на себе
```


Download

Рисунок 3.6 – Ответ со списков структурных блоков текста вакансии

Был сделан простой веб-интерфейс, в котором можно загрузить вакансию в формате HTML в виде обычного текстового файла и получить её структурированное представление в формате JSON.

Для отправки данных на сервер используются HTTP POST-запросы, в теле которых в формате JSON передается текст вакансии. Для енд-пойнта «/process-vacancy-file» через параметр «vacancy_file» передается файл, для енд-пойнта «/process-vacancy» - в JSON-параметре «text» передается сериализованный текст описания.

Ответ сервиса имеет следующий вид:

```
{
  "is_success": false,
  "message": "string",
  "key_skills": [
    "string"
  ],
  "add_skills": [
    "string"
  ],
  "content": [
    {
      "block_id": 0,
      "id_hh": 0,
      "title": "string",
      "content": "string",
      "content_type": 0,
      "semantic_type": 0
    }
  ]
}
```

Ответ содержит следующие поля:

– key_skills: ключевые навыки;

– add_skills: дополнительные навыки;

– content: раздел, содержащий структурированное представление текста вакансии, включающее в себя:

- id_hh – идентификатор вакансии на сайте «HH.ru»;

- title – заголовок блока;
- content – текстовое содержание блока;
- content_type – тип содержимого, возможны варианты:

0: текстовый (text); 1: список (list); 2: возможно список (list_br), в случае, когда элементы списка формируются с помощью не предназначенных для этого тегов, например
; 3: заголовок (block_title), в случае если не удалось обнаружить содержимое блока;

- semantic_type – семантический тип блока, возможны варианты:

0: описание вакансии (description); 1: обязанности, требования и желательные требования (responsibilities, requirements); 2: условия работы (conditions);

- block_id – идентификатор блока в тексте вакансии.

Пример ответа сервиса:


```
{
  "is_success": true,
  "message": null,
"key_skills": [
    "redux",  "clickhouse",  "mvvm", "hibernate", ... ],
  "add_skills": [
    "качество",  "redis",  "spring", "java", … ],
  "content": [
    {
      "block_id": 1,
      "id_hh": 0,
      "title": "",
      "content": "Наша компания разрабатывает ...",
      "content_type": 0,
      "semantic_type": 0
    },
    {
      "block_id": 4,
      "id_hh": 0,
      "title": "Задачи",
      "content": "Разработка внутренних инструментов ...",
      "content_type": 1,
      "semantic_type": 1
    },
```

```
    {
      "block_id": 6,
      "id_hh": 0,
      "title": "Требования",
      "content": "Опыт разработки на Java/Kotlin от 2-х
лет;...",
      "content_type": 1,
      "semantic_type": 1
    }
  ]
}
```


API сервиса использует следующие коды ответа:

200 – запрос выполнен успешно;

400 – ошибка при формировании запроса со стороны клиента – переданный текст описания не имеет содержимого или он не имеет HTMLразметки;

422 – неверный тип переданных данных (ошибка сериализации);

500 – внутренняя ошибка сервера.

Распознавание семантического типа блока выполняется с помощью методов машинного обучения, а точнее – классификации на основе логистической регрессии.

К сожалению, ввиду имеющейся неоднозначности содержимого блоков и определенного нежелания работодателей придерживаться определенных стандартов при создании и размещении объявлений о вакансии, и как следствие, отсутствие общепринятой структуры объявлений, распознать более детально тип блока средствами машинного обучения не удалось.

Сервис реализован на языке Python, и использует такие библиотеки как pymorphy3, nltk, torch,beautifulsoup4, requests, pandas, numpy, fastapi, SentenceTransformer, umap, sklearn.

Используется файловое хранилище данных.

Исходный код проекта размещен в открытом репозитории по адресу https://github.com/svwk/vkr.

Сервис использует предварительно обученные модели, и для своей корректной работы требует периодического обновления этих моделей.

