|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»    [*Факультет социально-экономических и компьютерных наук*](https://perm.hse.ru/scs/) |
| Белов Егор Александрович  **РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ВЫЧИСЛЕНИЯ МЕТРИК В СОЦИАЛЬНЫХ МЕДИА**  *Курсовая работа*  студента образовательной программы «Программная инженерия» по направлению подготовки *09.03.04 Программная инженерия*  Руководитель  к.ф.-м.н., доцент, доцент кафедры информационных технологий в бизнесе  Е.Б. Замятина |

Пермь, 2024 год

Аннотация

Автор: Белов Егор Александрович.

Название работы: Разработка приложения для вычисления метрик в социальных медиа.

Данная работа содержит процесс анализа и разработки приложения для расчета калорийности продуктов и блюд. Курсовая работа содержит 58 страницу и 5 глав. В данной работе содержится 22 рисунков, 8 приложений.

Оглавление

[Введение 4](#_Toc159967155)

Введение

2 части вирутальная сеть и реальная

Работа посвящена созданию приложения для вычисления метрик в социальных медиа.

В современном мире социальные сети стали неотъемлемой частью жизни людей. Ежедневно миллионы пользователей проводят в них время, общаясь с друзьями, делясь новостями и интересами, а также потребляя разнообразный контент. В этом контексте социальные медиа выступают важной платформой для бизнеса, который активно использует их для продвижения своих товаров и услуг.

Реклама в социальных сетях, будучи мощным инструментом, требует точного таргетирования, чтобы достичь максимальной эффективности. Анализ поведения пользователей и вычисление релевантных метрик становятся необходимыми компонентами успешной рекламной стратегии. В данном контексте разработка приложения, способного автоматизировано выявлять перспективных пользователей для рекламы, представляет собой важный шаг в оптимизации маркетинговых усилий.

В этом свете, настоящая курсовая работа ориентирована на создание инструмента, который не только позволит вычислять метрики в социальных медиа, но и определять наиболее релевантных пользователей для целевой рекламы. Такой подход сделает рекламные кампании более таргетированными и эффективными, отвечая на вызовы современной маркетинговой парадигмы.

Оценка эффективности рекламных кампаний в социальных сетях невозможна без отслеживания ключевых метрик.

В контексте разработки приложения для вычисления метрик в социальных медиа представлены следующие направления научной новизны:

* Разработка алгоритма для поиска наиболее релевантных для рекламы пользователей.
* Использование модели виртуальной сети в системе информационного моделирования.
* Использование онтологического подхода, который позволяет сохранять знания о пользователях социальных сетей, их интересов и связей.

Объектом курсовой работы будет исследование социальных медиа.

Предметом данной работы являются социальные метрики, которые необходимо проанализировать.

Разработка приложения для автоматического сбора и анализа данных о пользователях социальных сетей позволит:

* Сократитьвремяитрудозатраты **-** приложение будет автоматически собирать данные, что позволит пользователям экономить время и силы.
* Повыситьэффективностьрекламныхкампаний **-** приложение позволит пользователям более точно таргетировать рекламные объявления, что приведет к повышению их эффективности.
* Снизитьрасходы **-** приложение будет бесплатным, что позволит пользователям экономить деньги на использовании сторонних инструментов.

Основная цель работы заключается в создании инструмента, способного автоматизированно собирать информацию о пользователях социальной сети, вычислять различные метрики и на основе полученных данных определять наиболее влиятельных пользователей для лучшего распространения рекламы. Этот подход позволит повысить эффективность маркетинговых кампаний, сделав их более таргетированными и направленными на целевую аудиторию.

Кроме того, в рамках работы предусмотрено создание модели виртуальной сети в системе информационного моделирования. Это дополнительный инструмент, способствующий более глубокому анализу взаимосвязей между пользователями, их интересами и поведенческими паттернами. Создание такой модели позволит более точно определить сегменты аудитории, на которые следует ориентироваться при проведении рекламных кампаний.

Цель данной работы достигается выполнением следующих задачи:

* Провести анализ существующих методов сбора данных из социальных сетей.
* Разработать алгоритмы для вычисления ключевых метрик пользователей социальных сетей.
* Разработать алгоритм для поиска наиболее релевантных для рекламы пользователей.
* Использовать модель виртуальной сети в системе информационного моделирования и создать ее веб-интерфейс.
* Реализовать приложение, используя выбранные методы и алгоритмы.
* Провести тестирование приложения и анализ его эффективности.

Таким образом, данная работа представляет собой комплексный подход к оптимизации рекламных стратегий в социальных медиа, разделенный на несколько ключевых этапов. В первую очередь, в ней реализуется разработка приложения для вычисления метрик, необходимых для анализа эффективности рекламных кампаний. Далее, создается модель виртуальной сети в системе информационного моделирования, которая включает в себя визуализацию виртуальной сети. Кроме того, проводится анализ реальной сети для вычисления метрик на основе фактических данных. Наконец, осуществляется сбор информации в онтологии и отслеживание через журналы событий, что позволяет более полно и точно анализировать результаты и принимать обоснованные решения по оптимизации рекламных стратегий.

Глава 1. Анализ

В данной главе рассматривается начальный этап работы, который включает анализ теоретических материалов, освещающих особенности изучения процесса распространения информации в социальных сетях с точки зрения подходов к моделированию социальных сетей и процессов распространения, а также учет характеристик, присущих исследуемому процессу и пользователям, которые являются его участниками.

