МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**КУРСОВАЯ РАБОТА**

**ТЕХНОЛОГИЯ АНАЛИЗА ДАННЫХ СОЦИАЛЬНОЙ СЕТИ**

**В ЭКОСИСТЕМЕ PYTHON**

Работу выполнил А.С. Ролдугин

(подпись, дата) (инициалы, фамилия)

Факультет компьютерных технологий и прикладной математики 3 курс

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Научный руководитель

доц., канд.т.н. Т.А. Приходько

(подпись, дата) (инициалы, фамилия)

Нормоконтролер

ассистент, А.А . Климец

(подпись, дата) (инициалы, фамилия)

Краснодар 2019

СОДЕРЖАНИЕ

Введение……………………………………………………………………..……3

1. Анализ данных социальных сетей – формализация задачи……………..…….5

2. Экосистема Python для анализа данных…………………………..…………..11

2.1 Инструменты анализа данных…………………………………………12

2.2 Визуализация данных …………………………………………...…….14

3. Инструменты взаимодействия с API социальной сети……………….………15

3.1 Проблема ограничения частоты запросов ……………………...…….16

3.2 Извлечение и обработка потоковых данных………………..……...…..16

3.3 Очистка данных социальной сети……………………….……….……17

4. Исследование данных – сущность и временные ряды ..……………..18

4.1 Извлечения хештегов из ленты ..……………………………..……...…18

4.2 Временные ряды……………………………………………………..…20

Заключение………………………………………………………………..….21

Список литературы…………………………………………………….……22

# 

# **ВВЕДЕНИЕ**

Анализ данных социальных сетей (social network data analysis) – одно из современных направлений всестороннего исследования. Интерес исследователей к этому направлению связан с тем, что оно предоставляет новый набор объяснительных моделей и аналитических инструментальных средств, которые находятся вне рамок обычных количественных методов. При этом в данной области накоплен богатый математический аппарат, позволяющий строить весьма сложные модели социальных взаимодействий, описывающие практически любые социальные системы.

Анализ данных социальных сетей используется для исследования и моделирования информационных потоков в сетях, прогнозирования путей развития социальных ситуаций, объяснения специфики исполнения социальных ролей, анализа процессов социального обмена, изучения структуры социальных организаций и взаимодействий между ними, решения задач социометрии, экономической социологии, социологии массовых коммуникаций и Интернета, истории, политики, бизнеса и международных отношений.

Анализ данных социальных сетей имеет прежде всего научное применение и используют его как метод изучения самых разнообразных групп и процессов в обществе. Как и любой метод, берущий свое начало в научной среде, анализ данных социальных сетей наполнился мощным математическим инструментарием, который сегодня, при условии корректной автоматизации и оптимизации, позволяет решать чрезвычайно широкий круг задач.

Целью данной работы была разработка МОДУЛЕЙ (модульной системы) системы сбора и анализа (и визуализации) данных социальной сети. Для решения поставленной задачи была разработана комплексная система, взаимодействия с API социальной сети, извлечение, очистка, обработка и исследование потоковых данных отдельно выбранной социальной сети. Под общим анализом социальных сетей – подразумевается исследование социальных отношений и процессов, скорости и количества обмена информацией между людьми, группами людей, организациями, компьютерами, URL и степенью их вовлеченности.

Курсовая работа состоит из пяти глав.

Первая глава работы содержит общие теоретические сведения об анализе данных социальных сетей и алгоритмах её решения.

Вторая глава данной работы посвящена инструментам анализа и визуализации данных.

В третьей главе рассматривается программная реализация взаимодействия с API социальной сети.

Четвертая глава содержит информацию по исследованию, обработке, очистке данных.

В пятой главе проводится анализ данных через сущности и временные ряды.

1. Анализ данных социальных сетей – формализация задачи

Предметом исследования являются технология анализа данных отдельной выбранной социальной сети. Распространённые социальные сети представляет собой совокупности социальных акторов и набора связей между ними. В качестве социальных акторов могут выступать индивиды, социальные группы, организации, города, страны. Под связями понимаются не только коммуникационные взаимодействия между акторами, но и связи по обмену различными ресурсами и деятельностью, включая эмоциональные отношения. Полученная сеть взаимодействий может быть проанализирована различными методами теории графов, теории информации, математической статистики.

