Санкт-Петербургский государственный университет

**Кафедра математического моделирования энергетических систем**

**Чуднов Геннадий Сергеевич**

**Магистерская диссертация**

**Исследование рынка труда с применением современных методов анализа данных**

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа “Математическое и информационное обеспечение экономической деятельности”

Заведующий кафедрой, Захаров В. В.
доктор физ.-мат. наук,
профессор

Научный руководитель, Корхов В. В.
кандидат физ.-мат. наук,
доцент

Рецензент, Араканцев К. Г.
кандидат техн. наук,
ведущий инженер-программист,

АО "НПП "Радар ммс"

Санкт-Петербург

2017

Содержание

[Определения, обозначения и сокращения 4](#_Toc481681244)

[Введение 5](#_Toc481681245)

[Постановка задачи 7](#_Toc481681246)

[Обзор литературы 8](#_Toc481681247)

[Глава 1. Рекомендательные системы 11](#_Toc481681248)

[1.1 Подходы к построению рекомендательных систем 12](#_Toc481681249)

[1.1.1 Фильтрация на основе содержания 13](#_Toc481681250)

[1.1.2 Коллаборативная фильтрация 15](#_Toc481681251)

[1.1.3 Рекомендательные системы, основанные на знаниях 17](#_Toc481681252)

[1.2 Рекомендательные системы для рынка труда 18](#_Toc481681253)

[1.3 Примеры реализаций рекомендательных систем для рынка труда 19](#_Toc481681254)

[Глава 2. Формализация задачи 22](#_Toc481681255)

[2.1 Формализация задачи разработки рекомендательной системы для рынка труда 22](#_Toc481681258)

[2.2 Требования к программному комплексу 24](#_Toc481681259)

[2.3 Архитектура программного комплекса 27](#_Toc481681260)

[Глава 3. Сбор данных 29](#_Toc481681261)

[3.1 Выбор информационного источника 29](#_Toc481681265)

[3.2 Процедура сбора данных 31](#_Toc481681266)

[Глава 4. Подготовка данных 34](#_Toc481681267)

[4.1 Первичная подготовка данных 34](#_Toc481681272)

[4.2 Векторная модель представления документов 36](#_Toc481681273)

[4.3 Построение онтологии 37](#_Toc481681274)

[4.4 Процедура создания словаря ключевых слов 39](#_Toc481681275)

[4.5 Перенос вакансий в in-memory базу данных 41](#_Toc481681276)

[Глава 5. Разработка рекомендательной системы 43](#_Toc481681277)

[5.1 Обработка резюме пользователя 43](#_Toc481681283)

[5.2 Метрика подобия резюме и вакансии 44](#_Toc481681284)

[Выводы 46](#_Toc481681285)

[Заключение 49](#_Toc481681286)

[Дальнейшая работа 51](#_Toc481681287)

[Список литературы 53](#_Toc481681288)

# Определения, обозначения и сокращения

SNA – Social Network Analysis (Анализ социальных сетей)

CFRs – Collaborative filtering recommender system

CBRs – Content-based recommender system

KBRs – Knowledge-based recommender system

HR – Hybrid recommender system

# Введение

 Ни для кого не секрет, что мы проживаем в эпоху так называемой информационной перегрузки. Под понятием информационной перегрузки понимается проблема переизбытка информационного пространства, что влечет за собой определенные трудности при принятии решений в выборе того или иного предмета. По мере того, как растет объем данных в сети, увеличивается сложность поиска релевантной информации. Иными словами, отыскать необходимую информацию в сети Интернет становится достаточно трудоемкой задачей.

 Следовательно, возникает потребность в программных системах, которые фильтровали бы большой поток информации и выдавали конечному пользователю наиболее релевантный список результатов. Современный человек нуждается в подсказке о том, какие товары могут быть ему в наибольшей степени интересны среди всех прочих. И именно для решения этой непростой задачи были придуманы рекомендательные системы.

 Итак, рекомендательная система – это программное средство, которое генерирует рекомендацию товара или услуги для пользователя, основываясь на сведения о его профиле. По методам конструирования рекомендаций, можно выделить несколько основных подвидов рекомендательных систем: системы на основе совместной фильтрации, на основе анализа содержимого профилей и гибридные системы.

 В частности, рекомендательные системы подходят и для сферы рынка труда. Стремительный рост современных цифровых технологий и методов обработки данных значительно сократил время, которое требуется кандидату для того, чтобы ознакомиться с размещенными объявлениями о работе и установить связь с работодателем. Однако с ростом популярности интернет-сервисов по онлайн-рекрутменту увеличилось также количество вакансий и резюме. Это создает определенные трудности как для соискателей работы, так и для рекрутеров. Работа рекрутеров стала более сложной, поскольку из-за развития информационных технологий они стали получать намного больше резюме на каждую из открытых вакансий. Стало практически невозможно произвести ручную обработку резюме и отобрать те, которые соответствуют требованиям компании. Фильтрация и методы поиска отбирают из сотни резюме только те, которые подходят под нужные критерии. Большинство известных подходов базируются на разборе текста резюме, чтобы получить полезную информацию, и отфильтровать полученную информацию. Кроме того, резюме различаются по формату и стилю, что добавляет трудностей при анализе данных и выделению структурных зависимостей. В тоже самое время перед соискателем работы возникают не меньшие трудности. Соискателю приходится пройти через длительную и утомительную процедуру отбора результатов поиска, чтобы проверить каждое из описаний должностных обязанностей, что часто приводит к информационной перегрузке.

 Предлагаемая работа рассматривает проблему построения эффективной рекомендательной системы для соискателя работы. Принимаются во внимание все основные недостатки уже существующих аналогичных систем, такие как проблема синонимии и скорость генерации рекомендации. Предлагаемые подход нацелен на выявление наиболее актуальных и важных объявлений о работе, тем самым экономя огромное количество времени и усилий ищущего работу человека.

# Постановка задачи

Предметом исследования является набор вакансий и резюме в слабоструктурированной форме.

Цель работы заключается в проектировании и реализации эффективного алгоритма, позволяющего произвести анализ сходства навыков и пожеланий кандидата, полученных из его резюме, с должностными требованиями и условиями работы, указанными в объявлении о работе работодателем. В представленной работе учитывается опыт предыдущих исследователей в этом направлении и узкие места существующих решений, и решается проблема синонимии.

Результатом исследования будет реализованный программный комплекс - рекомендательная система, предоставляющая соискателю работы перечень наиболее подходящих ему вакансий на основе информации, полученной из резюме.

# Обзор литературы

В статье [1] автор рассказывает о методах извлечения полезной информации из базы данных резюме. Была сделана попытка получить резюме, выбирая их на основе определенных требований для заданной вакансии. Подход, используемый в данной работе, применяет метод скоринга для отбора лучших резюме. Категории навыков, конкретные навыки и уникальные навыки соискателя рассматриваются для определения уникальности резюме. Это помогает рекрутерам увеличить скорость просмотра резюме.

Автор статьи [2] применил три различных техники Data Mining для анализа данных соискателей: наивный байесовский классификатор, дерево принятия решений и правило принятия решений. Данные методы были реализованы, протестированы на входных данных, а также было произведено их сравнение. Сравнение трех методов показало, что метод дерева решений имеет наивысшую точность.