1.1. Использование социальных сетей для распространения информации

Интернет — это важная часть современной жизни. Он стал незаменимым средством коммуникации, источником информации, площадкой для самовыражения и многое другое. Большинство людей пользуются интернетом ежедневно и практически ни на что не могут обойтись без его помощи. Так что же делает интернет таким популярным?

Во-первых, интернет предоставляет доступ к огромному количеству информации. Он является огромной библиотекой знаний, где можно найти ответы на любые вопросы, изучить новые темы и получить актуальные новости. Люди используют интернет для образования, самообразования и развлечения. Благодаря интернету, информация стала легко доступной и распространяется во всемирном масштабе.

Во-вторых, интернет обеспечивает возможность связи и общения. Благодаря социальным сетям, мессенджерам и электронной почте, мы можем легко общаться с друзьями, родственниками и коллегами, независимо от расстояния. В интернете можно также найти единомышленников, присоединиться к группам или сообществам с общими интересами. Это открывает огромные возможности для социального взаимодействия и создания новых связей.

Социальные сети предоставляют пользователям возможность создавать персональные профили, заполнять их информацией о себе, загружать фотографии и видео, а также делиться своими мыслями и впечатлениями в формате постов.

Комментарии и лайки к постам позволяют людям выражать свое мнение, а также демонстрировать свою поддержку или несогласие с высказываниями других пользователей.

Общение через интернет позволяет преодолеть пространственные и временные ограничения. Люди могут общаться с друзьями и родственниками, находящимися в других городах или даже странах, без необходимости путешествовать и тратить деньги на звонки или почтовые отправления.

Социальные сети предоставляют возможность общаться не только с близкими людьми, но и находить интересные группы и сообщества, в которых можно обсуждать общие интересы, делиться полезной информацией, получать советы от экспертов и просто общаться с единомышленниками.

Реклама в социальных сетях предоставляет предпринимателям ряд возможностей. Размещение информации в определенной сети позволяет создать положительный имидж компании, сэкономить средства, так как реклама в интернете является наименее затратным способом продвижения товаров и услуг, а также направить рекламу на целевую аудиторию, которая уже заинтересована в покупке. Преимущества размещения рекламы в социальных сетях включают доступ к личной информации пользователей, такой как пол, возраст, образование, хобби и т.д. Это позволяет посещать страницы пользователей, составляющих целевую аудиторию, и узнавать их мнение о товарах и услугах компании. Пользователи также могут делиться информацией между собой, создавая "вирусный маркетинг".

Другие преимущества использования социальных сетей в качестве платформы для размещения рекламы включают:

* наличие большой аудитории;
* возможность более быстрого продвижения по сравнению с другими способами;
* низкие затраты или их отсутствие;
* возможность общения с потенциальными клиентами прямо на базе социальных сетей, используя их простые и доступные всем пользователям инструменты.

Среди пользователей социальных сетей можно выделить влиятельных лидеров мнений. Часто в маркетинге используется стратегия, включающая знаменитостей для распространения информации. Обычные пользователи чаще доверяют информации, полученной от таких лидеров. Это означает, что маркетинговое воздействие распространяется не только на самих знаменитостей, но и на их многочисленных последователей, увеличивая эффективность кампании. Для использования этого инструмента маркетинга можно применять различные методы влияния на лидеров мнений, такие как материальное вознаграждение или бартерные сделки.

Целью данной работы является вычисление метрик, которые будут определять наиболее влиятельного и релевантного пользователя, благодаря которому распространение рекламы будет наиболее оптимально и эффективно.

1.2. Анализ метрик в социальных медиа

Метрики позволяют отслеживать эффективность работы в социальных сетях и планировать выпуск контента или проведение рекламных кампаний.

Взаимная направленность — это свойство, которое указывает на то, является ли отношение между вершинами бинарным. Другими словами, это свойство показывает, является ли связь двунаправленной. В контексте социальных сетей, например, взаимная направленность может означать, что пользователь подписан на другого пользователя и этот другой пользователь также подписан на него. Это свойство может быть полезным для анализа связей и отношений между пользователями в социальной сети. Взаимная направленность между вершинами в графе может быть определена с использованием формулы коэффициента взаимной направленности (Reciprocity coefficient). Формула для вычисления коэффициента взаимной направленности (R) выглядит следующим образом:

R=A+B/2⋅T,

Где:

* *T* - количество двунаправленных связей (отношений), то есть связей, где вершина A связана с вершиной B и наоборот;
* *A* - количество связей, исходящих из вершины A;
* *B* - количество связей, входящих в вершину B.

Коэффициент взаимной направленности R принимает значения от 0 до 1. Значение ближе к 1 указывает на более высокую взаимную направленность между вершинами, что означает более сильные взаимосвязи и отношения в графе.

​Гомогенность — это свойство, которое указывает на степень появления связей между акторами, которые имеют схожие характеристики, такие как пол, возраст или интересы. В контексте социальных сетей, гомогенность может означать, что пользователи с похожими интересами или предпочтениями чаще взаимодействуют друг с другом, например, подписываются на одни и те же страницы или группы, комментируют и лайкают посты друг друга. Гомогенность в графе социальной сети может быть оценена с помощью коэффициента гомогенности (Homophily coefficient). Формула для расчета коэффициента гомогенности (H) выглядит следующим образом:

H = T / N,

Где:

* *T* - количество связей между акторами с схожими характеристиками (например, с одинаковыми интересами);
* *N* - общее количество связей в графе.

