Санкт–Петербургский государственный университет

Скурихина Ирина Сергеевна

Выпускная квалификационная работа

Методы для визуализации пользовательского веб-графа дискуссии в социальных сетях

Уровень образования: бакалавриат

Направление 02.03.02 «Фундаментальная информатика и информационные технологии»

Основная образовательная программа СВ.5003.2018 «Программирование и информационные технологии»

Научный руководитель:
доцент кафедры технологии программирования, к.т.н.
Блеканов И. С.

Санкт-Петербург
2022 г.

СОДЕРЖАНИЕ

Введение

В наши дни веб активно эволюционирует. Люди со всех уголков земли могут принимать активное участие в генерации контента. Им не нужно знать технологии, языки программирования и т.п., чтобы создать свой веб-ресурс. Каждый человек может с легкостью завести сообщество или страницу в социальной сети, так как для всего этого есть интуитивно понятные и доступные инструменты. В связи с этим пользователей социальных сетей становится все больше и больше (на данный момент в Twitter зарегистрировано 1.3 миллиарда аккаунтов, 330 миллионов из них являются активными пользователями), а следовательно, и количество информации, которое они генерируют, непрерывно растет.

Исследования, которые ведутся в настоящий момент [7], [8], [12], направлены на изучение поведения пользователей в дискуссиях в контексте реальных событий. Так как любое событие, происходящее в нашем мире, всегда сопровождается его обсуждением в социальных сетях.

Наука, занимающаяся исследованием социальных сетей в терминах теории графов — это и есть анализ социальных сетей (ACC или “social network analysis / SNA” в англоязычной литературе). Две основные задачи ACC — это сентиментный и структурный анализ сетей. Сентиментный анализ занимается изучением эмоций пользователей в сети, в его основе лежат нейролингвистическое программирование и машинное обучение. Структурный же анализ занимается исследованием сетей, посредством представления их в виде графов, в его основе лежит поиск эффективного метода визуализации графа для моделирования его структуры.

Данная работа посвящена структурному анализу пользовательских дискуссий в сети по реальным событиям.

Актуальность

В связи с активно растущим количеством информации в сети из-за разного контингента, числа пользователей и т.п. перед нами стоим интересная и при этом сложная задача — как лучше усвоить данную информацию.

Проблематика больших данных крайне актуальна из-за их разнообразия, высокой скорости поступления и конечно же большого объёма. Исследования, занимающиеся данной проблемой, решают такие задачи, как нахождение важных пользователей в сети [7], выявление их влияния на других пользователей [8], а также нахождение скрытых сообществ [12]. Поэтому семантический и структурный анализ дискуссий в сети является очень важным.

Проблема визуализации больших данных связана в первую очередь с нахождением и развитием методов, которые помогут компактно и эргономично их представить.

Глобальной же проблемой является именно восприятие большой информации, так как количество узлов в графе может достигать нескольких миллионов. Выбор эффективного метода раскладки графа позволяет отследить образовавшиеся сообщества, оценить важность узла по его расположению и размеру (например концентрация популярных пользователей в центре раскладки или же наоборот на ее периферии; зависимость размера узла от его степени), а так же выявить структуру и основные свойства графа. Всё это необходимо для качественного усвоения большой информации различными специалистами.

Цель работы

Целью данной работы является разработка методов и инструментов для визуализации структуры дискуссий разного объёма в рамках реальных событий в социальных сетях, позволяющих эффективным образом представлять пользователей и их связи для качественного усвоения даже самой большой информации экспертами из смежных предметных областей: социологами, политологами и т.д.

Задачи работы

Для реализации поставленной цели в работе были определены следующие задачи:

- Провести обзор научной литературы по тематике анализа социальных сетей
- Провести обзор методов визуализации графов
- Провести обзор существующих решений для раскладки графов
- Провести тестирование и апробацию существующих алгоритмов визуализации графов на четырех реальных дискуссиях разного объёма, для выявления эффективных методов раскладки графов
- Разработать программный комплекс, состоящий из конкретных методов и инструментов, для работы с дискуссией в виде пользовательского веб-графа, а также адаптации этих инструментов на призму понимания и анализа структурных особенностей пользовательских дискуссий в социальных сетях

Практическая значимость работы

Данная дипломная работа несет следующие аспекты практической значимости:

- Результаты проведенного тестирования могут использоваться в качестве базы исследовательской, аналитической и проектной деятельности авторов, изучающих тему структурного анализа социальных сетей
- Разработанный программный компонент может быть внедрен в промышленную эксплуатацию экспертами: политологами, историками, социологами и другими, для качественного усвоения и анализа информации в сети разного объёма

Глава 1. Обзор алгоритмов визуализации графов и существующих решений

1.1 Представление структуры пользовательской дискуссии

Граф G – это пара (V, E), где V = {v1, v2, … vn} – множество вершин, а E = {(u,v)| u∈V, v∈V} – множество ребер графа [3]. Стоит отметить, что к компьютерным дисциплинам применительно вместо графа чаще использовать термин «сеть».