В работе [3] была разработана рекомендательная система для рынка труда iHR. В первой версии iHR авторы применяли стратегию фильтрации на основе содержимого для рекомендации всем пользователям в своей системе. iHR собирает профили пользователей из множества информационных каналов, включая демографическую информацию пользователей, извлеченные данные из внешних источников (например, резюме и домашние страницы) и повседневную деятельность пользователей. Объединение нескольких источников информации обеспечивает полную картину интересов пользователей. После этого авторы статьи признали тот факт, что разные пользователи могут иметь разные характеристики, и подход с использованием одной рекомендации может оказаться неприемлемым для всех пользователей. Поэтому они разработали новую версию iHR, которая будет использовать различные методы для групп пользователей [4]. Для этого, во-первых, они группируют кандидатов, используя метод кластеризации k-средних. Затем, для выдачи рекомендации каждому пользовательскому кластеру использовались три подхода к рекомендациям: фильтрация содержимого, совместная фильтрация, гибридная фильтрация.

Авторы статьи [5] используют каскадный фреймворк для извлечения информации из резюме. Для разбиения резюме на части используется скрытая марковская модель. Затем к полученным блокам текста применяется метод опорных векторов, чтобы получить подробную информацию, относящуюся к каждому блоку. В рамках этого процесса происходит извлечение информации, такой ​​как ФИО, адрес и образование.

В широко используемом наборе инструментов для обработки естественного языка GATE используется семантический анализатор JAPE, который применят механизм регулярных выражений для извлечения знаний из текстов [6]. В работе [7] авторы также прибегают к помощи регулярных выражений каскадные регулярные над блоками текста.

Авторы статьи [8] предложили подход, основанный на фильтрации содержания, и использовали метод расстояния Минковского для расчета близости вектора соискателя и рекрутера. Целью их работы является предоставление списка потенциальных претендентов на рабочее место работодателю. Процесс конструирования рекомендации осуществляется в три этапа: анализ содержимого, уточнения содержимого и сопоставления. В качестве входных данных для рекомендательной системы были выбраны резюме соискателей из социальной сети LinkedIn.

В работе [9] рассматривается рекомендательная система по трудоустройству на основе фильтрации содержимого, которая использует модель векторного пространства. В модели векторного пространства каждое вакансия преобразуется в вектор n-мерного пространства, где каждое измерение соответствовало текстовой функции из общего набора функций набора работ. Помимо этого, каждая вакансия представляется в виде вектора весов характеристик, где каждый вес определял степень ассоциации между функцией и вакансией. В результате, оценка правдоподобия вычисляется для каждого кандидата и для каждой вакансии, а затем выводится ранжированный список доступных работ, который строится исходя из полученных ранее баллов.

# Глава 1. Рекомендательные системы

В последнее время мы ежедневно сталкиваемся с огромным количеством информации из различных источников. Этот поток информации делает задачу поиска полезных или подходящих предметов, таких как фильмы, журналы, книги, веб-сайты, песни, книги и даже рабочие места, большой проблемой. Отдельно взятый пользователь не в состоянии охватить взглядом весь перечень предметов, а значит он нуждается в подсказке со стороны, какие товары или услуги наиболее точно соответствуют его интересам и запросам. По этой причине все больше и больше приложений получают широкое развитие, а также появляются новые методы поддержки принятия человеческих решений, предлагающих клиентам услуги, продукты и различные типы информации [10]. Одной из областей исследований в этом направлении является направление рекомендательных систем (РС).

Рекомендательные системы - это программные инструменты, которые используют различные методы и алгоритмы, чтобы изолировать нерелевантную информацию от огромного количества данных и создать персонализированные предложения из небольшого подмножества из них, которые пользователь может изучить в разумные сроки. В действительности, рекомендации – это догадки, предположения, сделанные системой о предмете, который клиент, скорее всего, предпочитает. Для построения таких рекомендаций существует несколько различных методов, и для того, чтобы предоставлять персонализированные рекомендации, необходимы знания о интересах пользователя. Эти интересы могут быть собраны из оценок товаров или услуг, истории покупок или других взаимодействий с системой, сохраненных в профиле пользователя.

Рекомендательные системы имеют некоторые особенности, которые отличают их от других общих научных областей, таких как распознавание образов, искусственный интеллект, интеллектуальный анализ данных и т. д.

Рекомендационная система персонализирована. Выработанные рекомендации предназначены для оптимизации опыта одного конкретного пользователя. Рекомендованная система предназначена для того, чтобы помочь пользователю выбрать один из дискретных вариантов. Как правило, предметы уже известны заранее и не генерируются на заказ.

Известно множество примеров применения рекомендательных систем. Многие интернет-сервисы используют рекомендательные системы в целях предоставления персонализированных предложений пользователю. Наличие такого сервиса помогает увеличить популярность сайта, а также продажи, в случае площадок для электронной коммерции. К таким сервисам относятся, например, Amazon.com, Last.fm, LinkedIn, Facebook и многие другие.

## Подходы к построению рекомендательных систем

Для поиска подходящих товаров для пользователей существуют различные техники рекомендаций и классификаций. Прежде всего, следует провести различие между персонализированными и не персонализированными рекомендации. Персонализированные рекомендации основаны на информации от индивидуальных пользователей, тогда как не персонализированные рекомендации дают одинаковые результаты для всех пользователей и могут использовать статистические данные, например, самые продаваемые товары. В связи с бесспорной полезностью персонализированных рекомендательных систем для общества, исследователи заинтересованы в их изучении и развитии. Все они различают, по крайней мере, следующие методы [11]:

* Фильтрация на основе содержания (content-based recommenders);

Производится сравнение пользователей с пользователями, чтобы предлагать предпочтительные товары от подобных пользователей.

* Коллаборативная фильтрация (collaborative filtering recommenders);

Производится поиск товаров и услуг, похожих на те, которые нравится пользователю.

* Системы, основанные на знаниях (knowledge-based recommenders);

Используют дополнительные знания, чтобы вычислить товары и услуги, которые наилучшим образом соответствуют потребностям пользователя.

* Гибридная рекомендательная система (hybrid recommenders);

Гибридные системы сочетают в себе подходы фильтрации на основе содержания и коллаборотивной фильтрации.

Во всех этих методах используются разные подходы к рекомендациям, и все они имеют свои сильные и слабые стороны, что делает некоторые более подходящими в определенных областях, нежели другие. Рассмотрим эти типы систем более подробно, а также укажем их достоинства и недостатки.

## Фильтрация на основе содержания

Системы рекомендаций на основе контента анализируют характеристики элементов, чтобы идентифицировать среди них те, которые предоставляют особый интерес для пользователя. Анализируется набор характеристик предметов, которые оцениваются целевым пользователем и формируют профиль интересов этого пользователя на основе характеристик предметов. Процесс рекомендации заключается в сопоставлении атрибутов профиля пользователя с набором свойств содержимого товара. Этот тип системы рекомендаций может использоваться во множестве различных областей, таких как рекомендации в интернет-магазинах, новостных статей и контента в социальных сетях.

Одна из распространенных техник для этой стратегии состоит в хранении ключевых слов документов в n-мерных векторах. Каждому слову назначается свой вес. Для определения весов могут быть использованы техники TD-IDF, расстояние Кульбака — Лейблера, Okapi BM25. Далее, после того, как построены векторы ключевых слов, применяется Коэффициент Отиаи (косинусный коэффициент), для сравнения вектора профиля пользователя с возможными векторами документа. Чем меньше угол, тем ближе документ к профилю пользователя. Формула косинусного коэффициента имеет следующий вид:

 (1)

где A – вектор профиля пользователя, B – вектор документа.