Коэффициент гомогенности *H* также принимает значения от 0 до 1. Значение ближе к 1 указывает на более высокую гомогенность в графе, что означает, что акторы с схожими характеристиками взаимодействуют чаще, чем акторы с различными характеристиками.

Транзитивность связей — это свойство, которое указывает на увеличение вероятности появления связей между акторами, у которых есть связи с одними и теми же вершинами. Другими словами, если у актора А есть связь с актором В, и у актора В есть связь с актором С, то вероятность того, что у актора А будет связь с актором С, увеличивается. В контексте социальных сетей, это может означать, что если пользователь А подписан на пользователя В, и пользователь В подписан на пользователя С, то вероятность того, что пользователь А также подпишется на пользователя С, возрастает. Транзитивность связей в графе социальной сети может быть измерена с помощью коэффициента транзитивности (Transitivity coefficient). Формула для расчета коэффициента транзитивности (T) выглядит следующим образом:

T = 3\* количество замкнутых троек / количество троек,

Где

* Количество замкнутых троек - это количество троек вершин (A, B, C), где существуют связи (A, B), (B, C) и (A, C);
* Количество троек - общее количество троек вершин в графе.

Коэффициент транзитивности T также принимает значения от 0 до 1. Значение ближе к 1 указывает на более высокую транзитивность связей в графе, что означает более вероятное появление связей между акторами, у которых уже есть общие связи.

Разница в распределении — это свойство, которое указывает на то, что у одних акторов есть большое количество связей, а у других - минимальное. Важным в данном случае является феномен "богатый становится богаче", который приводит к высокой дисперсии вершин. В контексте социальных сетей, это может означать, что некоторые пользователи имеют большое количество подписчиков и взаимодействий, в то время как другие пользователи имеют минимальное количество подписчиков и взаимодействий. Разница в распределении связей в графе социальной сети может быть оценена с помощью различных метрик, таких как коэффициент Гини (Gini coefficient) или коэффициент Робин Гуда (Robin Hood index).

Коэффициент Гини измеряет степень неравенства в распределении связей между акторами в графе. Чем ближе коэффициент Гини к 1, тем выше неравенство в распределении связей. В контексте социальных сетей, высокий коэффициент Гини может указывать на явление "богатый становится богаче", где некоторые пользователи имеют значительно большее количество связей, чем другие.

Коэффициент Робин Гуда, наоборот, оценивает, сколько связей необходимо перераспределить с богатых акторов (тех, у которых много связей) на бедных акторов (тех, у которых мало связей), чтобы достичь более равномерного распределения. Чем выше коэффициент Робин Гуда, тем больше связей необходимо перераспределить.

Использование этих метрик позволяет оценить степень неравенства в распределении связей между акторами в социальной сети и выявить феномен "богатый становится богаче".

Центральность — это метрика, которая позволяет определить значительность или влияние определенного узла или группы в сети. В контексте социальных сетей, центральность может указывать на то, насколько влиятельным является пользователь или группа пользователей. Например, пользователь с высокой центральной позицией может иметь большое количество подписчиков, получать много лайков и комментариев, а также иметь большое количество связей с другими пользователями. Центральность в графе социальной сети может быть оценена с помощью различных метрик:

* **Центральность по степени (degree centrality)**- эта метрика определяет, насколько узел важен, основываясь на количестве связей, которые он имеет с другими узлами. Пользователь с высокой центральностью по степени обычно имеет большое количество подписчиков и/или связей с другими пользователями.
* **Центральность по посредничеству (betweenness centrality)** - эта метрика оценивает, насколько узел важен для коммуникации между другими узлами в сети. Пользователь с высокой центральностью по посредничеству может играть ключевую роль в передаче информации или контроле над потоком коммуникации в сети.
* **Центральность по близости (closeness centrality)** - эта метрика измеряет, насколько близко узел к другим узлам в сети. Пользователь с высокой центральностью по близости обычно может быстро достичь других пользователей в сети и имеет прямой доступ к большому количеству информации.

Ассортативность — это свойство, которое указывает на склонность к образованию связей между вершинами большой степени. В контексте социальных сетей, ассортативность может означать, что пользователи с большим количеством подписчиков или взаимодействий чаще взаимодействуют друг с другом, чем с пользователями с меньшим количеством подписчиков или взаимодействий.

Диаметр сети — это метрика, которая указывает на максимальное количество шагов, необходимых для того, чтобы соединить две самые удаленные вершины в сети. В контексте социальных сетей, диаметр сети может указывать на то, насколько быстро информация может распространяться между пользователями. Например, если диаметр сети небольшой, то информация может быстро распространяться между пользователями, в то время как если диаметр сети большой, то информация может распространяться медленнее.

PageRank - это алгоритм, разработанный компанией Google, который используется для оценки важности веб-страниц на основе структуры ссылок между ними. Этот алгоритм также может быть применен к другим типам сетей, включая социальные сети, для определения значимости узлов в сети.