Таким образом социальная сеть – это ни что иное, как объединение социальных позиций (акторов) и их связей, с математическое точки зрения – социальный граф. Важнейшими характеристиками социальной сети как абстрактной категории являются узлы – собственно, участники коммуникационного процесса – или вершины социального графа и связи между ними – ребра социального графа.

Основные методы анализа данных социальных сетей - методы теории графов, в частности, направленные графы и представляющие их матрицы, применяемые для изучения структурных взаимосвязей актора; методы нахождения локальных свойств субъектов, например, центральности, престижа, положения, принадлежности к некоторым подгруппам; методы определения эквивалентности акторов, включая их структурную эквивалентность; блоковые модели и ролевые алгебры; анализ диад и триад; вероятностные модели, включая модели марковских процессов. Далее мы рассмотрим вышеперечисленные методы и приведем примеры содержательных результатов, полученных с их помощью.

1.1 Графовые модели социальной сети

Данный класс моделей может быть разделен на три типа:

**- Стохастические блоковые модели**, задаваемые матрицей X размера k × k, где k - число блоков акторов. Элемент xij ∈ [0;1] показывает плотность связей между акторами, принадлежащими к блоку i, и акторами, принадлежащими к блоку j. При этом граф не содержит дополнительных ребер и вершин, соответствующих связям акторов внутри одного блока.

- **Вероятностные графовые модели**, задаваемые матрицей X размера k × k, где k - число акторов. Элемент xij ∈ [0;1] показывает вероятность взаимодействия актора i и актора j в течение определенного периода времени.

- **Обычные графовые модели**, задаваемые матрицей связности X размера k × k со свойствами, указанными выше.

Для анализа графовых моделей социальных сетей помимо вышеуказанных коэффициентов используется коэффициент плотности Δ - отношение числа ребер в анализируемом графе к числу ребер в полном графе с тем же числом вершин (полный граф — это граф, в котором все вершины соединены между собой) [1].

Графовые модели социальных сетей используются для моделирования экономических и коммуникационных связей индивидов, анализа процессов распространения информации, нахождения различных неформальных объединений и связанных подгрупп, на которые можно разбить общую сеть социальных взаимодействий.

1.2 Анализ центральности и других локальных свойств

Существует несколько подходов для определения центральности акторов. Во-первых, центральные акторы соответствуют тем вершинам сети, которые имеют наибольшее число ребер, т.е. имеют наибольшую степень. Во-вторых, центральная вершина — это вершина, для которой расстояние до остальных вершин сети минимально. В силу этого путь от центральных акторов к другим акторам является наиболее простым, первые имеют большую вероятность получить информацию, циркулирующую в сети, и контролировать распространение этой информации. Третья мера центральности - промежуточность или степень включенности данного актора в маршруты между другими акторами. В этом случае центральным считается актор, который может контролировать наибольшее число путей в социальной сети как между отдельными акторами, так и между их группами [1].

При анализе данных реальных социальных взаимодействий часто наблюдается ситуация доминирования одного актора над другим. Моделью этой ситуации будет ориентированный граф с ребрами двух различных типов, соответствующими отношениям иерархии и передачи информации. Ориентированный граф возникает также при анализе известности акторов в сети. В этом случае для определения центральных вершин необходимо рассматривать входную и выходную степень каждой вершины. Акторы с более высоким статусом будут иметь существенно более высокую входную степень по сравнению с выходной, т.е. их будут знать значительно больше акторов, чем они знают сами.

Для анализа степени центральности различных акторов используется индекс центральности вершины xi в связном графе G, имеющем одну ось симметрии: , где  - сумма расстояний от вершины xi до остальных вершин,  - общая сумма дистанций в графе G [1].

Для сравнения графов по степени их центральности используется понятие индекса центральности графа G: .

Важными характеристиками сети социальных взаимодействий являются сбалансированность и транзитивность. Сбалансированность — это отсутствие ситуаций типа «позитивное взаимодействие (дружба, партнерство) между А и Б, а также между А и В, но негативное взаимодействие (вражда, соперничество) между Б и В». Утверждается, что сбалансированные сети психологически более комфортабельны для акторов и более устойчивы по сравнению с несбалансированными [2].

Транзитивность — это выполнение условий вида «если есть взаимодействие между А и Б, а также между Б и В, то имеет место взаимодействие между А и В». Данные характеристики описывают локальные связи акторов и часто используются при анализе диад и триад.