Стратегия построения рекомендательной системы, в основе которой лежит фильтрация содержания, обладает несколькими известными изъянами:

* Ограниченный анализ содержимого

Рекомендатор в большой степени зависит от доступной информации из документов. Таким образом, документы либо должны содержать машиночитаемый текст, либо они должны быть вручную классифицированы пользователями. Кроме того, популярность двух документов, если они имеют одни и те же векторы, не может быть различена такими системами на основе контента.

* Проблема холодного старта

Для того чтобы рекомендательная система работала, пользователи должны сначала указать свои предпочтения. Без этой информации никакие рекомендации не могут быть сделаны.

Рисунок 1. Фильтрация на основе содержания.

##  Коллаборативная фильтрация

Стратегия, в основе которой лежит коллаборативная фильтрация, рассматривается как одна из наиболее успешных стратегий к созданию систем рекомендаций. Такие рекомендательные системы используют известные предпочтения группы пользователей, чтобы давать рекомендации или предсказания неизвестных предпочтений для других пользователей [12]. Для предоставления рекомендаций система находит аналогичных пользователей, которые имеют одинаковые характеристики, вкусы, предпочтения с рассматриваемым пользователем, и вычисляет наиболее подходящие виды товаров для этого пользователя на основе предпочтений аналогичных пользователей [12]. Ключевой момент в CFR - вычисление сходства между пользователями. Концепция подхода совместной фильтрации отображена на Рис. 2. Данный вид рекомендательных систем имеет следующий перечень недостатков:

* Элементы, которые являются новыми в системе, должны оцениваться пользователями до использования в процессе рекомендации. Эта проблема, так же как и проблема нового пользователя, в литературе часто упоминается как проблема холодного старта.
* Разреженность рейтингов.

Люди склонны оценивать похожие предметы, что приводит к разреженности рейтингов, то есть когда только несколько товаров получили оценку пользователей.

* Белые вороны.

Особенные пользователи, заинтересованным в редких предметах, чье мнение не совпадает с мнением большинства.

* Синонимия

 Одни и те же сущности имеют разные наименования.



 Рисунок 2. Совместная фильтрация.

 Клиент 1 предпочитает товары 1-4, следовательно, клиент 2 получит рекомендацию о товаре 3 или 4.

##  Рекомендательные системы, основанные на знаниях

Основанные на знаниях системы рекомендаций представляют собой специальный тип системы рекомендаций, основанный на явных знаниях об ассортименте товаров, предпочтениях пользователей и критериях рекомендации. Эти системы применяются в тех случаях, где не могут применяться альтернативные подходы, такие как коллаборативная фильтрация и фильтрация на основе контента. Основным преимуществом основанных на знаниях систем рекомендаций является отсутствие проблем с холодным стартом. По сравнению с другими описанными до сих пор рекомендациями, этот метод не страдает от проблемы холодного старта. С другой стороны, рекомендация пользователю в такой системе в большой степени зависит от имеющихся знаний и требует дополнительных усилий для сбора и поддержки знаний.

* + 1. Гибридные рекомендательные системы

Чтобы избежать некоторых из упомянутых недостатков рассмотренных выше подходов к построению рекомендации пользователю, возможна комбинации из двух или более методов. В частности, рекомендательную систему на основе содержимого можно объединить с рекомендательной системой, использующей стратегию совместной фильтрации, для преодоления нескольких проблем.



Рисунок 3. Гибридная система.

## Рекомендательные системы для рынка труда

В последнее время большинство ищущих работу прибегают к услугам Интернета для поиска работы в системах онлайн-рекрутинга. Методы поиска информации, используемые этими веб-сайтами, в большинстве своем основаны на методе введения вручную поискового запроса с некоторыми расширенными фильтрами для ранжирования результатов. Улучшающиеся с каждым годом методы машинного обучения позволяют программного извлечь соответствующую информацию о соискателе и подобрать наиболее подходящие под его навыки предложения от работодателей без активного ввода ключевых слов со стороны пользователя. Более того, принимая во внимание быстрый рост объемов информации в этих системах поиск работы становится достаточно трудоемкой задачей. Иными словами, соискатели столкнулись с проблемой избытка информации, так называемой информационной перегрузкой. Отсюда следует, что существует потребность в разработке систем рекомендаций по трудоустройству, которые автоматически предлагают ранжированный список доступных вакансий, соответствующих конкретным запросам и интересам соискателей. Никто не станет отрицать такой очевидный факт, что эти системы играют важную роль в установлении связей между работниками и работодателями. Как правило, системы рекомендаций по трудоустройству служат не только для работодателей, но также и для кандидатов. Ищущие работу получают ранжированный список рекомендуемых вакансий, связанных с их предпочтениями и навыками, в то время как работодатели получают ранжированный список потенциальных кандидатов, соответствующих их потребностям.

Разница между традиционными рекомендательными системами и рекомендационными системами по трудоустройству заключается в том, что первые рекомендуют товары или услуги пользователям, а вторые рекомендуют одного типа пользователей (например, работодателей) другому типу пользователей (например, кандидатам на работу). Иными словами, рекомендательные системы в сфере рынка труда разрабатываются таким образом, чтобы предлагать либо список доступных предложений с работой соискателю на основе его предпочтений и интересов, либо получать список кандидатов на работу работодателю или рекрутеру на основе требований к работе.

Таким образом, рекомендательные системы для рынка труда предназначены для того, чтобы предложить ранжированный список рабочих мест, который может быть связан с интересом сотрудника. Большинство существующих систем используют только один подход, чтобы рекомендовать всех сотрудников, в то время как определенный метод обычно достаточно хорош для группы сотрудников.

## Примеры реализаций рекомендательных систем для рынка труда

В последние годы рекомендательные системы по трудоустройству привлекли интерес сообщества к их изучению. Существует несколько популярных систем, такие как PROSPECT[13], CASPER[14], Proactive [15], iHR [4], а также двусторонний рекомендатор[16]. В этих системах используются различные подходы к вынесению рекомендаций. Для составления рекомендаций используются стратегии фильтрации на основе содержимого, совместной фильтрации, knowlegde-based и гибридную стратегии.

CASPER (Case-based Profiling for Electronic Recruitment) - пример методики персонализации контента в области онлайн-найма. Его цель - персонализировать поиск работы, задачу, которая выполняется в два этапа: сначала выполняется поиск заданий на основе сходства на стороне сервера, а во-вторых, применяется персонализация на стороне клиента. На первом этапе традиционные запросы не выполняются, а берутся из базы данных, которая лучше всего соответствует запросу, и сравнивает ее с другими запросами. Для сравнения используются различные показатели сходства, например, сравнение навыков поддерживается онтологией предметной области. После того, как аналогичные работы были получены, в качестве второго шага результат фильтруется согласно профилю пользователя. Профиль состоит из ранжированных работ, и классификатор использует эти данные для группировки результатов поиска в релевантные и нерелевантные работы. Для построения профиля пользователя используется неявная обратная связь от различных действий пользователя, в том числе просмотр вакансий, отправка резюме по электронной почте. Система CASPER использует гибридный подход для вычисления рекомендаций.

PROACTIVE - это веб-приложение, которое помогает соискателям найти подходящие открытие вакансии несколькими способами. Эта система ориентирована на работу, связанную с информационными технологиями. В начале каждого сеанса система выбирает и отображает работы, опубликованные за последние 24 часа. Когда пользователь находит интересную работу, он назначает ее в качестве подходяшей работы. Рекомендации создаются на основе свойств избранных работ пользователя. Когда пользователь добавляет работу в избранное, создается новый список рекомендуемых работ. Наконец, анализируя предпочтения пользователя, рекомендательная система предлагает набор вакансий. Таким образом, в профиле пользователя учитывается только обратная связь от добавленных им закладок. В расширенной версии в профили была добавлена так называемая неявной отрицательная обратная связь. Всякий раз, когда ищущий работу читает описание вакансии, но не добавляет эту вакансию в избранное, получается отрицательная обратная связь. Добавление отрицательной обратной связи имеет смысл, чтобы провести различие между релевантными и нерелевантными рабочими местами.