Формула PageRank выглядит следующим образом:

*PR*(*u*)=(1−*d)/N* ​ +*d* ∑*v*∈ *Bu* ​​ *PR*(*v*) / *L*(*v*),​

Где:

* *PR*(*u*) - PageRank узла *u*;
* *d* - коэффициент демпфирования, который обычно принимает значение около 0.85;
* *N* - общее количество узлов в сети;
* *Bu*​ - множество узлов, которые ссылается на узел *u*;
* *L*(*v*) - количество исходящих ссылок узла *v*.

Эта формула описывает итеративный процесс, в котором каждый узел получает оценку PageRank, основанную на оценке значимости узлов, ссылающихся на данный узел, и их собственной оценке PageRank. Алгоритм продолжает итерации до тех пор, пока изменения оценок PageRank между итерациями не станут достаточно малыми.

PageRank позволяет определить важность узлов в сети на основе их связей с другими узлами, что делает его мощным инструментом для анализа структуры сетей и идентификации ключевых узлов.

В этом разделе были представлены различные метрики и характеристики, позволяющие анализировать структуру и динамику социальных сетей.

Использование этих метрик может помочь в разработке эффективных маркетинговых стратегий, изучении распространения информации, обнаружении лидеров мнений и моделировании социальных сетей.

1.3. Назначение системы

Данная работа содержит три основные части, такие как создание модели виртуальной сети в системе информационного моделирования, создание базы знаний в онтологии и отслеживание информации в журнале событий, реализация сервиса по сбору информации о пользователях социальной сети и подсчет метрик.

1.4. Виртуальные и реальные сети, какие есть

1.5. Онтологии

Для исследования социальной сети первоочередно требуется анализ информационных каналов, то есть, исследование того, как пользователи взаимодействуют друг с другом и откуда они получают информацию. Эффективным методом представления социальной сети является использование графа, где узлы представляют собой пользователей или сообщества, а рёбра отображают каналы связи между ними.

Такой подход обладает рядом преимуществ:

* Позволяет выявить различные типы взаимосвязей между участниками.
* Пользователи описываются с помощью атрибутов: каждый человек или группа имеют свои характеристики.
* Совокупность разнообразных связей между участниками формирует структуру сети.
* Некоторые участники могут быть более тесно связаны друг с другом, чем с остальными.

Таким образом, становится ясным, что важно сохранить информацию о структуре социальной сети. Каждый элемент этой сети может быть представлен как узел, который связан с другими элементами и имеет свои характеристики. Однако необходимо, чтобы эту информацию могли понять как люди, так и компьютеры. Здесь возникает вопрос: в каком формате следует хранить информацию о реальной социальной сети?

Для решения этой задачи было предложено применять онтологии.

Онтология — это детальное определение концепции или предметной области. Под детальностью понимается способ представления понятий и их определений таким образом, чтобы они могли быть однозначно поняты как человеком, так и компьютером. Современные онтологии можно рассматривать как систему, состоящую из набора <A, S, T>, где:

* T - набор концепций или понятий, принадлежащих предметной области;
* S - набор различных типов связей, поддерживаемых в этой области;
* A - отображение между элементами из набора T и S.

Онтологии служат своего рода словарем для представления и обмена знаниями в определенной предметной области, а также определяют связи между терминами в этом словаре. Можно также представить онтологию в виде графа.

Глава 2. Проектирование

2.1. Бизнес-процессы, прецеденты ????

2.1. Системы имитационного моделирования

Информационное моделирование систем (SIM) представляет собой процесс создания моделей сложных взаимосвязанных систем. Информационные модели систем - это цифровые аналоги связанных систем, таких как приборы и системы управления, электроснабжение и связь. Объекты, смоделированные в SIM, соответствуют объектам физической системы в пропорции 1:1. Компоненты, связи и функции определяются и моделируются так, как они существуют в реальном мире.

SIM играет важную роль в инженерии, науке и различных отраслях промышленности по нескольким причинам:

* Системы информационного моделирования позволяют инженерам и дизайнерам создавать цифровые модели сложных систем до их физической реализации. Это помогает сократить время и затраты на разработку новых продуктов и технологий, а также идентифицировать потенциальные проблемы ещё на этапе проектирования.
* Используя СИМ, исследователи могут анализировать поведение системы в различных условиях и оптимизировать её работу. Это позволяет предвидеть и управлять различными аспектами системы, такими как производительность, энергопотребление, надежность и безопасность.
* SIM позволяет проводить виртуальные тесты и симуляции, которые могут быть недоступны или слишком дороги в реальной среде. Это позволяет тестировать новые идеи, стратегии и процессы без риска повреждения оборудования или угрозы для безопасности.
* Путем создания моделей различных сценариев и прогнозирования их последствий, SIM помогает оценивать и управлять рисками, связанными с различными аспектами системы. Это может быть особенно важно в критических отраслях, таких как медицина, авиация и энергетика.
* SIM также широко используется в образовательных целях для обучения студентов и профессионалов в различных областях науки, техники и управления. Виртуальные симуляции позволяют студентам получать практический опыт работы с реальными системами без необходимости доступа к дорогостоящему оборудованию.

Таким образом, Системы информационного моделирования являются мощным инструментом, который помогает улучшить процессы проектирования, анализа, тестирования и управления различными системами, что способствует развитию науки, технологии и промышленности.

Для данной работы была выбрана система информационного моделирования AnyLogic.

AnyLogic — это мощная платформа для создания многопроцессорных динамических системных моделей, используемая для симуляции и анализа процессов в различных областях, включая производство, логистику, транспорт, здравоохранение, финансы и многое другое.