1.3 Сила структурной позиции актора

Основным показателем, который определяет различия в ресурсах акторов, является сила структурной позиции актора. В теории сетевого обмена для измерения данной характеристики вводится индекс силы актора i: , где  - число непересекающихся путей длины k, проходящих через вершину i. Сила актора *i* по сравнению с *j* есть  [3].

Анализ силы структурных позиций является весьма эффективным методом построения моделей, объясняющих различия в результатах деятельности акторов, например, заработной платы работников или прибыли компаний от продажи некоторого товара, особенно в случае малого числа наблюдений.

1.4 Анализ связанных подгрупп

Связанные подгруппы характеризуются наличием большого числа связей между входящими в них акторами и существенно меньшим числом связей со сторонними акторами. Анализ таких подгрупп позволяет изучать устойчивость социальных структур. Простейший случай связанной группы — это клика или группа акторов, где каждый связан с каждым, и в данную группу не могут быть включены другие акторы, поскольку они не имеют связей со всеми членами клики. Таким образом клика — это максимальный полный подграф данного графа. Если анализировать процессы распространения информации в графах, то можно дать другое определение связанной группы, как множества акторов, где путь между двумя любыми акторами не содержит более одной промежуточной вершины. В результате информация от одного актора к другому в связанной группе передается с минимальными искажениями. Связанные группы также могут быть выделены с помощью многомерного шкалирования или факторного анализа матрицы связей графа.

Для анализа устойчивости групповой структуры во времени используется следующая техника. Вначале строится трехмерная матрица, в которой строки представляют оценки взаимодействий данного актора со всеми другими акторами, данные ими; столбцы являются собственными оценками взаимодействий актора; по оси Z располагаются периоды времени. Далее может быть построен график, показывающий изменения структуры подгрупп с течением времени. После этого применяется корреспондентский анализ для описания зависимостей между строками и столбцами данной матрицы. В результате можно визуализировать изменения статуса индивида на фоне изменений статусов подгрупп [4].

1.5 Структурная эквивалентность акторов

Этот подход является определенной противоположностью исследованию связанных групп. Акторы эквивалентны, когда они занимают одинаковые позиции в социальной структуре, т.е. когда эквивалентны структура и тип взаимодействий данных акторов с другими, при этом эквивалентные акторы не должны взаимодействовать друг с другом.

Наряду со структурной эквивалентностью используется регулярная эквивалентность акторов. В этом случае акторы эквивалентны, когда они одинаковым образом взаимодействуют с акторами одного типа.

Методы определения структурной эквивалентности позволяют анализировать графы с несколькими типами ребер. Например, в случае сети социальных взаимодействий стран мира в качестве этих взаимодействий могут выступать торговля, дипломатические контакты, войны, членство в международных организациях и блоках. Графы, отвечающие каждому типу взаимодействий, могут значительно отличаться, но все вместе они представляют целостную структуру взаимодействий стран мира, из которой можно вывести показатели силы, влияния, уровня развития и экономической мощи данного государства.

Важным показателем является сила связей между акторами, которая определяется как линейная комбинация продолжительности, эмоциональной насыщенности или конфиденциальности и значимости взаимных услуг, которые характеризуют данное взаимодействие и соответствующее ему ребро графа. Слабые связи являются важными источниками информации для актора, так как они служат "мостами" в другие социальные сети, где можно получить сведения о некоторой организации и индивиде [4].

Социальные сети могут рассматриваться как источник ресурсов для акторов данной сети. Эти ресурсы включают в себя любые нормативно значимые ресурсы, которые актор может мобилизовать через сетевые связи для достижения желаемого результата.

1.6 Стохастические модели

Основная идея вероятностных моделей направленных графов состоит в том, что каждая социальная сеть может быть рассмотрена как реализация **x** = {xij} случайного двумерного бинарного массива Х. Так как элементы массива Х являются зависимыми случайными величинами, то можно анализировать структуру зависимостей между соответствующими акторами социальной сети, находить вероятности существования определенных реализаций социальной сети и получать оценки ее параметров.