Введем понятия «ориентированный» и «неориентированный» граф. Неориентированным является тот граф, у которого ребра не имеют направления и (u,v) = (v, u). В случае, когда выполняется свойство (u, v) != (v, u), граф является ориентированным или орграфом [3].

Та или иная социальная сеть рассматривается, как социальное явление и напрямую связана с понятием социального графа.

Поэтому дискуссия по любому событию — это пользовательский ориентированный веб-граф [12]. В котором участники дискуссии являются множеством вершин графа, а связи между ними — ребрами.

В данной работе в качестве модели пользовательской дискуссии мы представляем направленный пользовательский веб-граф G: $G_D = G(V, E)$, где V означает всех пользователей, принявших участие в дискуссии D [12].

1.2 Обзор Force-directed алгоритмов

Повсеместно для визуализации веб-графов применяется метод «направленных сил», или «Force-directed method/approach» в англоязычной литературе. Данный метод работает по следующему принципу: в первую очередь задается случайное начальное состояние физической системы, которая состоит из пружин (ребер) и металлических колец (вершин). Система приходит в движение из-за деформации пружин посредством прикладывания силы к вершинам [1]. Происходят колебания, которые приводят к сжатию и растягиванию пружин (Рис. 1.1), а также к изменению положения колец-вершин. Работа алгоритма завершается, когда система достигает своего минимального энергетического состояния – то есть, когда положение вершин больше не изменяется относительно друг друга от одной итерации к другой (Рис. 1.2) [9].

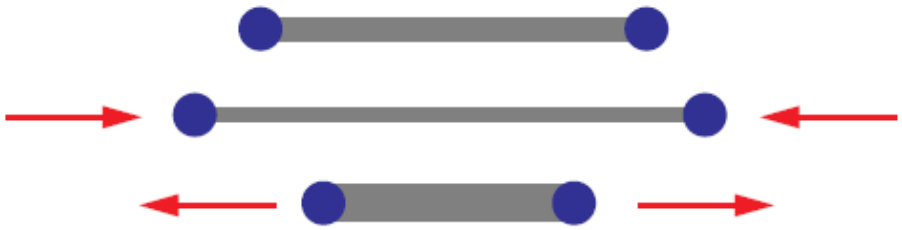

Рисунок 1.1 - Сжатие и растягивание пружин

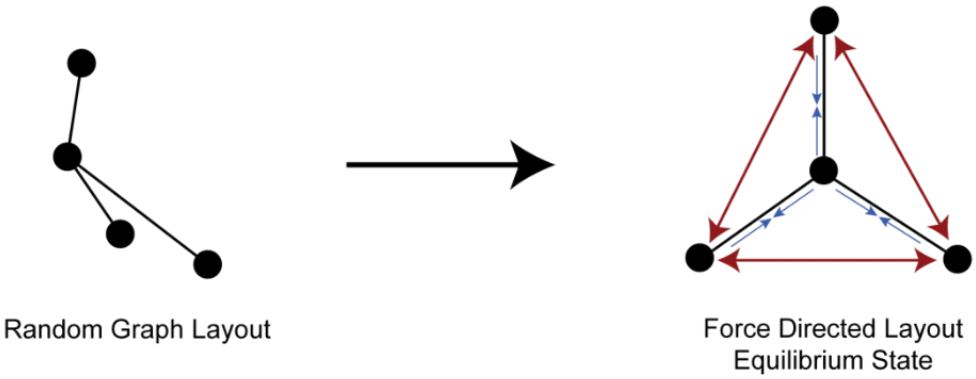

Рисунок 1.2 - Демонстрация работы метода «направленных сил»

«OpenOrd»

OpenOrd — это алгоритм, который предназначен для визуализации очень больших сетей, так как он работает на достаточно высокой скорости и при этом имеет среднюю степень точности. Стоит отметить, что хоть это и отличный вариант для больших сетей, он нежелателен для применения на небольших графах. Так как для них потеря точности может быть достаточно велика по сравнению с другими методами визуализации графов.