Для проверки корректности работы сервиса выполнено функциональное и интеграционное тестирование, результаты представлены на рисунке 3.7. Пример ответа сервера при неправильных переданных данных представлен на рисунке 3.8.

Для проверки возможности обеспечивать заданную нагрузку (3,48 RPS) было выполнено нагрузочное тестирование, результаты которого представлены на рисунке 3.9.

Нагрузочное тестирование показало, что сервис может обеспечивать пропускную способность в 7,3 RPS, что укладывается в требования к веб-сервису.

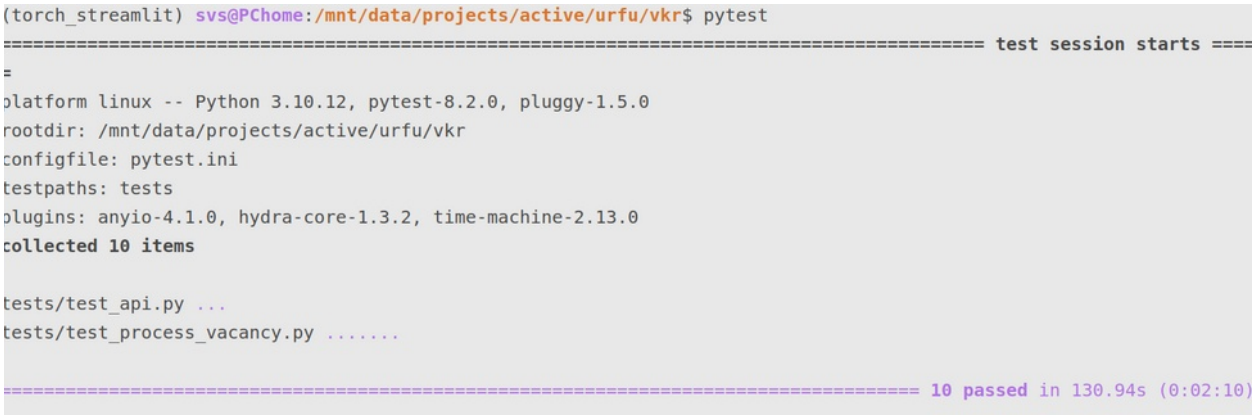


Рисунок 3.7 – Результаты интеграционного и функционального тестирования

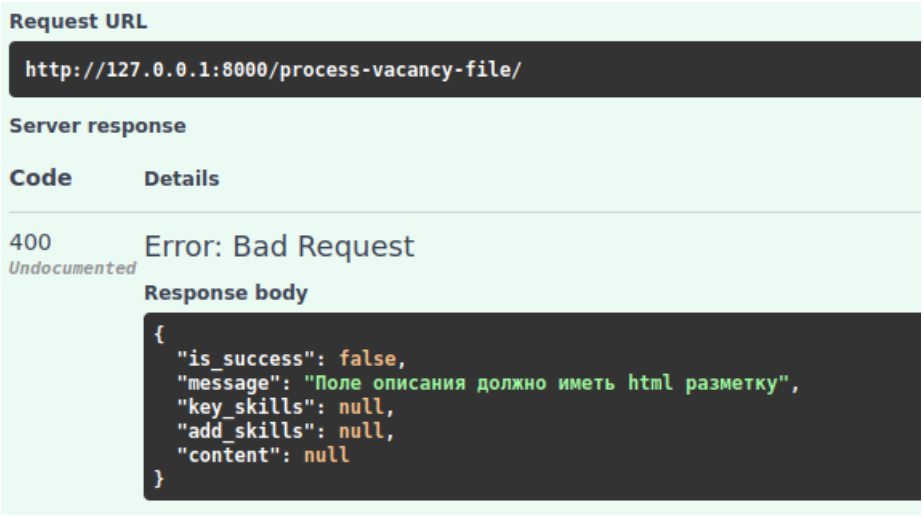


Рисунок 3.8 — Пример ответа при неверно переданных данных

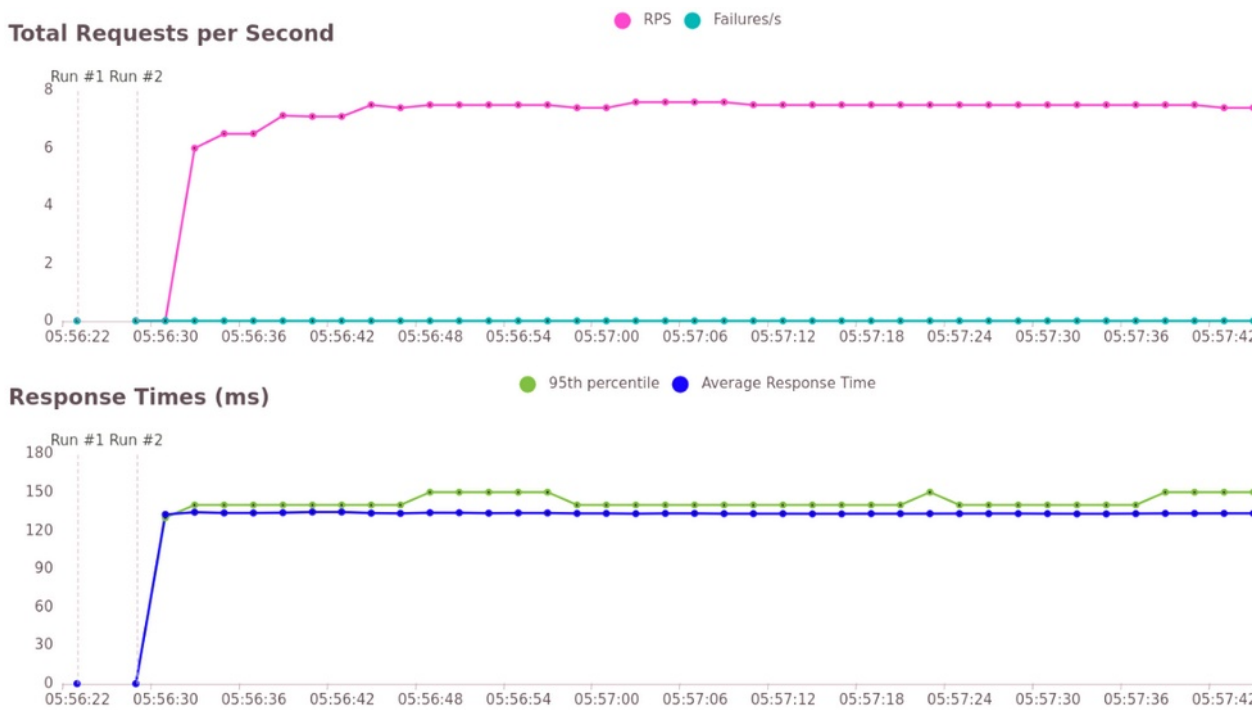


Рисунок 3.9 -Результаты нагрузочного тестирования веб-сервиса

Таким образом, результаты тестирования показали, что требования, сформулированные для разработки веб-сервиса, выполнены.

3.3. Выводы по главе 3

1. Была разработана архитектура приложения, которая использует методы и модели технологий естественного языка для обработки текста вакансий.
2. Использование микросервисной архитектуры позволяет эффективно разделить задачи обработки входящих веб-запросов, сбора данных, обработки данных средствами машинного обучения, а также обучения моделей.
3. Был разработан прототип веб-сервиса, который успешно справляется с задачей структуризации текста вакансии, которую он получает посредством HTTP-запроса.
4. Результат работы сервиса — информация о вакансии в JSON формате.

ЗАКЛЮЧЕНИЕ

В результате проведенных исследований можно сделать следующие выводы:

1. Доказано, что для сайтов по поиску вакансий предпочтительно использовать не полнотекстовый, а фактографический поиск, в котором атрибуты вакансий должны быть основой для их поиска.

2. Показано, что информационно-поисковая система, используемая на сайте по подбору персонала, должна предлагать возможные требуемые навыки из заранее заданного списка при поиске вакансий на определенную должность или специализацию.