Для двустороннего рекомендатора была изобретена система рекомендаций по трудоустройству соискателей, а также рекомендации кандидатов на рабочие места. Предполагается, что предпочтения работодателя и предпочтения ищущего работу должны быть рассмотрены в рекомендаторе. Для этой цели были реализованы два от системы: система для рекрутеров и система для кандидатов. Для самого процесса рекомендации была создана вероятностная модель, которая предсказывает, является ли работа или резюме подходящими или нет.

# Глава 2. Формализация задачи

В этой главе описываются принципы и концепции разработки рекомендательной системы для рынка труда, а также предлагается формализация задачи построения рекомендательной системы. Прежде всего, описываются требования к системе и обоснование необходимости в ее разработке. Далее будет изображена общая схема архитектуры программного комплекса, реализующего предлагаемую рекомендательную систему. Вдобавок к этому, будет приведено краткое описание основных компонентов программного комплекса. Более подробное описание принципов действия компонентов приложения будет приведено в следующих главах.

1.
2.

## Формализация задачи разработки рекомендательной системы для рынка труда

Формальное определение задачи позволит провести исследование проблемы в более точных терминах. Итак, формально задача генерирования рекомендаций по подбору людей и рабочих мест может быть сформулирована следующим образом:

* Входные данные:
* Выходные данные:
	+ Построенная функция полезности , которая возвращает число – оценка, насколько работа подходит кандидату .
	+ Для каждого кандидата рекомендательная система должна построить ранжированный список возможных рабочих мест для на основе полученной оценки от функции полезности .

Таким образом, пусть Y - множество всех вакансий, и X - множество всех возможных кандидатов на рабочее место, которые могут быть рекомендованы. Пусть f - функция полезности, которая измеряет пригодность кандидата-работника x для работы y. Отображение имеет следующую сигнатуру f: X x Y → R, где R - вполне упорядоченное множество (например, действительные числа в определенном диапазоне). Другими словами, функция возвращает некое значение, по которому можно сделать вывод о пригодности кандидата на работу.

Из вышеприведенного определения выделяются три основных сущности, которые характеризуют рекомендательные системы для сопоставления кандидатов и рабочих мест.

Можно выделить два разных типа отношений между соискателями и работодателями по принципу что рекомендуется и кому дается рекомендация. Ожидается, что рекомендательная система может предоставлять рекомендации двумя способами:

1) Рекомендация большинства подходящих кандидатов для указанной работы в соответствии с перечнем требований, описанных в вакансии. Такой тип рекомендации будет полезным с точки зрения рекрутера.

2) Рекомендация доступных на данный момент рабочих мест ищущим работу в соответствии с их предпочтениями, опытом и навыками, указанными в резюме.

В этой работе основное внимание уделяется только второй части, то есть на генерации рекомендаций возможных рабочих мест кандидатам в соответствии с их заполненным резюме. Предпринимается попытка предусмотреть предпочтения клиента в отношении рабочих мест по различным параметрам, таким как предлагаемой должности, предложенной шкале заработной платы, местонахождения офиса, должностных обязанностей и другие. Вместо отслеживания прошлой истории мест работ кандидата, система сосредоточена на его текущих предпочтениях и пожеланиях.

Существует множество алгоритмов, которые вычисляют пригодность кандидата для работы. Стоит отметить, что предлагаемая работа решает проблему соответствия людей и рабочих мест с использованием предложенного нами алгоритма в качестве функции пригодности, который позволяет добиться наибольшей точности.

Одной из ключевых проблем в задачах построения рекомендаций на основе содержимого на рабочем месте является полнота содержимого. Каждая работа и кандидат должны быть описаны на достаточном уровне детализации для того, чтобы система могла сделать наиболее точный прогноз.

## Требования к программному комплексу

В качестве отправной точки для этапа разработки необходимо собрать требования для рекомендательной системы.

* + Система оценки эффективности

Конфигурация рекомендательной системы содержит множество различных параметров, таких как веса для полей и действий пользователя, которые влияют на результаты рекомендации. Поскольку процесс рекомендации является сложным, без каких-либо показателей эффективности, трудно сказать, приводят ли изменения в конфигурациях к повышению точности или нет. Следовательно, требуется система, которая измеряет производительность, такую как точность различных настроек.

* + Интеграция релевантных отзывов в профиль пользователя

После регистрации действий пользователя предпочтения должны быть выведены и объединены в существующий профиль пользователя. Поскольку текущий профиль пользователя состоит из резюме, а предпочтения - для работ, будет необходим шаг преобразования и согласования различных структур данных.

* + Взвешивание навыков пользователя

Различные действия обеспечивают различные уровни доказательств интересов. Например, подача заявки на работу дает больше доказательств для интереса к этой работе, чем просто чтение описания работы. В результате этого требуется система взвешивания для различных действий, где действия с более высоким уровнем доказательств оказывают более сильное влияние на рекомендации. Эти веса должны быть конфигурируемыми для администратора.

* + Пользовательский интерфейс для конфигурации рекомендательной системы

Текущая реализация имеет один главный недостаток: многие конфигурации жестко закодированы в программе. Небольшие изменения в процессе рекомендации требуют внесения изменений в код. Эти изменения обычно могут делать только программисты и исключают обычных администраторов веб-сайта. Кроме того, это время для тестирования различных конфигураций, поскольку каждое изменение требует обновления кода. Как следствие, настоятельно необходим пользовательский интерфейс для настройки рекомендательной системы.

* + Общая реализация

Как уже упоминалось, подобные рекомендации, как на том же сайте, так и на других, могут следовать одному и тому же методу рекомендации. Таким образом, должна быть достигнута более общая реализация, которая не зависит от веб-сайта или настроек домена, таких как имена полей.

* + Гибкая интеграция таксономии

Большинство данных для процесса рекомендации связано с концепциями таксономии. Это помогает вывести более общие понятия, которые могут быть добавлены к процессу сопоставления. На данный момент каждое поле данных, относящееся к таксономии, обрабатывается одинаково: на этапе предварительной обработки общие понятия добавляются в поле и поэтому включаются в запрос соответствия. Поскольку каждое понятие находится в одной и той же области, никакое разделение между данными и выведенными понятиями невозможно. В некоторых ситуациях могут быть интересны различные весовые коэффициенты. Более того, не все поля должны следовать одному и тому же подходу: некоторые могут только хотеть соответствовать определенным концепциям, другие хотят включать некоторые родительские концепции и так далее. Соответственно, требуется более гибкая и настраиваемая интеграция таксономии.

* + Рассмотрение различных уровней навыков.

При заполнении формы резюме кандидаты могут указать свои уровни квалификации, например, кто-то является экспертом по программированию на Java, но имеет только базовые навыки в С++. Эта информация должна быть представлена в алгоритме рекомендации, поскольку навыки экспертов должны иметь большее влияние на результаты, чем базовые навыки.

* + Высокое качество рекомендаций

Желательна высокая точность, чтобы рабочие места соответствовали интересам и возможностям пользователя. На точность влияет метод рекомендации и различные настройки конфигурации.

* + Высокая производительность

Предлагаемая реализация должна быть способна вычислять рекомендации с очень низким временем выполнения.