Преимущества AnyLogic:

* Бесплатная лицензированная версия: Доступна Personal Learning Edition.
* Применение агентного моделирования - возможность построения модели социальной сети с использованием агентного подхода.
* Возможность создания модели случайной безмасштабной сети с использованием алгоритма, основанного на принципе предпочтительного присоединения.
* Моделирование процесса распространения контента - возможность использования моделей эпидемии, таких как SI, SIR и SEIR, для анализа распространения информации.
* Включение параметров активности и влияния - возможность учета публикационной активности и уровня воздействия в модели.
* Визуализация динамики распространения информации - возможность наглядного отображения процесса распространения контента в социальной сети.

Глава 3. Моделирование системы виртуальной сети

Глава 4. Реализация приложения

В данной главе будут описаны процессы реализации сервисов для сбора информации и вычисления метрик, а также приложения для просмотрарезультатов.

4.1. Выбор средств реализации

В качестве СУБД для хранения информации о пользователя была выбрана PostgreSQL. PostgreSQL — это реляционная система управления базами данных (РСУБД) с открытым исходным кодом, которая заслужила признание благодаря своей надежности, стабильности и гибкости. Широкое использование PostgreSQL в различных сферах делает ее предпочтительным выбором для множества разработчиков и организаций.

В качестве языка программирования был выбран Python, так имеет множество полезных возможностей:

* В экосистеме Python существует множество библиотек и фреймворков для веб-разработки, включая Flask, Django, Pyramid и другие. Это позволяет разработчикам выбирать инструменты в зависимости от требований и предпочтений проекта.
* Python имеет простой и понятный синтаксис, что делает его доступным для новичков и облегчает работу опытным разработчикам.
* Python широко применяется в различных областях разработки, включая веб-разработку, науку о данных, искусственный интеллект, автоматизацию и другие. Это обеспечивает разработчикам возможность использовать один язык для различных задач и проектов.
* Python предоставляет разработчикам множество удобных инструментов для работы с API и базами данных, что делает его предпочтительным выбором для создания приложений, взаимодействующих с внешними сервисами и хранилищами данных.

Также для вывода данных и удобного развертывания системы был выбрал фреймворк Flask.

Для взаимодействия и отображения онтологий будет использоваться Protégé - Бесплатный редактор онтологий с открытым исходным кодом и фреймворк для создания интеллектуальных систем.

4.2. Реализация сервиса для сбора информации

В данном разделе представлена реализация скрипта, который представляет собой Python-скрипт, который взаимодействует с социальной сетью ВКонтакте (VK) и выполняет различные задачи, такие как сбор данных о пользователях и группах, анализ действий пользователей и обновление онтологии полученной информацией.

Изначально необходимо определить сущности, которые будут отслеживаться, то есть пользователь, сообщество и пост, то есть класс Person, Community и Post.

Следующее, что необходимо сделать сформировать методы для взаимодействия с VK API, для этого создан класс VK. Данный класс обрабатывает взаимодействие с VK API, включая получение данных о пользователях, сообществах и записях, проверку статуса онлайн пользователя и получение понравившихся записей.

Для реализации класса VK будет использоваться официальная билиотека для работы с методами данной социальной сети – vk\_api. Необходимые для работы методы:

* users.get - этот метод позволяет получить полную информацию о пользователе по его идентификатору. В скрипте используется для извлечения данных о пользователе, таких как имя, фамилия, пол, дата рождения, фотография и другие атрибуты.
* groups.get - используется для получения списка идентификаторов групп, в которых состоит пользователь. Эта информация может быть полезна для анализа социальной сети пользователя и его активности в различных группах.
* groups.getById - этот метод получает полную информацию о группе по ее идентификатору. В скрипте используется для извлечения данных о сообществе, таких как название, активность и другие характеристики.
* friends.get - позволяет получить список идентификаторов пользователей, с которыми выбранный пользователь состоит в дружеских отношениях. В скрипте этот метод используется для получения списка друзей пользователя.
* wall.get - используется для получения списка последних постов на стене пользователя или сообщества. В скрипте этот метод используется для извлечения последних постов и анализа активности пользователей и групп.
* likes.isLiked - этот метод проверяет, добавил ли пользователь конкретный пост в список понравившихся. В скрипте он используется для проверки, понравился ли определенный пост пользователю.

Далее реализуем класс Ontology, который управляет созданием и сохранением онтологии, а также предоставляет методы для сохранения информации о пользователях, сообществах и записях в онтологию. Данный класс содержит следующие методы:

1) Метод \_\_init\_\_:

* Инициализирует объект класса Ontology.
* Принимает путь к файлу онтологии.
* Загружает онтологию из указанного файла и инициализирует объект блокировки threading.Lock().

2) Метод create\_ontology:

* Создает структуру онтологии, определяя классы и свойства для хранения информации о пользователях, сообществах и их взаимодействии.
* Определяет классы **Person**, **Community**, **Activity** и **Post**, а также свойства для хранения информации о них, такие как имя, фамилия, пол, фотография, активность и т.д.
* Сохраняет онтологию в файл по указанному пути.

3) Метод save\_person:

* Сохраняет информацию о пользователе в онтологию.
* Создает экземпляр класса **Person** и заполняет его атрибуты данными о пользователе (имя, фамилия, пол, дата рождения, фотография и т.д.).
* Добавляет информацию о друзьях и подписках пользователя.
* Сохраняет изменения в онтологию.