Описание работы: все вершины на старте помещаются в начало координат, затем проводятся итерации по оптимизации. Итерации подвергаются контролю с помощью алгоритма имитации отжига, который включает в себя пять фаз [4]:

- Жидкость (Liquid)
- Расширение (Expansion)
- Охлаждение (Cooldown)
- Сжатие (Crunch)
- Кипячение (Simmer)

На каждом этапе отжига изменяется несколько параметров оптимизации: температура, притяжение и демпфирование (гашение колебаний). Эти параметры определяют, насколько далеко вершины могут отдаляться друг от друга. На каждом шаге алгоритма вычисляется два возможных перемещения вершин. Первое возможное перемещение — это всегда случайный прыжок, расстояние до которого определяется температурой. Второе возможное перемещение рассчитывается аналитически (известное как прыжок через барьер). Это перемещение рассчитывается как взвешенный центр тяжести соседних вершин. Множитель демпфирования определяет, насколько близко к данному центру вершине разрешено приблизиться, а коэффициент притяжения определяет необходимость такого движения. Из этих двух возможных перемещений выбирается то, которое приводит к наименьшей внутренней суммарной энергии [4].

Конечный вид графа зависит от того, сколько по времени длился каждый этап (Рис. 1.3). По умолчанию уходит около 25% времени на стадию жидкости, 25% на стадию расширения, 25% на стадию охлаждения, 10% на стадию сжатия и 15% на стадию кипения (Рис. 1.4). Настройки по умолчанию основаны на обширном исследовании, которое провели сами авторы алгоритма.

Рисунок 1.3 - Результат применения метода OpenOrd. Liquid 25%, Expansion 25%, Cooldown 25%, Crunch 10%, Simmer 15%

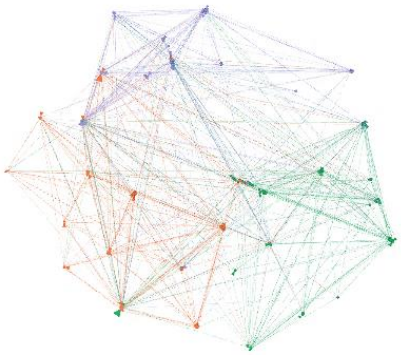

Рисунок 1.4 - Результат применения метода OpenOrd. Liquid 25%, Expansion 75%, Cooldown 25%, Crunch 10%, Simmer 15%

Существует настройка *Edge Cut (от 0 до 1)*, которая дает пользователям возможность управлять обработкой краев. Значение 0 скорее всего приведёт к тому, что кластеры будут примыкать или же совсем перекроют друг друга (Рис. 1.5).

Рисунок 1.5 - Результат применения метода OpenOrd. Edge Cut 0

Если же установить более высокое значение, то можно увеличить пространство между кластерами и сделать график визуально привлекательнее (Рис. 1.6).

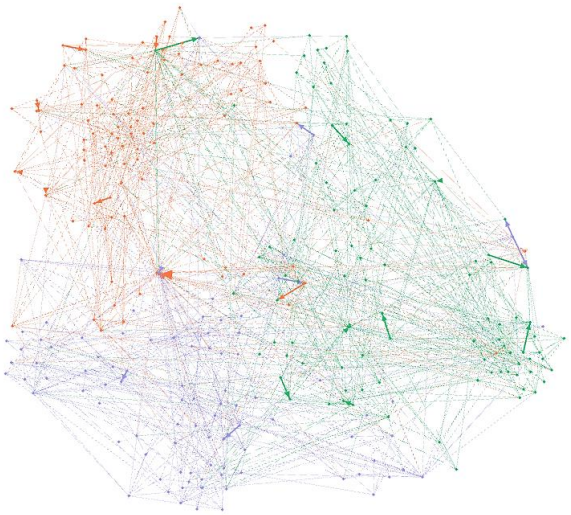

Рисунок 1.6 - Результат применения метода OpenOrd. Edge Cut 1

Параметры *Num Threads* и *Num Iterations* отвечают за количество используемой вычислительной мощности компьютера и увеличение числа итераций за счет дополнительной оптимизации. Это очень полезные параметры для работы с большой сетью.

«Frushterman-Reingold»

Алгоритм «Frushterman-Reingold» был разработан в 1991 году Томасом Фруштерманом и Эдвардом Рейнгольдом. Сложность алгоритма составляет $O(N^2)$. Для расчета положения узлов алгоритм FR использует две силы (притяжение и отталкивание). С помощью данного алгоритма можно эффективно построить двумерные представления графов, в которых содержится не более 1000 вершин. Так же стоит уточнить, что для построения не используется вес ребер.

Параметры [11]:

- *Area* - среднее между двумя силами, максимально равномерно распределяет узлы.
- *Gravity* - сила притяжения.
- *Speed* - компромисс между скоростью и точностью. Более высокие значения приводят к более быстрым, но менее точным результатам.

Данный метод укладки требует достаточно длительного времени выполнения. Вместо двух параметров: отталкивания и притяжения Фрухтерман-Рейнгольд использует один: *область (area) [11]. Он* действует как среднее между двумя силами и максимально равномерно распределяет узлы, что дает примыкание кластеров друг к другу. Так как параметр всего один, то алгоритм имеет меньше возможностей для настройки конечным пользователем (Рис. 1.7).