## Архитектура программного комплекса

На Рис. 4 проиллюстрирована схема архитектуры приложения, которое решает задачу выдачу ранжированного списка подходящих вакансий для соискателя. На Рис. 4 показаны основные компоненты и их взаимодействия на очень высоком уровне абстракции.



Рисунок 4. Архитектура системы построения рекомендаций

Программный комплекс состоит из следующих основных подсистем:

* Обработчик вакансий;
* База данных вакансий;
* Обработчик резюме;
* Система построения рекомендаций;

Детальный обзор каждого компонента рекомендательной системы будет дан в последующих главах.

# Глава 3. Сбор данных

В этой главе описывается процесс сбора тестовых данных – вакансий.

1.
2.
3.

## Выбор информационного источника

Для оценки качества и эффективности разрабатываемой рекомендательной системы была проведена экспериментальная процедура с использованием реальных данных.

 В процессе поиска источника данных учитывались следующие факторы:

1. Доступность.

Информационный ресурс должен либо предоставить готовую базу данных вакансий и резюме, либо обладать полнофункциональным API (программный интерфейс приложения), благодаря которому можно было бы собрать все необходимые сведения о резюме и вакансиях.

1. Актуальность.

Информационное наполнение данных должно в полной мере отражать всевозможные атрибуты вакансий для более детального анализа.

1. Единообразность.

Для обработки большого количества вакансий необходимым условием является наличие единого формата данных.

Рассматривались данные, доступные на веб-сайте www.kaggle.com. В 2010 году Kaggle была основана как платформа для интеллектуального моделирования и аналитических соревнований, на которых компании и исследователи публикуют свои данные, а статистики и разработчики со всего мира конкурируют за выпуск лучших моделей. Одним из конкурсов этой платформы был Job Recommendation Challenge, для которого организаторы опубликовали данные. В общих чертах, предоставлены данные о пользователях, объявлениях о вакансиях и резюме, которые пользователи сделали для публикации вакансий. В общей сложности приложения охватывают 13 недель. Все резюме на 7 групп, каждая группа представляет 13-дневное окно. Каждое 13-дневное окно делится на две части: Первые 9 дней - это период обучения, а последние 4 дня - период тестирования. Информация о вакансии состоит из: названия работы, описания работы, требований, навыков и т.д., а профиль соискателя состоит из: уровня образования, опыта работы, истории работы и т.д.

Данные с сайта kaggle.com предоставляются в виде не обновляемой базы данных и, таким образом, не могут отражать реальное состояние рынка труда на данный момент времени. Тем не менее, эти данные вполне пригодны для их анализа и построения рекомендательной системы на их основе.

 Помимо базы данных с вышеупомянутого сайта также принимались к рассмотрению вакансии и резюме, размещенные на интернет-портале по поиску работы и сотрудников, сайте hh.ru. В настоящее время, это один из крупнейших сайтов интернет-рекрутмента.

 Благодаря открытому API, который предоставляет сайт hh.ru, разработчики имеют доступ к обширной постоянно обновляющейся базе данных рынка труда, а это означает возможность глубокого анализа реальных данных. Составляя правильные обращения к сайту, можно получить детализированные сведения как о актуальных вакансиях, так и о вакансиях, перенесенных в архив. Более того, в случае авторизованного пользователя, предусмотрена возможность работы либо с резюме соискателя, либо с вакансиями работодателя. Подробное описание доступного функционала можно найти на сайте https://github.com/hhru/api.

Для того чтобы получить интересующую информацию о вакансиях или резюме, необходимо направить сайту GET-запрос по протоколу HTTPS. В качестве ответа на запрос возвращается результат выполнения операции в формате JSON, содержащий запрашиваемые поля. В том случае, если приложению требуется авторизация пользователя, это можно сделать по протоколу ОAuth2.

 Исходя из вышеперечисленных критериев, которым должны удовлетворять источники информации о рынке труда, был выбран вариант с веб-ресурсом hh.ru.

## Процедура сбора данных

Для извлечения сведений об актуальных вакансиях c сайта hh.ru была написана программа на языке программирования Python.

Программа получает список вакансий, опубликованных на сайте за определенный интервал времени. Для формирования GET-запросов применяется стандартная библиотека httplib из языка Python.

Разработчикам предоставляется функция поиска по вакансиям. В частности, при конструировании запроса можно задать такие поля, как даты, ограничивающие снизу и сверху диапазон дат публикации вакансий, индустрия компании, область поиска и параметры плагинации.

Поскольку в результате запроса возвращается лишь часть атрибутов, описывающих каждую вакансию, то необходимо сделать отдельный запрос на получение полных сведений для каждой из вакансий. Для достижения наибольшей эффективности и скорости получения атрибрутов вакансий, каждая вакансия обрабатывается в отдельном потоке операционной системы. Поддержка многопоточного программирования также заложена в стандарт языка программирования Python и доступна в виде подключаемого модуля threading.
 Используя механизм GET-запросов, можно получить такие атрибуты вакансии, как:

* Идентификатор вакансии;
* Описание вакансии;
* Информация о ключевых навыках, заявленных в вакансии;
* График работы;
* Требуемый опыт работы;
* Адрес вакансии;
* Тип занятости;
* Оклад;
* Регион размещения вакансии;
* Название профессиональной области;
* Email контактного лица;
* и другие свойства вакансии;

Как уже было сказано, ответы на запросы поступают в широко распространенном формате JSON, для обработки которого в “Питоне” предусмотрена соответствующая библиотека.

Среди ограничений, накладываемых на извлечение данных, стоит отметить, что выдается не более 500 вакансий при однократном запросе.

Необходимо добавить, что аналогичная процедура извлечения свойств и атрибутов доступна не только для вакансий, но и для резюме. В зависимости от того, какая область видимости задана для резюме соискателя, приложению либо требуется авторизация пользователя, либо нет. Сведения о резюме включают в себя такие поля, как фамилия, имя, отчество, возраст, пол, название города, ближайшая станция метро, готовность к переезду, желаемая зарплата, образование, опыт работы, перечень ключевых навыков, комментарий к контакту и другие поля.

Таким образом, накопив базу данных вакансий с помощью реализованной программы-сборщика, можно приступать к их анализу, о чем пойдет речь в следующей главе.

# Глава 4. Подготовка данных

Поскольку работодатели составляют должностные требования и описание условий работы в свободной слабоструктурированной форме, возникает задача произвести предварительную подготовку и фильтрацию этих данных для последующей работы. Точно так же, как и описание вакансии, текст резюме соискателя необходимо провести через дополнительные процедуры подготовки. Подробности и подходы к решению поставленной задачи будут рассмотрены в этой главе.

1.
2.
3.
4.

## Первичная подготовка данных

Поступающие в программный комплекс данные проходят три этапа обработки: разбор, сегментирование и токенцизация. Эти шаги проиллюстрированы на Рисунке 5.



Рисунок 5. Схема первичной обработки вакансий и этап преобразования в векторную модель.

В первую очередь через процедуру подготовки данных проходит поле вакансии, содержащие ее описание. Данное поле заполняется работодателем в свободной форме и в отличие от остальных атрибутов вакансии не имеет четко выраженной структуры. По этой причине указанное поле должно пройти несколько стадий обработки.

Содержимое размещенных объявлений о работе подвергается фильтрации с тем, чтобы произвести очистку текста от не несущих смысловой нагрузки конструкций и извлечения связанных атрибутов в форме абзацев и предложений.

На этапе парсинга(разбора) исходный текст описания вакансии разбивается на абзацы и отдельные предложения, для последующей обработки.