1.7 Графы зависимости

Используя данные социальной сети, можно построить граф зависимости D, показывающий, какие ребра или группы ребер условно зависимы. Два ребра называются условно зависимыми, если условная вероятность одновременного существования этих ребер, вычисляемая по остальным ребрам сети, не равна произведению условных вероятностей независимого существования этих ребер. Граф зависимости имеет ребра, связывающие все пары условно зависимых ребер социальной сети.

Ненулевые параметры модели сети соответствуют набору максимальных полных подграфов графа зависимости. Причем полным подграфом называется такой набор вершин, где каждая их пара связана ребром, т.е. полный подграф в графе зависимости соответствует набору ребер социальной сети, каждая пара которых является условно зависимой. Максимальный полный подграф — это полный подграф, который не полностью содержится в любом другом полном подграфе.

1. Экосистема Python для анализа данных

Ключевой потенциал систем анализа данных кроется в извлечении полезных выводов из данных. Цель данного процесса состоит в том, чтобы ответить на интересующие вопросы, используя технологии анализа данных, и достичь более полного понимания конкретной предметной области. Например, розничный онлайн-магазин может применить анализ данных, чтобы понять, как их потребители выбирают товары. По результатам анализа они смогут рекомендовать клиентам товары в зависимости от их покупательских привычек. В общем случае это улучшает качество обслуживания клиентов и их удовлетворенность, что в итоге увеличивает рентабельность любого предприятия. Многие организации в разных сферах деятельности могут применять технологии анализа данных для совершенствования своего бизнеса. Вот только некоторые примеры:

1) банковское дело:

— выявление лояльных клиентов с целью предложения им эксклюзивных

программ;

— распознавание мошеннических схем с целью уменьшения затрат;

2) медицина:

— изучение поведения пациентов с целью предсказания посещений хирурга;

— помощь врачам в идентификации успешных способов лечения в зависимости от анамнеза пациента;

3) розничная торговля:

— изучение схем поведения во время покупок с целью улучшения качества

обслуживания клиентов;

— увеличение эффективности рекламных кампаний за счет более точной

направленности;

— анализ оперативных данных транспортных потоков для поиска самого

быстрого маршрута доставки продуктов питания.

По сути дела, вопрос cостоит в том, как пользователи делятся своими данными посредством социальных сетей. Организации больше не ограничиваются анализом данных, собираемых непосредственно, поскольку теперь они обладают доступом к гораздо большему объему данных.

2.1 Инструменты анализа данных

В данной главе рассмотрим стек инструментов программной реализации, используемых для решения задачи разработки модульной системы сбора, анализа и визуализации данных социальной сети.

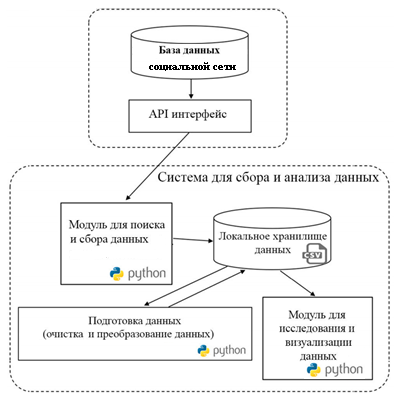


Рис.1 Архитектура системы анализа данных социальной сети

* + 1. Язык программирования Python

Язык программирования Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Данный язык поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и так далее.

Одной из причин, почему Python прекрасно подходит для анализа данных, математических задач и прочих научных расчетов, является то, что Python – это удобное и выразительное средство связи между читабельным кодом и быстрыми библиотеками, написанными на С\C++.

Другая причина заключается в том, что на данный момент существует много реализаций различных методов анализа данных, соответствующих библиотек и прочих вспомогательных инструментов таких как, например, пакет для научных вычислений NumPy, пакет для работы с данными Pandas и так далее.

* + 1. Библиотека Pandas

Pandas – программная библиотека на языке Python для обработки и анализа данных. Работа Pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Pandas предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Название библиотеки происходит от эконометрического термина «панельные данные», используемого для описания многомерных структурированных наборов информации. Pandas распространяется под новой лицензией BSD.

Возможности библиотеки, используемые в данной работе:

* объект DataFrame для манипулирования индексированными массивами двумерных данных;
* инструменты для обмена данными между структурами в памяти и файлами различных форматов;
* встроенные средства совмещения данных и способы обработки отсутствующей информации;
* слияние и объединение наборов данных.