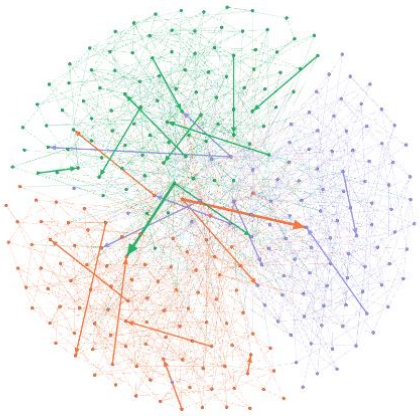

Рисунок 1.7 - Результат применения метода Fruchterman Reingold. Область 10 000, гравитация 10

Функция *Гравитации (Gravity):* если задать достаточно высокие значения, то сеть будет вытянута к центру. Если *Гравитация* равна 0, то узлы будут стремиться к центрам отдельных кластеров (Рис. 1.8) [11].

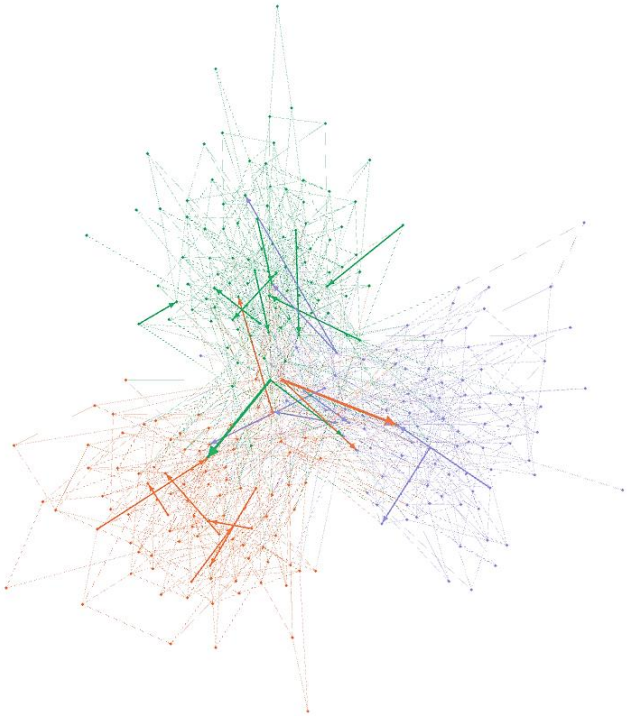

Рисунок 1.8 - Результат применения метода Fruchterman Reingold. Область 1 000, гравитация 0

Скорость (Speed): данный параметр можно использовать для ускорения завершения алгоритма за счет более низкого уровня точности [11].

«Yifan Hu Multilevel»

Алгоритм «Yifan Hu Multilevel» был создан Йфаном Ху в 2005 году. Его сложность составляет O(N*log(N)). Существуют ограничения на размер графа: 100 – 100 000 вершин. В данном алгоритме вес ребер не задействован, как и в предыдущем. «Yifan Hu Multilevel» работает быстрее предыдущего алгоритма за счёт расчета силы притяжения и отталкивания узлов, расположенных по соседству (а не для для всей сети), однако из-за этого он теряет в точности.

Параметры [1]:

- Step ratio: чем выше коэффициент шага, тем лучше качество (за счет снижения скорости).
- Optimal distance: оптимальное расстояние между узлами (Рис. 1.9, 1.10).
- Theta: чем меньше тета, тем точнее результаты (соответственно алгоритм замедляется).

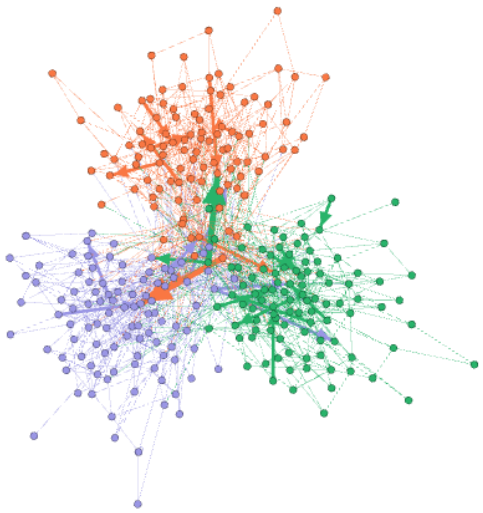

Рисунок 1.9 - Результат применения метода Yufan Hu. Оптимальное расстояние 100, относительная сила 0.2