Под этапом сегментирования подразумевается извлечение таких атрибутов вакансии, как название компании, территориальное расположение компании и т.п. В то время как для резюме сохраняются поля, содержащие сведения об образовании кандидата, его опыте работы, ключевые навыки и т.п.

На этапе токеницазии из текста исключаются символы, которые не передают никакой значимой информации. Вместо этих символов программа вставляет на их место пробелы. К таким незначащим символам относятся, например, точки, запятые, дефисы, подчеркивания и другие. Более того, в этот этап включена процедура удаления так называемых стоп-слов. К таким словам относятся междометия, цифры, частицы, предлоги и т.п.

Заключительным этапом первичной обработки является стемминг.

Под стеммингом подразумевается процедура, которая принимает на вход слово, и выявляет его основу. К примеру, на основе слов “языки”, “языков”, “языка” будет выведено слово “язык”.

 В результате проведения вышеперечисленных процедур создается программный объект, который содержит отформатированные сведения о вакансиях и загруженном резюме, пригодные для следующих этапов обработки.

 Алгоритм первичной обработки ожидает на вход данный в JSON-формате. JSON - это открытый, общедоступный и машиночитаемый стандарт, который облегчает обмен данными, и наряду с XML является основным форматом обмена данными, используемым в современной сети. JSON поддерживает все основные типы данных: числа, строки и логические значения, а также массивы и хэши.

 Для осуществления стадий первичной обработки использовался язык программирования Python и сторонний подключаемый модуль NLTK (Natural Language Toolkit). Этот программный модуль включает в себя набор библиотек для обработки естественного языка, который к тому же поддерживает работу с русским языком.

## Векторная модель представления документов

После того, как над исходным текстом вакансии была проведена процедура первичной обработки, необходимо представить текст в более удобной для машинного анализа форме. Одним из распространенных методов представления текста является векторное представление документов.

При использовании векторной модели представления слов, исходный документ (резюме или объявление о работе) реконструируется в вектор параметров. Вектор состоит из слов, для которых задается численная характеристика, а именно вес, который определяет важность и значимость слова в рассматриваемом тексте. Таким образом, для каждой вакансии, которая прошла первичную обработку, ставится в соответствие вектор одной и той же длины.

Более конкретно, вектор для j-го документа имеет вид:

*xj* = (*w1j*, *w2j*, …, *wnj*) (2)

где *xj* — векторное представление *j*-го документа, *wij* — вес *i*-го cлова в *j*-м документе, *n* — общее количество различных слов во всех документах из набора.

Однако описывать каждый документ вектором настолько большой длины не разумно. Предлагаемый подход решения проблемы основан на составлении словаря ключевых слов для каждой профессиональной отрасли. Принадлежность вакансии к той или иной профессиональной области обозначена в ее атрибутах, а сам список профессиональных отраслей можно получить с помощью GET-запроса к одному из справочников сайта https://api.hh.ru.

Для достижения цели преобразования документов в вектора параметров был применен пакет библиотек scikit-learn для языка программирования Pyhton. Scikit-learn - это пакет библиотек для машинного обучения для языка Python. Она имеет различные алгоритмы классификации, регрессии и кластеризации, включая метод опорных векторов, случайные леса, повышение градиента, k-средних, и предназначен для взаимодействия с численными и научными библиотеками NumPy и SciPy.

Прежде чем перейти к вопросу о том, каким образом задаются веса для элементов вектора, необходимо рассмотреть дополнительные концепции и архитектурные решения, которые применяются в данной работе.

## Построение онтологии

Под термином онтология понимается задача детальной формализации и структурирования определенной сферы знаний с использованием концептуальной схемы.

Одна из главных сложностей, с которой сталкиваются рекомендательные системы – это проблема синонимии. Другими словами, когда один и тот же по смыслу термин записан в различной форме, или определенный термин в одном документе является более общим понятием, чем указанный термин в другом.

Прежде чем вектор будет окончательно сохранен в базу данных, он будет дополнен родительскими концепциями таксономии. Например, если у кого-то есть навыки в MySQL, родительское понятие «Базы данных» будет добавлено к вектору. Это позволяет алгоритму сопоставления принимать во внимание родственные понятия, например, для задания поиска знаний PostgreSQL, кандидат с MySQL будет соответствовать, поскольку обе концепции являются братьями и сестрами и имеют одного и того же родителя. Конечно, такое соответствие должно быть менее важным, чем идеальное совпадение.

Большинство разработанных рекомендательных систем предлагают производят сопоставление на основе точного соответствия ключевых слов. Хотя этот метод обеспечивает правдоподобные результаты, он не использует лингвистические свойства, специфичные для конкретной области. Когда мы рассматриваем текст, используемый в описаниях должностных обязанностей и резюме, для описания соответственно требований к работе и личной квалификации, мы обнаруживаем иерархические отношения между навыками и квалификацией. Во-вторых, существует значительное количество неоднозначностей между конкретными словами области и их соответствующей интерпретацией.

Чтобы добавить в документ наблюдаемые взаимосвязи, которые существуют между навыками, была разработана специальная система онтологии навыков предметной области. Онтология предоставляет формальное определение объектов, свойств и взаимосвязей между объектами для данной области. Таким образом, создается таксономия, которая разделяет переменные, необходимые для вычисления сходства между навыками, тем самым устанавливая формальные отношения между ними. Используя таксономию, предоставляемую через DBpedia, строится онтология. Например, на Рисунке 6 проиллюстрирована онтология языков программирования.



Рисунок 6. Таксономия языков программирования.

Иерархия навыков представляет собой множество триплетов, включающих в себя субъект, предикат и объект. Для извлечения этой информации использовалась библиотека rdflib языка Python.

 Итак, благодаря реализации онтологии в рекомендательной системе для рынка труда решается задача синонимии.

## Процедура создания словаря ключевых слов

Возвращаясь к вопросу о назначении весов отдельно взятым словам из документов, необходимо отметить следующее.

Как говорилось ранее, для построения векторных или семантических сетевых профилей наиболее важные характеристики соответствующих документов извлекаются и добавляются в модель. Чтобы определить наиболее важные термины, часто требуется предварительная обработка документов с неструктурированным текстом. Методы поиска информации анализируют текст, извлекают ключевые слова и создают вектор, содержащий ключевые слова с соответствующими весами, которые обозначают важность термина. Весы как правило рассчитываются с помощью TF-IDF (Term Frequency - Inverse Document Frequency), который имеет вид (формула (3)). Термин-частота (TF) подсчитывает, сколько раз слово появляется внутри документа, в то время как обратная частота документа (формула 4) уменьшает вес терминов, которые встречаются во многих документах.

 *(3)*

где n\_t это число вхождений слова t в текст, а сумма – это общее число слов в рассматриваемом документе.

 *(4)*

где N – количество всех документов из набора, docFreq – число документов, где встречается слово t.

Следовательно, вес слова в документе будет определяться формулой (5):

 *(5)*

При поиске документов обычно приложение поиска информации хочет добиться высокого отзыва, что означает, что на самом деле извлечено столько релевантных документов, насколько это возможно, и высокая точность, когда доля релевантных документов в результате максимально высока. Чтобы увеличить качество, используются более широкие термины с помощью высокочастотной частоты, тогда как, с другой стороны, более точные термины лучше подходят для точности. Например, если кто-то ищет термин, который включен в каждый документ, достигается максимальный результат, но это не обязательно означает, что все они действительно полезны для пользователя. В результате этого для коэффициентов конечного веса должны учитываться факторы из обоих, частота и частота сбора. Схема TF-IDF следует этому принципу, так как термины с наибольшим весом - это те, которые часто встречаются в документе и редки над совокупностью документов [18].