3. Показано, что для реализации фактографического поиска вакансий требуется выполнить структуризацию текста вакансий с извлечением навыков и записью их в соответствующие атрибуты документа. Установлено, что лучшим методом векторизации текстов вакансий являются трансформеры, а для извлечения информации в условиях постоянно меняющегося рынка труда является кластеризация.

4. Выявлено, что классификация с помощью алгоритма логистической регрессии является эффективным методом определения семантического типа разделов описания вакансии.

5. Кластеризация отдельных выражений, представляющих собой формулировки требований, является более эффективным методом извлечения требований к навыкам, чем кластеризация объявлений целиком, и позволяет группировать навыки по смыслу выполняемой функции, а не по виду профессиональной деятельности.

6. Установлено, что извлечение информации из текстов объявлений позволяет организовать более эффективный способ поиска вакансий. Методы машинного обучения и технологии обработки естественного языка позволяют изменить подход к организации поиска текстовых документов, открывают возможность использования для хранения более быстрых реляционных баз

данных и могут стать обязательным элементом таких систем.

7. Установлено, что методы машинного обучения и технологии обработки естественного языка могут эффективно решать задачи структуризации текстов вакансий и извлечения из них информации о требуемых навыках.

8. Доказано, что библиотеки для машинного обучения на языке Python могут использоваться в веб-приложениях с различным стеком технологий благодаря возможности выделения их в отдельный веб-сервис при использовании микросервисной архитектуры.

9. Установлено, что язык программирования Python плохо подходит для создания высоконагруженных сервисов и для обработки больших массивов данных, особенно с использованием методов машинного обучения.

10. Разработана архитектура веб-приложения для сбора данных с сайтов трудоустройства и подбора трудовых вакансий с использованием структурированной информации о них.

11. Реализован веб-сервис, выполняющий структуризацию текстов вакансий и извлечение из них информации о требуемых навыках методами машинного обучения с использованием технологий обработки естественного языка, и предоставляющий возможность интеграции с ним посредством RESTful API интерфейса. Производительность веб-сервиса достаточна для работы в рамках одного приложения с микросервисной архитектурой.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Куприянов, А. Н. Развитие рекрутинга на рынке труда в России : специальность 08.00.05 «Экономика и управление народным хозяйством: экономика труда»: автореферат на соискание ученой степени кандидата экономических наук / Куприянов Антон Николаевич ; Российская академия государственной службы при Президенте Российской Федерации. – Москва, 2011. – 26 с. – Текст : непосредственный.

2. Таштамирова, Д. С. Представления о рынке труда и роль операторов рынка труда в формировании структуры занятости населения : специальность 22.00.03 «Экономическая социология и демография» : автореферат на соискание ученой степени кандидата социологических наук / Таштамирова Дина Саламовна ; ГОУВПО «Хабаровский государственный технический университет». – Хабаровск, 2005. – 24 с. – Текст : непосредственный.

3. Маркова, К. В. Стратегии поиска работы на рынке труда : специальность 08.00.05 «Экономика и управление народным хозяйством: экономика труда»: автореферат на соискание ученой степени кандидата экономических наук / Маркова Ксения Викторовна ; МГУ им. М.В. Ломоносова. – Москва, 2003. – 28 с. – Текст : непосредственный.

4. Бейльханов, Д. К. Информационная технология принятия управленческих решений при подборе разработчиков программного обеспечения : специальность 05.13.10 «Управление в социальных и экономических системах» : диссертация на соискание ученой степени кандидата технических наук / Бейльханов Дамир Кайржанович ; ФГБОУ ВПО «АГТУ». – Астрахань, 2015. – 126 с. – Текст : непосредственный.

5. hh Статистика: сервис открытой аналитики рынка труда – Уровень конкуренции, ожидаемые и предлагаемые зарплаты, количество вакансий и резюме по регионам и профобластям. Смотрите данные в динамике и сравнивайте нужные вам показатели. – URL: https://stats.hh.ru/ (дата обращения: 29.05.2024) – Текст: электронный.