2.1.2 Библиотека NumPy

NumPy - библиотека для Python, предоставляющая поддержку многомерных массивов и высокоуровневых математических функций, предназначенных для работы с многомерными массивами. Предлагает инструменты быстрой и эффективной обработки массивоподобных структур данных. Хранение и манипуляция числовыми данными с использованием встроенных в Python структур данных (к примеру, списков или словарей) оказывается намного медленнее, чем использование массива из NumPy. Кроме того, массивы NumPy часто используются другими библиотеками в качестве контейнеров данных для различных алгоритмов, выполняющих операции с векторами и матрицами.

2.2 Визуализация данных

Визуализация данных — это область информатики, связанная с визуальным представлением данных. Визуальное представление – мощный инструмент, позволяющий исследовать сложные данные, и эффективный способ представления и передачи результатов анализа в целом. Посредством визуализации люди получают возможность увидеть аспекты, которые не заметны с первого взгляда.

В конце концов, изображение стоит тысячи слов - удачная визуализация позволяет читателю впитывать сложные понятия при помощи простого изображения.

Например, визуализация данных может использоваться исследователями-аналитиками во время первичного анализа данных, чтобы понять их суть. Кроме того, исследователи-аналитики могут также использовать визуализацию для общения с неспециалистами и объяснения, что есть интересного в данных.

*Python* предлагает большое количество инструментов для визуализации данных, например библиотеку *Matplotlib*, которая позволяет выводить изображения типографского качества в многочисленных форматах.

1. Инструменты взаимодействия с API социальной сети

Рассмотрим инструменты взаимодействия с *Twitter* - одной из популярных социальных сетей. *Twitter* предоставляет услугу микроблогинга, своего рода электронный журнал, в котором информация следует короткими сообщениями – твитами, размер которых ограничен 140 символами. В отличие от других социально сетевых платформ, таких как *VK* или *Facebook, Twitter* - сеть не двунаправленная, то есть связи не обязательно могут быть взаимными.

Пользователи *Twitter* обсуждают происходящие события в режиме реального времени, включая торжества, телевизионные сериалы, спортивные состязания, политические и бизнес новости, погодные явления и многое-многое другое.

С учетом разнообразия применений, *Twitter* представляет собой потенциальный огромный интерес для аналитиков данных. Социальная сеть *Twitter* предлагает серию прикладных программных интерфейсов (*АРI*) для доступа к своим данным:

* чтение твитов;
* извлечение информации из профилей пользователей;
* отправку сообщений от имени пользователя.

Чтобы настроить проект для доступа к данным в *Twitter*, нужно выполнить два предварительных этапа:

* зарегистрировать приложение;
* выбрать клиента *Twitter API*.

После регистрации получаем для входа в систему *Keys* и *Access Token* (Ключи и Маркеры доступа), а также необходимые для дальнейшего доступа к данным разным данным социальной сети *Twitter* – Ключ потребителя (*Consumer Key*) и секретный ключ потребителя *Consumer Secret*, которые называют *API Key* и *API* *Secret*. В свою очередь уровень доступа *Access Level* определяет, какие действия может выполнять приложением в *Twitter* от имени пользователя.

3.1 Проблема ограничения частоты запросов

*API* социальной сети *Twitter* ограничивает доступ для приложений. Маркеры доступа устанавливают ограничения для каждого пользователя. Таким образом, когда приложение использует аутентификацию только для приложения, ограничение частоты запросов действует глобально для всего приложения, тогда как при использовании аутентификации от лица пользователя приложение может повысить общее число запросов к *АРI*. По достижении установленных пределов вместо запрошенных данных *Twitter* вернет сообщение об ошибке. Более того, если продолжить и дальше посылать запросы к *АРI,* время ожидания, требующееся для возобновления нормального доступа, увеличится, потому что *Twitter* может рассматривать данную ситуацию как потенциальную угрозу злоумышленников. Если создаваемому приложению требуется послать много запросов, необходимо предусмотреть способ избежать такой ситуации. В стандартной библиотеке *Python* имеется модуль time позволяющий приостанавливать выполнение программного кода на произвольные интервалы вызовом функции *time.sleep().*

3.2 Извлечение и обработка потоковых данных

Социальная сеть *Twitter* поддерживает несколько *API*, которые условно можно разделить на два класса:

- *REST API* - поисковый:

- *STREAMING API* – потоковый.