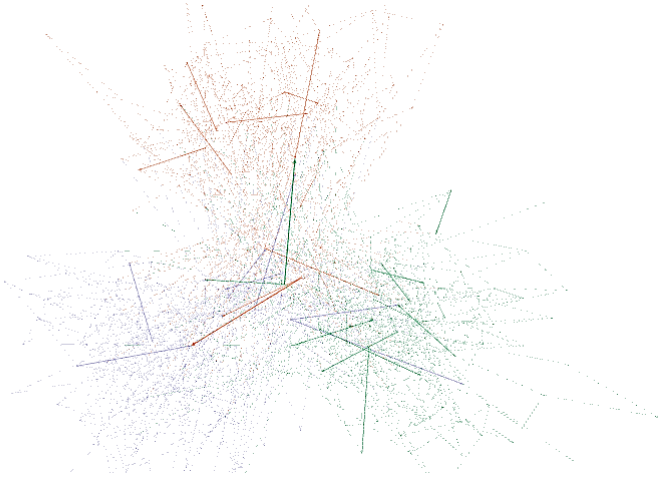

Рисунок 1.10 - Результат применения метода Yufan Hu. Оптимальное расстояние 1000, относительная сила 0.2

Так же для настройки отталкивания и притяжения узлов имеется еще параметр: *Относительная сила* [1]. Увеличение данного параметра усиливает отталкивание и отдаление узлов друг от друга (Рис. 1.11). Уменьшение же стягивает узлы, так как силе притяжения назначается больший вес, чем силе отталкивания.

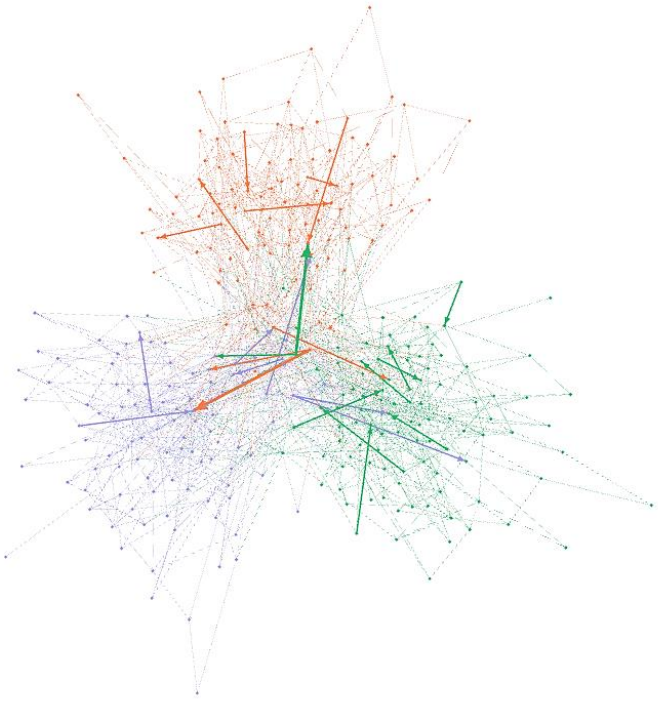

Рисунок 1.11 - Результат применения метода Yufan Hu. Оптимальное расстояние 100, относительная сила 5

«Force Atlas 2»

В 2007 году создатели Gephi разработали Алгоритм «Force Atlas 2» для визуализации безмасштабных сетей (графы, в которых степени вершин распределены по степенному закону). Нововведениями в данном алгоритме являются модели сил притяжения, гравитации и отталкивания. Благодаря им визуализация получается более строгой, а также пересечение ребер сводится к минимуму. Сложность алгоритма равна $O(N^2)$, что позволяет визуализировать графы от 1 до 10 000 вершин. Так же при построении учитывается вес ребер [6]. На выходе данного алгоритма можно увидеть максимально наглядную раскладку графа, так как при проектировании метода был сделан акцент на качестве визуализации.

Параметры [5]:

- *Scaling*: отвечает за масштаб расширения графика.
- *Dissuade hubs*: обеспечивает более высокие силы отталкивания между узлами.
- *Prevent overlap*: предотвращает перекрытие узлов (очень сильно замедляет алгоритм).
- *Gravity:* обеспечивает более высокую силу притяжения между узлами.
- *LinLogMode:* плотнее прижимает узлы друг к другу внутри каждого кластера.
- *StrongGravityMode:* стягивает узлы к центру.

Начнем с алгоритма *Force Atlas* — это классический силовой подход, в основе которого лежат принципы отталкивания, притяжения и гравитации, который подходит как для малых наборов данных, так и для достаточно больших. *Force Atlas* — один из самых медленных методов визуализации среди представленных выше, так как имеет самую высокую степень точности.

Но всегда можно остановить построение графа до завершения алгоритма, если необходимая точность уже достигнута.