6. Eurostat Statistic Explained: ICT specialists in employment - Текст : электронный. – URL: https://ec.europa.eu/eurostat/statisticsexplained/index.php/ICT_specialists_in_employment#Number_of_ICT_specialists (дата обращения: 01.04.2024) – Текст: электронный.

7. Об утверждении Стратегии развития отрасли информационных технологий в РФ на 2014-2020 годы и на перспективу до 2025 года : Распоряжение Правительства Российской Федерации No 2036-р : [принят Правительством Российской Федерации 1 ноября 2013 года]. – Москва. – 2013. – Текст : непосредственный.

8. В IT больше не войти: падение зарплат и прогноз рынка труда на 2024 год С. Токарева – Где и кем выгодно работать в 2024 году. – URL: https://iz.ru/1645404/sofiia-tokareva/v-it-bolshe-ne-voiti-padenie-zarplat-i-prognoz-rynka-truda-na-2024-god (дата обращения: 12.04.2024) – Текст: электронный.

9. Игнатьева, О. В. Информационно-поисковые и аналитические системы: учеб. пособие / О. В. Игнатьева, С. А. Кулькин ; ФГБОУ ВО РГУПС. – Ростов н/Д 2017. – 150 с.– ISBN 978-5-88814-584-5. – Текст : непосредственный.

10. Маннинг, К. Д. Введение в информационный поиск / К. Д. Маннинг, П. Рагхаван, Х. Шютце Вильямс, 2020. – 528 с.– ISBN 978-5-907203-20-4. – Текст : непосредственный.

11. Бабина, О. И. Построение модели извлечения информации из технических текстов : специальность 10.02.21 «Прикладная и математическая лингвистика» : автореферат на соискание ученой степени кандидата филологических наук / Бабина Ольга Ивановна ; ТюмГУ. – Челябинск, 2006. – 24 с. – Текст : непосредственный.

12. Андриенко, Е. В. Исследование и разработка методов и моделей поиска адекватной информации в полнотекстовых базах данных : специальность 05.13.17 «Теоретические основы информатики» : автореферат

на соискание ученой степени кандидата технических наук / Андриенко Евгений Владимирович ; ТРТУ. – Таганрог, 2004. – 26 с. – Текст : непосредственный.

13. Рушди, А. А. Разработка методов и алгоритмов тематически ориентированного распределенного поиска информации в глобальных сетях типа Интернет : специальность 05.13.11 «математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» : автореферат на соискание ученой степени кандидата технических наук / Рушди Ахмад Амамра ; . – Санкт-Петербург, 2002.

14. Борисюк, Ф. В. Система поиска текстовых документов на основе автоматически формируемого электронного каталога : специальность 05.13.18 «Математическое моделирование, численные методы и комплексы программ» : автореферат на соискание ученой степени кандидата технических наук / Борисюк Федор Владимирович ; . – Нижний Новгород, 2010. – 23 с. – Текст : непосредственный.

15. Волков, С. С. Теоретическое обоснование и разработка интеллектуальной русскоязычной информационно-поисковой системы : специальность 05.13.01 «Системный анализ, управление и обработка информации» : автореферат на соискание ученой степени кандидата технических наук / Волков Сергей Сергеевич ; . – Краснодар, 2002. – 26 с. – Текст : непосредственный.

16. Григорьев, А. С. Разработка метода и создание системы полнотекстового поиска на основе статистической обработки ограниченного контекста слова : специальность 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» : автореферат на соискание ученой степени кандидата технических наук / Григорьев Александр Сергеевич ; . – Москва, 2006. – 20 с. – Текст : непосредственный.

17. Жмайло, С. В. Исследование и разработка теории и методики построения тезаурусов для информационного поиска в полнотекстовых базах

данных: На примере тезауруса по безопасности инженерных систем : специальность 05.13.17 «Теоретические основы информатики» : автореферат на соискание ученой степени кандидата технических наук / Жмайло Светлана Васильевна ; . – Москва, 2005. – 32 с. – Текст : непосредственный.