4) Метод save\_community:

* Сохраняет информацию о сообществе в онтологию.
* Создает экземпляр класса **Community** и заполняет его атрибуты данными о сообществе (название, активность и т.д.).
* Сохраняет изменения в онтологию.

5) Метод save\_post:

* Сохраняет информацию о посте в онтологию.
* Создает экземпляр класса **Post** и заполняет его атрибуты данными о посте (владелец, теги, репосты и т.д.).
* Проверяет, является ли владелец поста пользователем или сообществом, и сохраняет соответствующую информацию.
* Сохраняет изменения в онтологию.

6) Методы post\_viewed и post\_liked:

* Добавляют информацию о просмотре и лайке поста в онтологию.
* Создают связи между пользователями и постами, указывая, кто посмотрел пост и кто поставил лайк.
* Сохраняют изменения в онтологию.

Класс NewsFeed управляет глобальной лентой новостей, храня записи для каждого владельца (пользователя или сообщества) и предоставляет методы для добавления и извлечения записей.

Класс EventLog управляет ведением журнала событий, включая добавление трассировок, запись журналов в файлы и определение различных типов событий, таких как статус онлайн, добавление, просмотр, лайк и копирование записей.

Классы GroupEventMiner и UserEventMiner наследуются от класса Thread. Метод run() предоставляет возможность запуска асинхронного сбора информации для определенного пользователя или сообщества.

Диаграмма классов представлена на рисунке 4.1.

Для обеспечения асинхронной и распределенной обработки задач или сообщений будет использоваться RabbitMQ. RabbitMQ является брокером сообщений, который позволяет различным компонентам приложения (например, микросервисам или рабочим потокам) обмениваться данными или задачами через очереди сообщений.

При запуске определяются параметры, такие как интервалы времени для различных действий, целевые пользователи, целевые группы, а также инициализируются онтология, API VK, журнал событий и лента новостей.

Настраивается подключение к RabbitMQ и объявляются необходимые обменники и очереди.

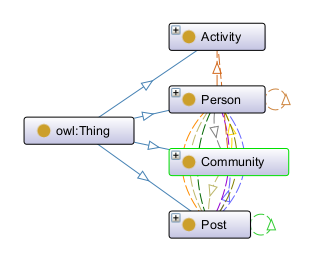
Определяется функция обратного вызова для обработки входящих сообщений от RabbitMQ, которая получает данные о пользователе или группе из VK, запускает добытчики событий для пользователей и групп, обновляет онтологию полученными данными и публикует сообщения для обновления сущностей.

Наконец, начинается потребление сообщений от RabbitMQ и запускается поток записи журнала событий.

Для запуска агента используются данные из файлов users.txt и groups.txt. В каждом из этих файлов содержатся уникальные идентификаторы пользователей и групп социальной сети. При работе агента можно расширить список групп, добавив новый идентификатор в очередь сообщений.

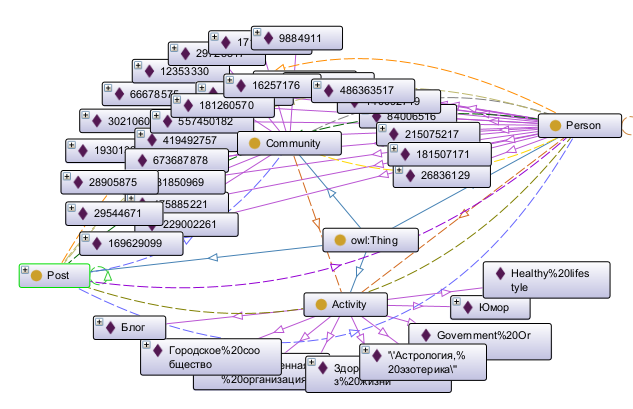
Скрипт читает данные из двух текстовых файлов: users.txt и groups.txt. Каждый файл содержит идентификаторы пользователей или групп, которые будут отправлены в RabbitMQ для обработки. Далее создается соединение с сервером RabbitMQ и открывается канал для обмена сообщениями. После чего происходит отправка сообщений в очередь RabbitMQ. Каждый элемент из списка groups и users преобразуется в формат JSON и отправляется в очередь с указанным обменником ("social\_data") и пустым ключом маршрутизации.

Результатом работы программы является онтология и журнал событий. Структура онтологии представлена на рисунке 4.2.

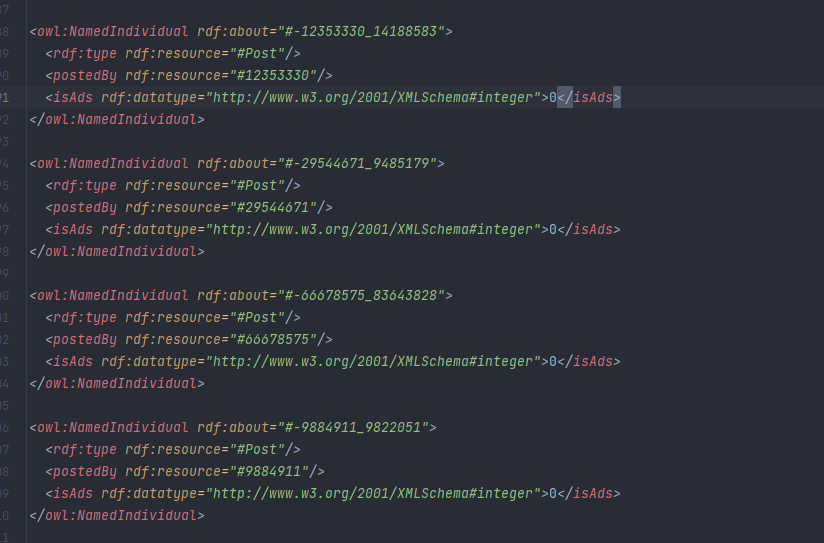


***Рисунок 4.2. Структура онтологии***

Полученная онтология представлена на рисунках 4.3 и 4.4.