Затем, после произведения расчета TD-IDF для всех документов, во всех кластерах (профессиональные отрасли) для всех параметров высчитываются среднее арифметическое. Сортируя полученный список, получаем перечень наиболее значимых слов, которые будут включены в словарь.
После того, как был получен список ключевых слов, приводится в действие процедура выявления родственных слов, описанная в предыдущем параграфе, и затем словарь ключевых слов пополняется терминами, находящимися выше по иерархии для уже полученных навыков.

Обобщая сказанное, после вычисления ключевых слов и пополнения этого списка родственными навыками на основе построенной онтологии, были составлены словари значащих слов для каждой из профессиональных отраслей. Впоследствии эти словари будут использоваться для превращения полученных вакансий в вектора, которые соотносятся только со словами из словарей.

##  Перенос вакансий в in-memory базу данных

Получив набор вакансий и переведя их в векторное представление, используя параметризированный словарь, полученный в предыдущих параграфах, возникает необходимость в выборе базы данных, где будут храниться эти вакансии.

 Учитывая, что на этапе построения рекомендации пользователю, рекомендательной системе потребуется иметь доступ к каждой вакансии из базы данных, то было бы целесообразно хранить данные в in-memory базе данных.

 In-memory база данных - это система управления базой данных, которая в первую очередь полагается на основную память для хранения компьютерных данных. Напротив, традиционные системы управления базами данных используют механизм хранения информации на дисках. Базы данных с основной памятью быстрее, чем базы данных с дисковой оптимизацией, поскольку доступ к диску медленнее, чем доступ к памяти, внутренние алгоритмы оптимизации более просты и выполняют меньше инструкций центрального процессора.

 После изучения существующих решений была выбрана NoSQL база данных MongoDB, которая имеет поддержку языка Python, и для хранения данных использует формат BSON. MongoDB представляет документы JSON в двоично-кодированном формате BSON. BSON расширяет модель JSON, чтобы предоставить дополнительные типы данных, упорядоченные поля и быть эффективным для кодирования и декодирования на разных языках.

# Глава 5. Разработка рекомендательной системы

После того, как программный комплекс скачал определенное количество вакансий, размещенных на сайте, обработал их, и преобразовал в вектора одинаковой длины, параметризированных ключевыми словами из словарей для конкретной профессиональной отрасли, пользователь загружает свое резюме в систему и ожидает получить ранжированный список из наиболее подходящих к его резюме вакансий. В этой главе рассматривается алгоритм построения рекомендации пользователю.

1.
2.
3.
4.
5.

## Обработка резюме пользователя

Подобно тому как обрабатывается каждая скачанная вакансия, обрабатывается и загруженное в программный комплекс резюме. Предоставленный соискателем документ проходит через стадию первичной обработки и преобразуется в вектор той же длины, которая задана для векторов вакансий.

Поскольку в данной работе используется векторное представление для объявления о вакансиях, то резюме кандидата должно быть преобразовано в ту же структуру. Схема обработки резюме изображена на Рисунке 7.



Рисунок 7. Схема обработки резюме пользователя.

При построении вектора кандидата учитывается такие поля, как перечень ключевых навыков и графа “обо мне”.

Более детально процедура первичной обработки и трансформации текста в векторное представление описаны в главе 4.

## Метрика подобия резюме и вакансии

Основная цель, которая преследуется при разработке рекомендательной системы для рынка труда, это построение наиболее эффективного и действенного процесса сопоставления возможностей и пожеланий кандидата на должность с требуемыми навыками в объявлении о работе. Чем больше навыков кандидата сочетается с указанными навыками в тексте объявления, тем выше оценка дается этому объявлению. Если бы рекомендательная система была основана на принципах совместной фильтрации, то не было бы возможности гарантировать, что предлагаемая работа действительно подходит текущему пользователю-кандидату, поскольку только тот факт, что похожему пользователю нравится эта работа, не может дать достаточно гарантий, что и данному кандидату также приглянется та же самая должность в этой компании.

В предлагаемой реализации рекомендательной платформы предполагается создание рекомендации только на основе содержимого предоставленного резюме и скачанных вакансий.

Несмотря на то, что в программном комплексе привлекаются дополнительные “знания”, используя разработанную подсистему таксономии, предлагаемую рекомендательную платформу нельзя отнести к категории систем на основе знаний, поскольку в системе не присутствует типичного метода сопоставления на основе знаний, например, на основе аргументации.

После того, как векторы с его условиями и весами были сгенерированы, необходимо выполнить сравнение на основе содержимого между профилем пользователя и доступными объявлениями о работе. Для поиска наиболее похожих заданий или наиболее похожих кандидатов часто применяется коэффициент Жаккара (формула (6)). Чем выше оценка, тем более актуальным является документ.

 *(6)*

где а — количество видов на первой пробной площадке, b — количество видов на второй пробной площадке, с — количество видов, общих для 1-й и 2-й площадок.

В области поиска информации известны проблемы с этим показателем сходства. Документы с большим объемом текста и в результате использования более крупных векторов имеют более высокий шанс получить высокий балл, чем более короткие документы, даже если более короткие документы могут быть релевантными.

Возможным способом, используемым в области поиска информации, является нормализация векторов по их евклидовой длине, приводящая к единичным векторам. Если эта нормализация применяется к скалярному произведению векторов, вычисляется косинус между этими двумя векторами. Оценка будет числом от 0 до 1, чем она выше, тем ближе и, следовательно, более похожи два вектора.

*α) = = (7)*

Таким образом, для оценки сходства резюме и объявления о работе используется косинусный коэффициент.

В дополнение к этому, необходимо также заранее отсеять не соответствующие требованиям объявления о работе по таким признакам, как:

* Уровень заработной платы;
* Место работы;

# Выводы

После реализации рекомендательной системы необходимо дать оценку качества получившегося продукта. В данном параграфе приводится сравнение качества выдаваемых рекомендаций пользователю с применением предложенной подсистемы онтологии и без нее. В дополнение к этому, производится замер времени, которое требуется для генерации рекомендации.

Для того, чтобы определить качество рекомендательной системы, необходимо ввести показатель эффективности системы. В области информационного поиска известно несколько подходов к проверке эффективности, которые можно разделить на две основные группы: онлайн и оффлайн.

Для оффлайн методов используются данные, полученные в результате предыдущих запусков системы, в то время как для вычисления онлайн оценок используется процедура наблюдения за пользователями, использующих систему рекомендаций в реальном времени.

Для онлайн методов решающее значение имеет доступность пользователей, которые собираются участвовать в экспериментах. При построении оценки такого вида часто применятся подход, известный как A/B тестирование.

Вычисление оффлайн оценок зависит от коллекции собранных данных. Такой подход не подразумевает активное участие пользователей с системой в реальном времени.

Перед проведением оценки необходимо определить соответствующие показатели для измерения производительности. Наиболее распространенные метрики:

* Точность (precision);
* Полнота (recall);
* F-мера;
* Выпадение (fall-out);
* Average precision;
* Precision at K;
* Mean average precision;

В качестве показателей для измерения эффективности системы в данной работе используются точность и полнота (формула (8)).

 *(8)*

где D\_rel – это набор всех релевантных документов из коллекции, а

D\_retr – это набор документов, которые нашла система.