18. Арутюнян, Р. Э. Разработка архитектуры программной системы автоматизированного сбора тематической информации в сети Интернет : специальность 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» : автореферат на соискание ученой степени кандидата технических наук / Арутюнян Роман Эрнстович ; . – Ростов-на-Дону, 2004. – 24 с. – Текст : непосредственный.

19. Как повысить свои шансы пройти ИИ-фильтр на сайтах вакансий – Если вы давно и безуспешно ищете работу на популярных сайтах по трудоустройству, то вы не одиноки. Как известно, рекрутеры выбирают не лучшего, а того, кто лучше всех других подходит под их критерии.... – URL: https://habr.com/ru/companies/getmatch/articles/756714/ (дата обращения: 14.04.2024) – Текст: электронный.

20. Рынок труда в России онлайн. – URL: https://rutrud.com/ (дата обращения: 01.04.2024) – Текст: электронный.

21. Хабр Карьера. – URL: https://career.habr.com/ (дата обращения: 01.04.2024) – Текст: электронный.

22. Сервис анонимного поиска работы в ИТ и Диджитал. – URL: https://geekjob.ru/ (дата обращения: 01.04.2024) – Текст: электронный.

23. Вакансии в IT. – URL: https://tproger.ru/jobs/ (дата обращения: 01.04.2024) – Текст: электронный.

24. Компания HeadHunter – hh.ru — сервис, который помогает найти работу и подобрать персонал в Москве более 20 лет! Создавайте резюме и откликайтесь на вакансии. Набирайте сотрудников и публикуйте вакансии. – URL: https://hh.ru/article/28 (дата обращения: 14.04.2024) – Текст: электронный.

25. Умный поиск: как искусственный интеллект hh.ru подбирает вакансии к резюме – Больше половины соискателей ничего не ищут, а создают резюме и просто ждут, когда их пригласят на собеседование или хотя бы пришлют подходящую вакансию. Когда мы думали, как для них должен выглядеть... – URL: https://habr.com/ru/companies/hh/articles/347276/ (дата обращения: 14.04.2024) – Текст: электронный.

26. «Навыки» на hh.ru: как методология и алгоритмы сделали из них «бриллиант» Д. Скорых – Как заставить навыки реально работать и быть не просто набором слов, а ориентиром на рынке труда для всех его участников? Сделать процесс найма skill-based осязаемым рабочим инструментом смог hh.ru с участием команды цифрового сервиса Frontliner. Рассказываем, при чём тут точные науки, как команда разглядела клад среди хаотичного списка ключевых навыков и какие возможности это дало. – URL: https://hh.ru/article/31289 (дата обращения: 14.04.2024) – Текст: электронный.

27. Проверь на калькуляторе: как ИТ-специалистам понять свою реальную «стоимость» на рынке? Д. Скорых – Вот бы существовал такой способ, чтобы ИТ-соискателям можно было легко рассчитать зарплату, на которую они могут претендовать со своим скил-сетом. Стоп, так он уже есть! Команда hh.ru запустила бета-версию калькулятора навыков, который выполняет эти функции. Рассказываем, как создавался и работает продукт, а также почему считать на калькуляторе — теперь отличная идея (хотя в школе нас учили иначе). – URL: https://hh.ru/article/31689 (дата обращения: 14.04.2024) – Текст: электронный.

28. Сайты вакансий: обзор лучших сайтов по поиску работы, их отличия и специфика – Сайты вакансий: зачем они нужны, как работают, в чем их преимущество. Отличие сайтов вакансий от агрегаторов. Обзор сайтов трудоустройства в России: топ-5 лучших сайтов вакансий и агрегаторов. – URL: https://www.kp.ru/guide/saity-vakansii.html (дата обращения: 14.04.2024) – Текст: электронный.

29. Как машинное обучение помогает искать подходящие вакансии на SuperJob. – URL: https://www.superjob.ru/pro/5471/ (дата обращения: 14.04.2024) – Текст: электронный.