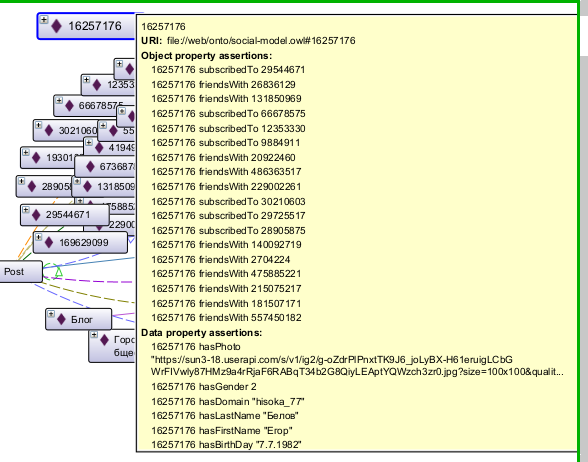


***Рисунок 4.3. Результат онтологии***



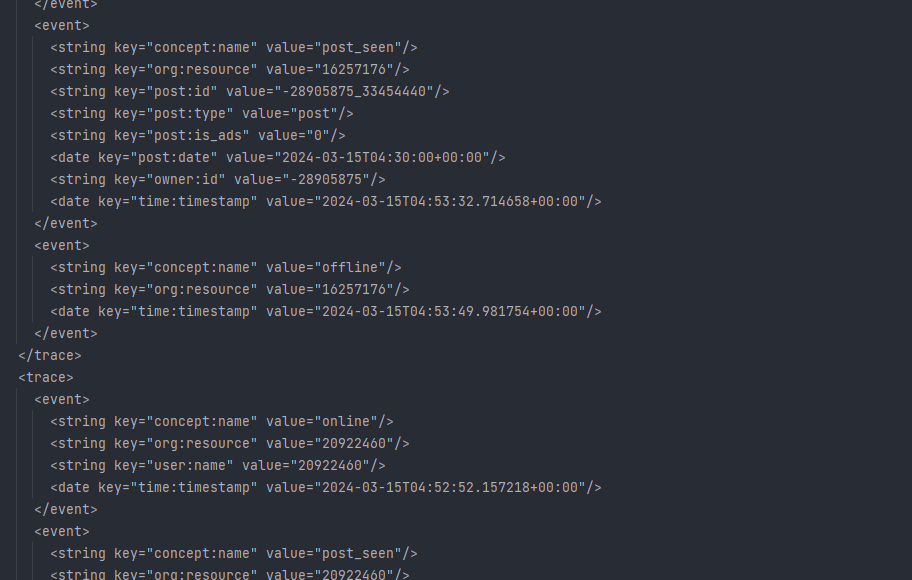
***Рисунок 4.4. OWL код полученной онтологии***

На рисунке 4.5 представлена индивидуальная информация о пользователе и его связях с другими сущностями.



***Рисунок 4.5. Индивидуальная информация***

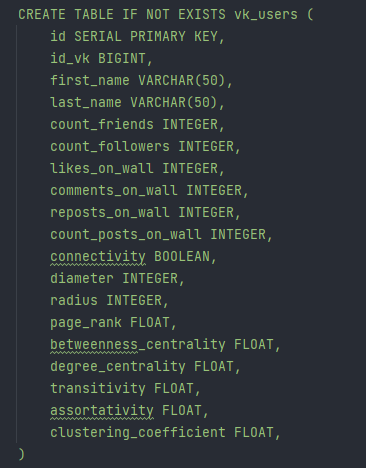
Также был сформирован журнал событий, представленный на рисунке 4.6.



***Рисунок 4.6. Журнал событий***

4.3. Реализация сервиса для вычисления метрик

Сначала необходимо создать базу данных, в которой будет храниться информация о пользователях и его метриках. SQL-запрос на создание БД представлен на рисунке 4.7.



***Рисунок 4.7. Создание БД***

База данных будет содержать информацию о пользователях, такую как: ID ВК, Имя, Фамилия, Количество друзей и подписчиков, Количество лайков, репостов и комментариев на стене пользователя, Общее количество постов на стене, а также метрики: Связность, Диаметр, Радиус, PageRank, Центральность, Транзитивность, Ассортативность, Коэффициент кластеризации.

Далее необходимо реализовать методы для взаимодействия с VK APi, такие как:

* request\_count\_followers – метод, для подсчета количества подписчиков, использующий метод API users.getFollowers;
* request\_count\_friends – подсчет количества друзей с помощью friends.get;
* request\_count\_likes\_from\_wall – метод для расчета количества лайков, комментариев, репостов на стене пользователя и общее количество постов, используя wall.get;
* request\_users\_get – получение информации о пользователях с помощью users.get.