Параметр *Сила отталкивания* отвечает за величину отталкивания между узлами. Если данный параметр имеет большое значения, то узлы будут располагаться достаточно далеко друг от друга. Также его можно использовать вместе с противоположным параметром *Сила притяжения.* Большое значение которого стянет узлы ближе друг к другу, что создаст более кластеризованную сеть. Для лучшей визуализации также можно задать размеры узлов в зависимости от их степеней (Рис. 1.12).

Рисунок 1.12 - Результат применения Force Atlas. Сила отталкивания 20 000

Еще одна полезная функция — это флажок *«Учитывать размер».* Этот переключатель контролирует перекрытие маленьких узлов большими (Рис. 1.13).

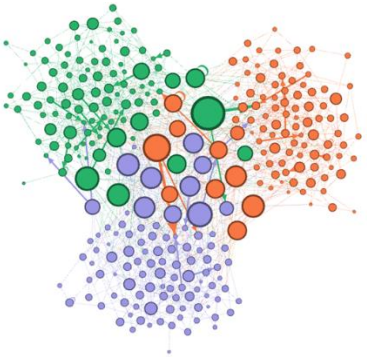

Рисунок 1.13 - Результат применения настройки «Учитывать размер»

Если необходимо стянуть узлы к центру графа, то можно выставить большое значение параметру *Гравитация* (Рис. 1.14). А если наоборот волнует чрезмерная скученность, то меньшие значения позволят ее избежать.

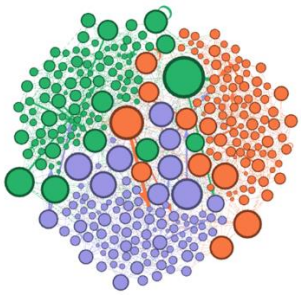

Рисунок 1.14 - Результат применения настройки «Гравитация». Значение гравитации 30000.

Включённый параметр *«Ослабление хабов»* дает возможность вынести более крупные узлы на край макета, чтобы увидеть граф в иной перспективе (Рис. 1.15).

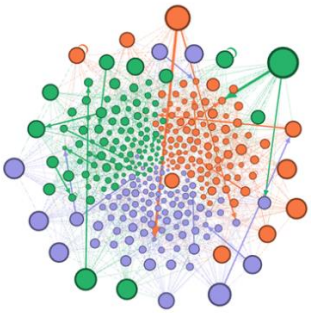

Рисунок 1.15 - Результат применения настройки «Ослабление хабов»

Параметр *скорости* отвечает за снижение качества построения при его увеличении и наоборот.

Force Atlas 2 — это более быстрая, обновленная версия Force Atlas. Отличие от *FA*: наличие параметра *число потоков*, дает большую мощность для построения макета. Значение по умолчанию задается равным 2, но оно может быть увеличено в зависимости от возможностей локальной машины.

Параметр *Разреженность* задает расстояние между узлами (Рис. 1.16).

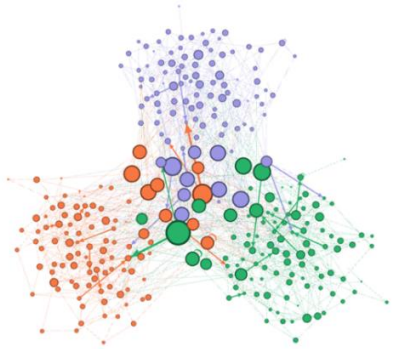

Рисунок 1.16 - Результат применения Force Atlas 2. Разреженность 200

Аналогом параметра *«Учитывать размер»* будет параметр *«Запрет перекрытия»*, который можно использовать с теми же целями (аналогично очень сильно замедляет работу алгоритма) .

«Влияние весов рёбер» равное 1.0 соответствует нормальному влиянию, а значение 0 означает, что веса ребер не учитываются при построении. Уровни выше единицы стягивают сильно связанные узлы ближе друг к другу (Рис. 1.17).

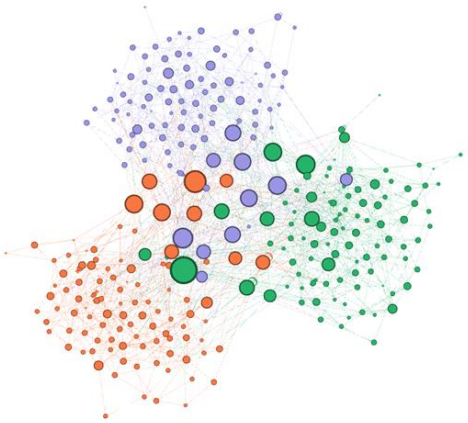

Рисунок 1.17 - Результат применения настроек «Запрет перекрытия» и «Влияние весов рёбер»

Опция *«Гравитация»,* как и в других пакетах, дает возможность стягивать узлы к центру графа или же наоборот отодвигать их от него. Также можно выделить узлы с сильной связью в сочетании с настройкой *«Влияние весов рёбер»* (Рис. 1.18).