Поисковые *REST API* позволяют обращаться только к прошлому.

Взаимодействуя с *Twitter* через *REST API*, можно искать фактически существующие твиты, опубликованные и доступные для поиска. Эта группа *API* ограничивают количество твитов, которые можно получить, ограничивая не только частоту запросов, но интервалы времени между запросами. Как правило для доступа открыты данные не более чем недельной давности, то есть более ранние твиты извлечь нельзя. Второй важный недостаток *REST API*, состоит в том, что не гарантируются возврат абсолютно всех сообщений, которые были опубликованы в *Twitter*, так как некоторые могут быть недоступны для поиска или еще не были проиндексированы.

Потоковый *STREAMING API*, после установки соединения и условии сохранения *HTTP* соединения открытым позволяет двигаться вперед по линии времени получая все твиты, соответствующие заданным критериям фильтрации, по мере их публикации.

В общем случае потоковый интерфейс *STREAMING API* предпочтительнее, когда требуется загрузить большое количество твитов, так как взаимодействие с платформой ограничивается лишь поддержкой соединения открытым. Недостаток такого подхода заключается в его продолжительности, потому что приходится ждать публикации твитов, прежде чем получится их собрать и проанализировать.

В свою очередь *REST API* удобно использовать, когда требуется отыскать твиты, созданные определенным пользователем, или получить доступ по определенной ленте, а *STREAMING API* может пригодиться, когда требуется выполнить фильтрацию по конкретному ключевому слову и скачать связанный с ним массивный объем твитов (например, реакция на конкретное событие в прямом эфире).

3.3 Очистка данных социальной сети

Для сбора данных социальной сети *Twitter* и последующей их обработки необходимо приложение – клиент, реализующее различные формы взаимодействия и обращения к *Twitter API*. Для этих целей воспользуемся библиотекой *Tweepy*, обладающей достаточно широким функционалом и возможностями.

На первом этапе после регистрации и настройки аутентификации приложения имеем: ключ потребителя, секретный ключ потребителя, маркер доступа и секретный маркер доступа.

Сохраним учетные данные в переменных окружения, чтобы отделить прикладную логику от конфигурации. Использование окружения для хранения конфигурации обеспечит независимость от языка программирования и операционной системы.

4 Исследование данных – сущность и временные ряды

4.1 Извлечения хештегов из ленты

Проведем частотный анализ на основе данных, собранных ранее (когда ранее?). Данный анализ позволит получить ряд статистик. Исследование и анализ сущности хештег, интересен прежде всего тем какими темами автор определяет твит.

Проведем анализ твитов <https://twitter.com/habr_com> - портал посвященный программному обеспечению. Узнаем какие технологии упоминаются на портале наиболее часто. Для этого воспользуемся сценарием Приложение 3 для извлечения хештегов из ленты <https://twitter.com/habr_com> и вывода наиболее распространённых тегов.

Запускается этот сценарий следующей командой:

**$ python twitter\_hashtag\_frequency.py user\_timeline\_Habr\_com. jsonl** Здесь *user\_timeline\_Habr\_com — это файл JSON Lines*, созданный ранее.

Данный сценарий принимает аргумент командной строки с именем файла

*. jsonl* и построчно читает его содержимое. Поскольку каждая строка содержит документ *JSONL*, сценарий загружает строку в переменную *tweet* и вызывает вспомогательную функцию *get\_hashtags (),* чтобы извлечь список хештегов. Этот тип сущностей сохраняется в переменной *hashtags,* которая объявляется как *collections.*

*Counter* – специальный тип словаря, который используется для подсчета хешируемых объектов - в нашем случае строк. Словарь *hashtags* хранит строковые ключи и соответствующие им частоты - в качестве значений.