Далее заполняем БД с помощью данных методов.

Следующим шагом является создание графа друзей определенного пользователя, то есть графа реальной сети. С помощью этого графа будет возможно вычислить метрики, встроенные Python библиотеки.

Определяем класс VkFriends. Этот класс содержит методы для работы с VK API и получения информации о пользователе, его друзьях и общих друзьях.

**Метод** request\_url - этот метод формирует URL-адрес запроса к API VK на основе переданных параметров.

**Метод** base\_info - этот метод используется для получения основной информации о пользователе, такой как его имя, фамилия и фото.

**Метод** friends - этот метод используется для получения списка друзей пользователя.

**Метод** common\_friends - этот метод используется для поиска общих друзей текущего пользователя с его друзьями. Метод делит список друзей на части по 25 человек и для каждой части запрашивает общих друзей с помощью метода friends.getMutual API VK. Результаты обработки сохраняются в список result в виде кортежей с информацией о пользователе и его общих друзьях. Если возникает ошибка при запросе общих друзей, она выводится на экран, и процесс продолжается. Метод возвращает список кортежей с информацией о пользователе и его общих друзьях.

Класс D3 является подклассом класса VkFriends. Он используется для генерации JSON-файла, который можно использовать для визуализации социального графа с помощью библиотеки D3.js.

В конструкторе вызывается конструктор родительского класса VkFriends, чтобы инициализировать экземпляр с использованием переданных аргументов.

self.friendships инициализируется результатом вызова метода common\_friends, который находит общих друзей текущего пользователя с его друзьями.

self.js инициализируется пустым словарем, который будет содержать узлы и ребра графа.

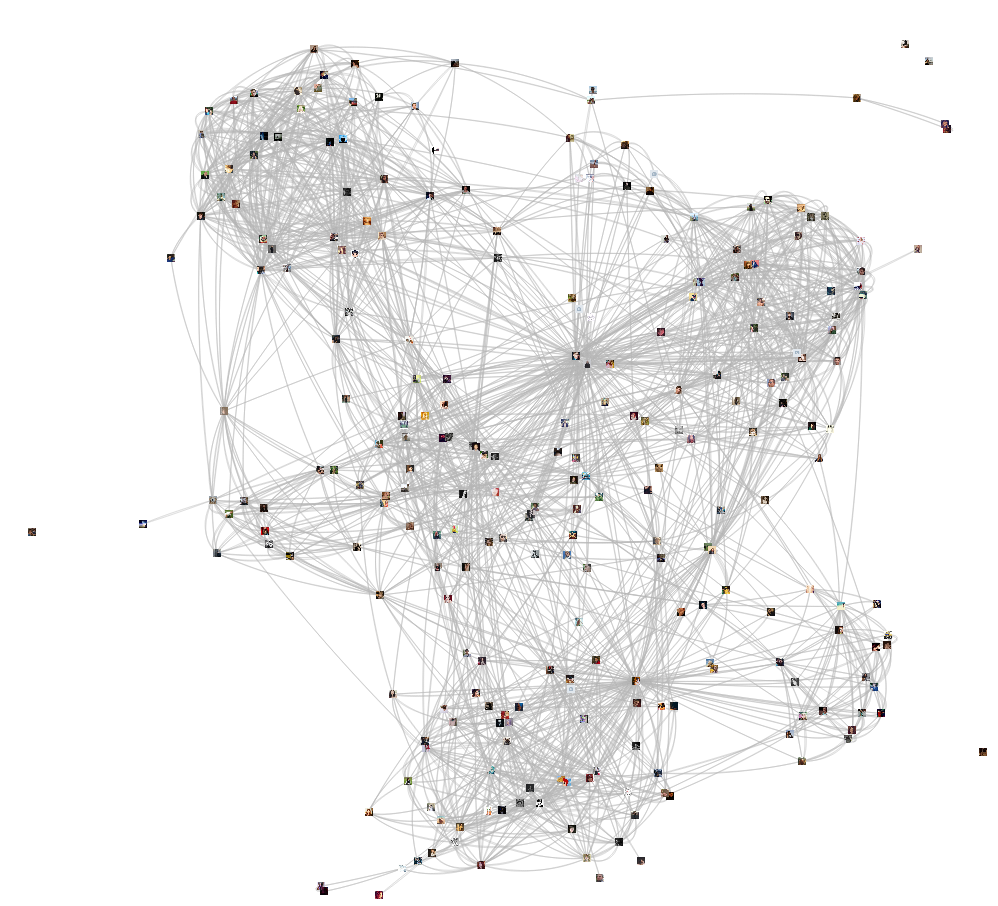
self.dict\_friends инициализируется результатом вызова метода to\_json(), который преобразует данные о друзьях в JSON-формат.

JSON-данные записываются в файл с помощью метода write\_json().

Диаграммы классов представлены на рисунке 4.8.

С помощью D3.js (Data-Driven Documents), библиотеки JavaScript для создания интерактивных и динамических визуализаций данных в веб-браузере, создадим веб-страницу для отображения графа друзей. Она позволяет связывать данные с элементами DOM и применять к ним различные визуальные эффекты с использованием стандартных веб-технологий, таких как HTML, SVG и CSS.

Граф представлен на рисунке 4.9.



***Рисунок 4.9. Граф друзей***

Объяснение про граф друзей друзей

4.4. Реализация приложения для просмотра результатов