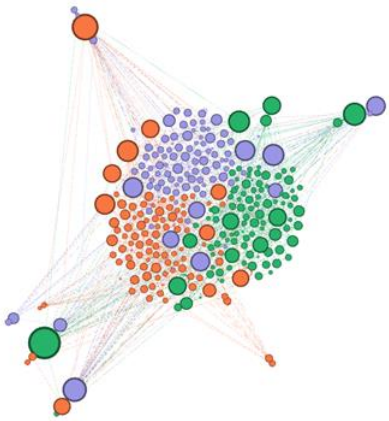

Рисунок 1.18 - Результат применения настроек «Гравитация» и «Влияние весов рёбер»

Выводы

Результаты анализа алгоритмов представлены в Таблице 1.1.

Таблица 1.1 - Сводная таблица по force-directed алгоритмам

Имя алгоритма	**Сильные стороны**	**Слабые стороны**	**Когда использовать**
Force Atlas	Включает много вариантов настройки и имеет высокий уровень точности	Может быть очень медленным и не подходит для больших сетей	Для анализа сетевых графов и для рассмотрения поведения сети
Force Atlas 2	Быстрее чем оригинал и может работать с очень большими сетями	Немного страдает общая точность	Подходит для сетевого анализа и исследований, обнаружения поведенческих паттернов
Fruchterman-Reingold	Точный и стремится строить графы лёгкие для чтения	Очень медленный и не подходит для больших сетей	Хорошо подходит для обобщенного вида на малый и средний размер сети
OpenOrd	Очень быстрый и может работать с очень большими сетями	Не очень точен на небольших сетях	Используется для быстрого понимания большой сетевой структуры
Yifan Hu	Достаточно быстр, по сравнению с другими методами	Не хватает опции для отдельной настройки отталкивания и притягивания	Для быстрого просматривания от малых до средних графов

1.3 Обзор существующих инструментов

На данный момент существует немало программных продуктов для анализа социальных сетей (Таблица 1.2).

Таблица 1.2 - Сводная таблица программных систем и библиотек для проведения анализа социальных сетей

Название	**Функционал**	**Входной формат**	**Выходной формат**	**Платформа**	**Условия распространения**
AllegroGraph	База графов. Визуализация RDF	RDF	EDF	Linux, Mac, Windows	Free и Commercial
EgoNet	Анализ эгоцентричных сетей	XML	CSV	Любая система с Java	Open Source
Gephi	Исследование, изменение графов и их визуализация	DOT,GML, GDF, GraphML, NET, GEXF, CSV, Database	GDF, GEXF, SVG, PNG	Любая система с Java 1.6 и OpenGL	Open Source (GPL3)
GraphStream	Библиотека работы со статическими и динамическим графами	DGS, DOT, GML, Edge list	DGS, DOT, GML, Images	Любая система с Java	Open Source
Graph-tool	Python-модуль для анализа и визуализации графов	DOT, GraphML	Более 25 форматов	GNU/Linux, MAC	Free (GPL3)
Graphviz	Визуализация графов	DOT	Более 25 форматов	Linux, Mac, Windows	Open Source (CPL)

sigma.js	Библиотека для визуализации графов	GEXF, JSON, XML	GEXF, XML	Поддержка JavaScript, HTML5 и WebGL	Open Source (MITL)
Mathematica	Анализ графов, расчет статистических данных, визуализация данных, оптимизация, распознавание изображения	Более 50 форматов	Более 50 форматов	Windows, Mac, Linux	Commerical
Wolfram Alpha	Анализ графов и временных выборок	Facebook API	Более 20 форматов	Современные браузеры	Free, Commercial

Критерии, по которым был проведен сравнительный анализ:

- Функциональность программы или библиотеки
- Число поддерживаемых входных и выходных форматов данных
- Поддерживаемые платформы
- Условия распространения

В результате анализа были выявлены следующие закономерности:

- Большинство систем приведено к единому стандарту и не имеют каких-либо особых опций для анализа именно социальных сетей
- Большая часть ПО предназначена для работы на персональном компьютере
- Требуются заранее подготовленные данные для анализа (исключение: Wolfram Alpha, он использует Facebook API)
- Лишь малая часть систем сочетает в себе сразу и возможность визуализации, и исследования графов

Глава 2. Адаптация методов раскладки графов для анализа пользовательских дискуссий

2.1 Постановка эксперимента

В работе поставлен эксперимент по проверке следующих методов раскладки графов:

- OpenOrd
- Yifan Hu
- Fruchterman-Reingold
- Force Atlas 2
- Circle pack layout

В рамках эксперимента использовалось программное обеспечение Gephi, так как оно имеет достаточно удобный и понятный интерфейс, включает в себя все необходимые укладки для проведения анализа, а также способно визуализировать графы очень большого размера.