Далее, после выбора используемой оценки эффективности, необходимо определить сценарий тестирования. В связи с невозможностью провести онлайн-тестирование представленной рекомендательной системы, в данной работе будет применен оффлайн-подход. В первую очередь, для расчета оценок precision и recall требуется определить множество релевантных объявлений о работе. Для предлагаемого сценария проверки в качестве способа построения этого множества используется ручное создание базы релевантных вакансий под несколько заранее определенных тестовых резюме. После этой процедуры были проведены вычисления метрик precision и recall для рекомендательной системы с онтологией и без нее.

Рисунок 8. Precision-recall.

Таким образом, можно сделать вывод, что введение подсистемы онтологии в работу рекомендательной системы даёт приводит к более точным результатам. Более того, аналогичный вывод можно сделать, исходя из тестового запуска программы, на вход которой подается резюме.

Требуемое время на построение рекомендации составляет несколько секунд и является приемлемым для аппаратной платформы, на которой проводились эксперименты.

Обобщая сказанное, стоит отметить, что представленная рекомендательная система показала хорошие результаты как по качеству выдаваемой пользователю рекомендации о работе, так и по времени выполнения программы.

# Заключение

В данной работе была представлена рекомендательная система для рынка труда, которая подбирает наиболее подходящие вакансии для пользователя, на основе информации, извлеченной из его резюме.

В процессе проведения исследования предметной области, были изучены существующие стратегии решения подобной задачи и выявлены критические недостатки и пробелы в альтернативных научных исследованиях. Помимо этого, в предлагаемом исследовании представлен краткий обзор современных подходов к построению рекомендательных систем, а также указаны достоинства и недостатки для каждого из них. Исходя из полученных знаний на этапе проектирования был сделан вывод использовать метод фильтрации на основе содержимого (CBRs) для построения рекомендательной системы.

Поскольку для проверки работы программного комплекса требуются тестовые данные, то были рассмотрены несколько вариантов источника таких данных. В результате, на основе ряда критериев в качестве ресурса с тестовыми данными (вакансии и резюме) был выбран сайт по онлайн-рекрутингу hh.ru.

Так как полученные из источника данные имеют слабоструктурированную форму, в представленной работе была решена задача приведения их в вид, удобный для последующего анализа с помощью программных средств. В итоге, каждая скачанная вакансия из набора, а также резюме пользователя, прошли процедуру первичной обработки и были преобразованы в векторное представление.

Учитывая приятое решение относительно формата представления данных, а именно векторное представление документов, возникла задача уменьшения размерности вектора. Для этой цели в приведенной работе введен дополнительный этап – создание словаря ключевых слов.

 Один из серьезных изъянов существующих рекомендательных система для области рынка труда, это проблема определения терминов, имеющих родственный смысл, но отличающихся своим написанием. Для решения этой задачи была предложена и реализована подсистема онтологии. Таким образом, после того, как был получен словарь из наиболее значимых слов для каждой из профессиональных отраслей, он был расширен родственными терминами с помощью разработанной подсистемы онтологии.

 При построении рекомендации не менее важным является фактор быстродействия системы в целом. Принимая во внимание количество собранных вакансий и частоту обращения к ним на этапе генерации рекомендации, было принято решение использовать в качестве хранилища in-memory базу данных, тем самым уменьшив время, требуемое для получения атрибутов вакансии при обращении к базе данных.

 Основываясь на изученной литературе, для построения ранжированного списка подходящих пользователю вакансий была предложена наиболее эффективная мера оценки сходства документов.

 В результате для функционирования предлагаемой рекомендательной системы были спроектированы и реализованы следующие программные компоненты:

* Система сбора вакансий;
* Система обработки вакансий;
* Система обработки резюме;
* Система построения онтологии;
* Система генерации рекомендаций;

# Дальнейшая работа

Несмотря на то, что предложенная стратегия построения рекомендательной системы улучшает качество результатов, несколько вопросов, выходящие за рамки этого исследования, остаются нерешенными и требуют дальнейшего изучения.

В данной работе было представлено лишь ядро рекомендательной системы, в то время как конечный пользователь привык работать с удобным графическим интерфейсом. Поэтому для удобства использования предлагаемый программный комплекс необходимо снабдить графическим интерфейсом пользователя. Это может быть мобильное приложение, приложение для рабочего стола, интегрированный виджет или полноценный сайт.

Более того, в настоящей реализации не обрабатывается образование соискателя работы, что, возможно, могло бы улучшить качество рекомендации.

В дополнение к этому, стоит отметить, что рекомендательные системы по-прежнему остаются областью с широким простором для исследования. Следует ожидать того, что в обозримом будущем могут появиться новые алгоритмы и подходы для выдачи рекомендаций пользователю. По этой причине, область исследования рекомендательных систем следует рассматривать как незавершенную задачу, решение которой может быть улучшено с появлением новых технологий.

# Список литературы

1. Charul Saxena. “Enhancing Productivity of Recruitment Process Using Data mining & Text Mining Tools”, 2011.
2. Yas A. Alsultanny, “Labor Market Forecasting by Using Data Mining”, 2013
3. W. Hong, L. Li, T. Li, and W. Pan, “ihr: An online recruiting system for xiamen talent service center,” in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD’13. New York, NY, USA: ACM, pp. 1177–1185, 2013.
4. W. Hong, S. Zheng, H. Wang, and J. Shi, “A job recommender system based on user clustering,” Journal of Computers, vol. 8, no. 8, pp. 1960– 1967, 2013.
5. Yu, K. , Guan, G. , & Zhou, M. (2005). Resume information extraction with cascaded hybrid model. In Proceedings of the 43rd annual meeting on association for com- putational linguistics (pp. 499–506). Association for Computational Linguistics.
6. Cunningham, H. , Maynard, D. , Bontcheva, K. , & Tablan, V. (2002). A framework and graphical development environment for robust nlp tools and applications. In ACL (pp. 168–175)
7. Chang, A. X. , & Manning, C. D. (2014). TOKENSREGEX: Defining cascaded regular ex- pressions over tokens. Technical Report CSTR 2014-02 . Department of Computer Science, Stanford University
8. Nikolaos D. Almalis, Prof. George A. Tsihrintzis and Nikolaos Karagiannis “A Content Based Approach for Recommending Personnel for Job Positions”, The 5th IEEE International Conference on Information, Intelligence, Systems and Applications, Bijing china, pp. 45 – 49, 27-29 june 2014.
9. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of ACM, Vol. 18, No. 11, pp. 613 – 620, 1975.
10. G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," Springer, pp. 257–298, 2011.
11. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” IEEE Trans. on Knowl. and Data Eng., vol. 17, pp. 734–749, June 2005
12. X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Adv. in Artif. Intell., vol. 2009, pp. 4:2–4:2, Jan. 2009.
13. A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla,“Prospect: A system for screening candidates for recruitment,” in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ser. CIKM ’10. New York, NY, USA: ACM, pp. 659–668, 2010.
14. Keith Bradley, Rachael Rafter, and Barry Smyth. Case-based user profiling for content personalisation. In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, pages 62–72. Springer-Verlag, 2000.
15. Danielle H. Lee and Peter Brusilovsky. Fighting information overflow with personalized comprehensive information access: A proactive job recommender. In Proceedings of the Third International Conference on Autonomic and Autonomous Systems, pages 21–, Washington, DC, USA, 2007. IEEE Computer Society.
16. Jochen Malinowski, Tobias Keim, Oliver Wendt, and Tim Weitzel. Matching people and jobs: A bilateral recommendation approach. In Proceedings of the 39th Annual Hawaii International Conference on System Sciences - Volume 06, page 137c, Washington, DC, USA, 2006. IEEE Computer Society.
17. Gerard Salton and Chris Buckley. Technical report.//Term weighting approaches in automatic text

 retrieval, 1987