Будучи подклассом типа *dict*, объект Counter является неупорядоченной коллекцией. Его метод *most\_common ()* отвечает за упорядочение ключей в зависимости от их значений (в порядке убывания частоты) и возвращает список кортежей (ключ, значение). Вспомогательная функция *get\_hashtags ()* извлекает список хештегов из твита. В качестве аргумента ей передается весь твит, загруженный

в словарь. Если сущности в твите присутствуют, словарь будет иметь ключ *entities.* Но, так как он может отсутствовать, нельзя напрямую обратиться к твиту, как *tweet [‘entities’],* потому что эта инструкция возбудит исключение KeyError в отсутствие данного ключа, поэтому воспользуемся функцией *get (),* передав ей пустой словарь как значение по умолчанию, возвращаемое в отсутствие сущностей. Второй шаг- извлечение хештегов из сущностей. Поскольку *entities* — это тоже словарь, а ключ *hashtags* тоже может отсутствовать, снова воспользуемся функцией *get (),* но на этот раз в качестве значения по умолчанию укажем пустой список. И, наконец, воспользуемся генератором списков, чтобы выполнить обход хештегов и извлечь из них текст. Хештеги нормализуются функцией *lower (),* которая переводит текст в нижний регистр, поэтому такие упоминания, как *#Java* или *#JAVA*, все будут сгруппированы в *#java.*

Анализ твитов портала <https://twitter.com/habr_com> с помощью разработанного сценария дал следующий результат:

webdev: 147

python: 122

javascript: 121

machineleaning: 110

gamedev: 83

appdev: 79

tech: 61

data: 58

map: 52

java: 52

datascience: 45

devops: 45

node.js: 43

deeplearning: 38

docker: 37

Из анализа сущностей автора видны наиболее часто используемые конкретные технологии в области IT технологий.

4.2 Временные ряды

Распределение твитов во времени один из важнейших и интересных аспектов исследования.

Временной ряд — это последовательность точек данных, которая состоит из наблюдений за тот или иной интервал времени. Поскольку твиты имеют поле **created\_at** с точным временем создания, мы можем перераспределить твиты по временным дискретным интервалам, чтобы выяснить, как пользователи в отдельности и целые группы реагируют на те или иные события во времени. Для такого анализа лучше всего подходят данные, собираемые через потоковый программный интерфейс Streaming API.

Проведем анализ на наборе данных, полученных во время финала чемпионата мира по футболу 2018 используя скрипт Приложение 5.

Воспользуемся возможностями библиотеки pandas по управлению временными

рядами и снова применим библиотеку для визуализации результатов:

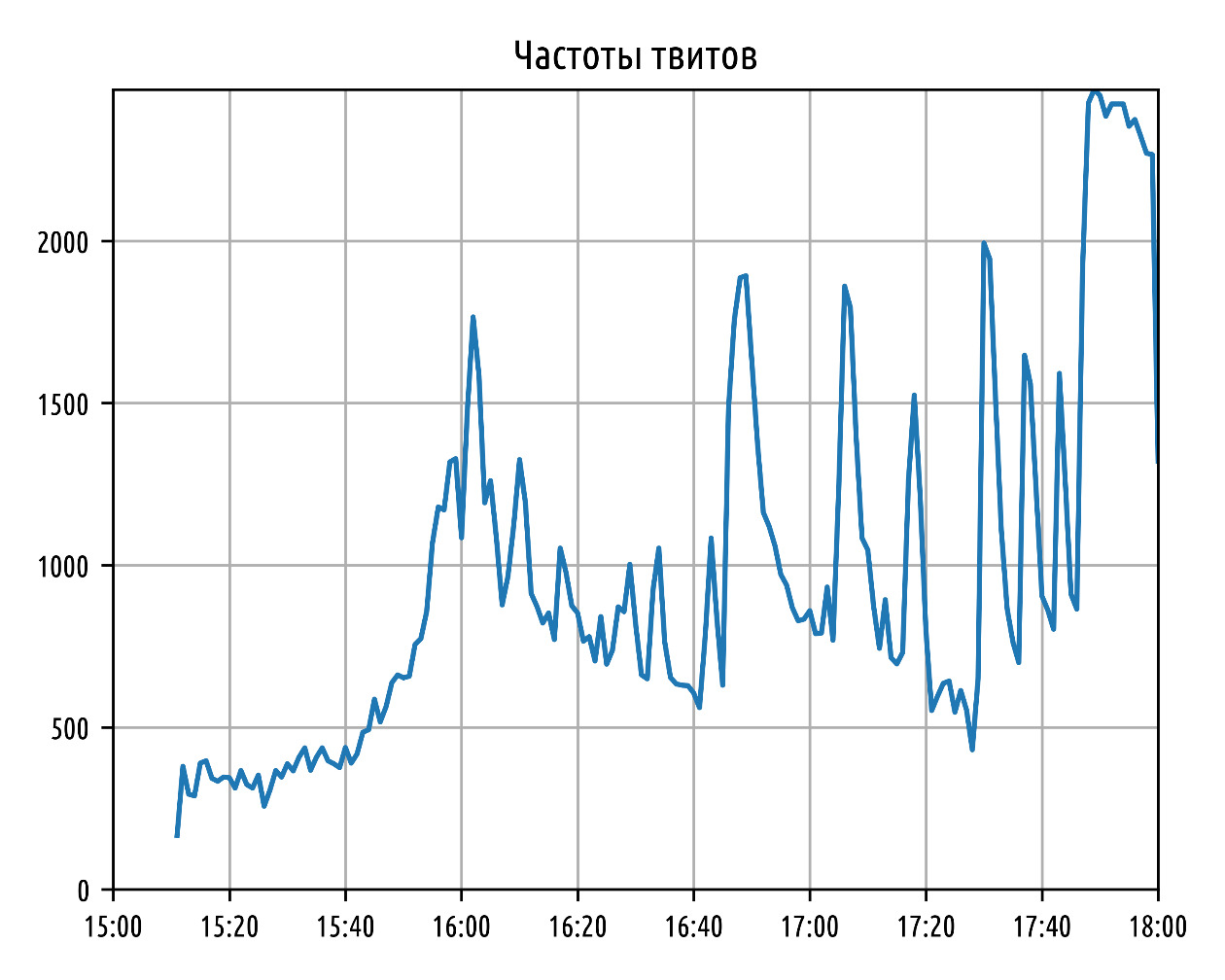


Рис.2 Частота твитов во время финала чемпионата мира 2018

Этот пример наглядно демонстрирует, как пользователи реагируют на такие грандиозные событиям. Таким образом данный анализ временных рядов крайне необходим многим компаниям. Посредством такого мониторинга мнений пользователей управленческие решения будут более оптимизированными и взвешенными во всех отношениях. А динамическая природа социальных сетей идеально подходит для выяснения реакции на любые события, новости, явления.

ЗАКЛЮЧЕНИЕ

В данной курсовой работе были реализованы модули для взаимодействия с программным интерфейсом Twitter API и Streaming API. Также были детально изучены способы сбора и загрузки твитов. Важнейшей отправной точкой исследования стало рассмотрение частотного анализа на основе сущностей в твите. В завершении был подробно разобран алгоритм анализа временных рядов, применение которого аналитиками данных может оказаться мощным инструментом анализа больших наборов данных.

Анализ данных социальных сетей – инструмент, мощность которого трудно переоценить. На примере нескольких метрик в проекте было продемонстрировано, насколько такой анализ может быть полезен для принятия важных решений и на сколько способен предоставить достаточно полную и подробную картину взаимодействий и реакций сети на те или иные события.

СПИСОК ЛИТЕРАТУРЫ

1. Вассерман С., Фауст K. Анализ социальных сетей. Методы и приложения. New York: Cambridge University Press, 1994.
2. Сорокин П.А. Система социологии. Т. 2. Социальная аналитика: Учение о строении сложных социальных агрегатов. М.: Наука, 1993.
3. Титов С.А. Исследование организационной культуры с помощью моделей социальных сетей// Cloud of Science. – 2015. – Т. 2. – №2. – С. 236–246.
4. Антон Коршунов. Задачи и методы определения атрибутов пользователей социальных сетей // Труды 15-й Всероссийской научной конференции
5. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL’2013 Антон Коршунов, Иван Белоб
6. Сергей Бартунов, Антон Коршунов. Идентификация пользователей социальных сетей в Интернет на основе социальных связей // Доклады Всероссийской научной конференции «Анализ изображений, сетей и текстов» – АИСТ'2012. Екатеринбург, 16-18 марта 2012 г
7. Документация на русском языке. [Электронный ресурс] Документация, URL: http://wiki.python.su /BeautifulSoup [Дата обращения: 18 октября 2019]
8. Давлатов Ш. Сравнительный анализ инструментов для сбора и анализа данных из открытых источников OSINT // Матер. 53-й науч. конф. аспирантов, магистрантов и студентов БГУИР. Минск, 2–6 мая 2017 г.
9. Вандер П. Python для сложных задач. Наука о данных и машинное обучение. O'ReillyMedia, 2018. 576 с.