30. Сравнили поисковую выдачу Superjob.ru и HH.ru – Исследование HR-mnenie. – URL: https://blog.hr-mnenie.com/sravnili_poiskovuyu_vydachu_superjobhh (дата обращения: 14.04.2024) – Текст: электронный.

31. Emply.ru ищет инвестора – Неделю назад наш инвестор сообщил, что по ряду причин его доходы сократились и он больше не может финансировать нас в полном объеме. С Августа команду придётся сократить. Для продолжения разработки и... – URL: https://habr.com/ru/articles/292556/ (дата обращения: 14.04.2024) – Текст: электронный.

32. Вертикальный поисковик вакансий – Как часто вам хотелось дописать в запросе к Google параметры похитрей: "… с видом на море", или "… мощностью более 500 л/с", или "… цена не больше 100 рублей"? Именно так... – URL: https://habr.com/ru/articles/209294/ (дата обращения: 14.04.2024) – Текст: электронный.

33. Козлов, П. Ю. Нейро-нечеткие методы и алгоритмы анализа электронных неструктурированных текстовых документов : специальность 05.13.17 «Теоретические основы информатики» : автореферат на соискание ученой степени кандидата технических наук / Козлов Павел Юрьевич ; ФГБОУ ВО «Национальный исследовательский университет «МЭИ». – Москва, 2018. – 20 с. – Текст : непосредственный.

34. Ботов, Д. С. Методы и алгоритмы интеллектуальной поддержки формирования образовательных программ по требованиям рынка труда на основе нейросетевых моделей языка (дубл) : специальность 05.13.10 «Управление в социальных и экономических системах» : диссертация на соискание ученой степени кандидата технических наук / Ботов Дмитрий Сергеевич ; ФГБОУ ВО «Уфимский государственный авиационный

технический университет». – Челябинск, 2019. – 160 с. – Текст : непосредственный.

35. Абрамов, А. О. Разработка приложения для анализа актуальных требований рынка труда на основе текстов IT вакансий. Тюменский государственный университет, 2020. – С. 123-131.

36. Ботов, Д. С. Интеллектуальная Поддержка Формирования Образовательных Программ На Основе Нейросетевых Моделей Языка С Учетом Требований Рынка Труда / Д.С. Ботов – Текст : непосредственный. // Вестник Южно-Уральского Государственного Университета. Серия: Компьютерные Технологии, Управление, Радиоэлектроника. 2019. Т. 19. № 1. – С. 5-19.

37. Ермаков, П. Д. Выделение Ключевых Понятий Из Неструктурированных Текстов На Примере Выделения Навыков И Требований Из Текстов Резюме И Вакансий / П.Д. Ермаков. Московский институт электроники и математики НИУ ВШЭ, 2014. – С. 43.

38. Akkol, E. Topic Modeling for Skill Extraction from Job Postings / E. Akkol, M. Olucoglu, O. Dogan // Knowledge Graphs and Semantic Web / eds. F. Ortiz-Rodriguez [et al.]. – Cham: Springer Nature Switzerland, 2023. – P. 277-289.

39. Nikolaev, I. E. An intelligent method for generating a list of job profile requirements based on neural network language models using ESCO taxonomy and online job corpus / I.E. Nikolaev – Text : direct // Business Informatics. 2023. Vol. 17. № 2. – P. 71-84.

40. Ботов, Д. С. Извлечение информации с использованием нейросетевых моделей языка на примере анализа вакансий в системах онлайн-рекрутмента / Д.С. Ботов, Ю.Д. Кленин, И.Е. Николаев – Текст : непосредственный. // Вестник Югорского Государственного Университета. 2018. № 3 (50). – С. 37-48.

41. Широбокова, С. Н. Формализованная модель формирования рейтинга вакансий и выделения требований рынка труда к ключевым профессиональным компетенциям // Перспективы Науки. 2020. № 9 (132). –

С. 28-32.

42. Яруллин, Д. В. Информационная система сбора и обработки требований работодателей к компетенциям ИТ-специалистов на основе методов денотативного анализа : специальность 2.3.4. «Управление в организационных системах» : диссертация на соискание ученой степени кандидата технических наук / Яруллин Денис Владимирович ; ФГАОУ ВО «Пермский национальный исследовательский политехнический университет». – Пермь, 2023. – 152 с. – Текст : непосредственный.

43. Яруллин, Д. В. Автоматизация Планирования Потребности В It-Специалистах На Основе Онтологического Моделирования / Д.В. Яруллин, Р.А. Файзрахманов, П.Ю. Фоминых – Текст : непосредственный. // Математические Методы В Технике И Технологиях - Ммтт. 2020. Т. 8. – С. 67-71.

44. Николаев, И. Е. Метод извлечения знаний и навыков/компетенций из текстов требований вакансий / И.Е. Николаев – Текст : непосредственный. // Онтология Проектирования. 2023. Т. 13. № 2 (48). – С. 282-293.

45. Копытько, С. М. Оценка соответствия навыков кандидата требованиям вакансии с применением алгоритма на основе ориентированных графов / С.М. Копытько, А.А. Кузин – Текст : непосредственный. // Телекоммуникации И Информационные Технологии. 2023. Т. 10. № 2. – С. 5-11.

46. Фомичев, Д. А. Кластеризация вакансий по их описанию с использованием машинного обучения и методов анализа текста / Д.А. Фомичев – Текст : непосредственный. // Международная Конференция По Мягким Вычислениям И Измерениям. 2023. Т. 1. – С. 201-204.

47. Николаев, И. Е. Сравнение Нейросетевых Моделей На Архитектуре Трансформеров В Контексте Задачи Оценки Компактности Векторных Представлений Семантически Близких Текстов Требований Европейской Классификации Навыков Esco / И.Е. Николаев, А.В. Мельников – Текст : непосредственный. // Вестник Южно-Уральского Государственного

Университета. Серия: Компьютерные Технологии, Управление, Радиоэлектроника. 2022. Т. 22. № 3. – С. 19-29.

48. Bersenev, A. IT job ads from hh.ru, 2021-2022 / A. Bersenev. – Mendeley Data, 2022.

49. Korobov, M. Morphological Analyzer and Generator for Russian and Ukrainian Languages / M. Korobov // Analysis of Images, Social Networks and Texts : Communications in Computer and Information Science / eds. M.Yu. Khachay [et al.]. – Cham: Springer International Publishing, 2015. Vol. 542. – P. 320-332.

50. sentence-transformers (Sentence Transformers) – In the following you find models tuned to be used for sentence / text embedding generation. They can be used with the sentence-transformers package. – URL: https://huggingface.co/sentence-transformers (дата обращения: 25.05.2024) – Текст: электронный.

51. sentence-transformers/stsb-xlm-r-multilingual · Hugging Face – We’re on a journey to advance and democratize artificial intelligence through open source and open science. – URL: https://huggingface.co/sentence-transformers/stsb-xlm-r-multilingual (дата обращения: 25.05.2024) – Текст: электронный.

52. Yang, Y. Multilingual Universal Sentence Encoder for Semantic Retrieval. 2019.

53. sentence-transformers/distiluse-base-multilingual-cased-v2 · Hugging Face – We’re on a journey to advance and democratize artificial intelligence through open source and open science. – URL: https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2 (дата обращения: 25.05.2024) – Текст: электронный.

54. sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 · Hugging Face – We’re on a journey to advance and democratize artificial intelligence through open source and open science. – URL: https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 (дата обращения: 25.05.2024) – Текст: электронный.

55. sentence-transformers/paraphrase-multilingual-mpnet-base-v2 · Hugging Face – We’re on a journey to advance and democratize artificial intelligence through open source and open science. – URL: https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2 (дата обращения: 25.05.2024) – Текст: электронный.

56. Beautiful Soup: We called him Tortoise because he taught us. – URL: https://www.crummy.com/software/BeautifulSoup/ (дата обращения: 27.05.2024) – Текст: электронный.