Выбранные алгоритмы проверялись на четырех реальных пользовательских дискуссиях, развернувшихся в сети Twitter. Объём дискуссий составлял от 11 000 до 700 000 узлов.

Поданные на вход данные:

- Файл вершин в csv формате, столбцы: Id, Label, TweetCount
- Файл ребер в csv формате, столбцы: Source, Target

Подробное описание дискуссий представлено в следующем пункте.

2.2 Проведение эксперимента

Бирюлёво - Россия

Конфликт в Западном Бирюлёво — массовые уличные беспорядки в московском районе Бирюлёво на межэтнической почве. Поводом к данному событию послужило убийство москвича Егора Щербакова мигрантом азербайджанцев Орханом Зейналовым, произошедшее 10 октября 2013 года.

Стоит отметить, что беспорядки в Западном Бирюлёво стали самыми массовыми в России за последнее время. Всего полицией было задержано около 400 человек в ходе их подавления.

Нами будет рассмотрена дискуссия, развернувшаяся в сети Twitter по конфликту в Бирюлёво.

Всего пользователей приняло участие в дискуссии (вершин ориентированного графа дискуссии) – 11 429;

Всего направленных связей между пользователями дискуссии (направленных ребер графа дискуссии) – 20 106.

Все методы достаточно хорошо справились с визуализацией данного графа, стоит отметить Openord, Yufan Hu и Circle pack layout (Рис. 2.1, 2.2, 2.3, 2.4, 2.5).

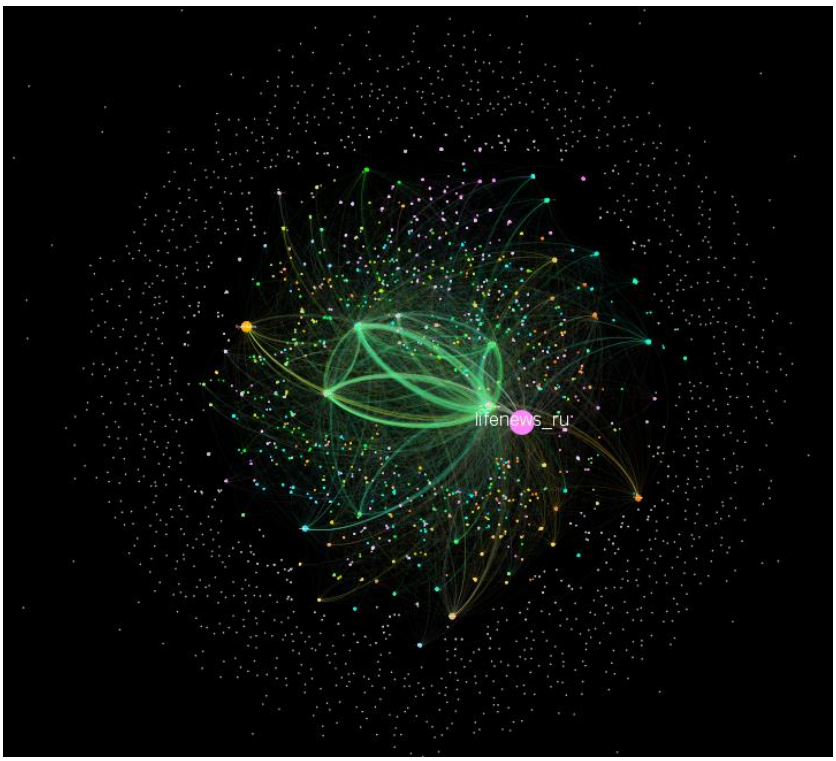

Рисунок 2.1 – Метод Openord. Liquid 25%, Expansion 75%, Cooldown 25%, Crunch 10%, Simmer 15%

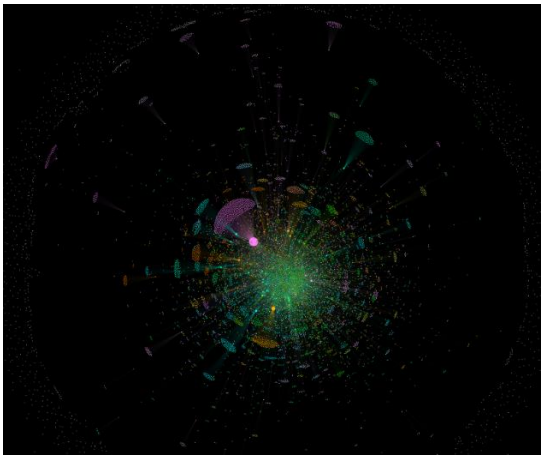

Рисунок 2.2 – Метод Fruchterman Reingold. Область 10000, гравитация 0, скорость 100

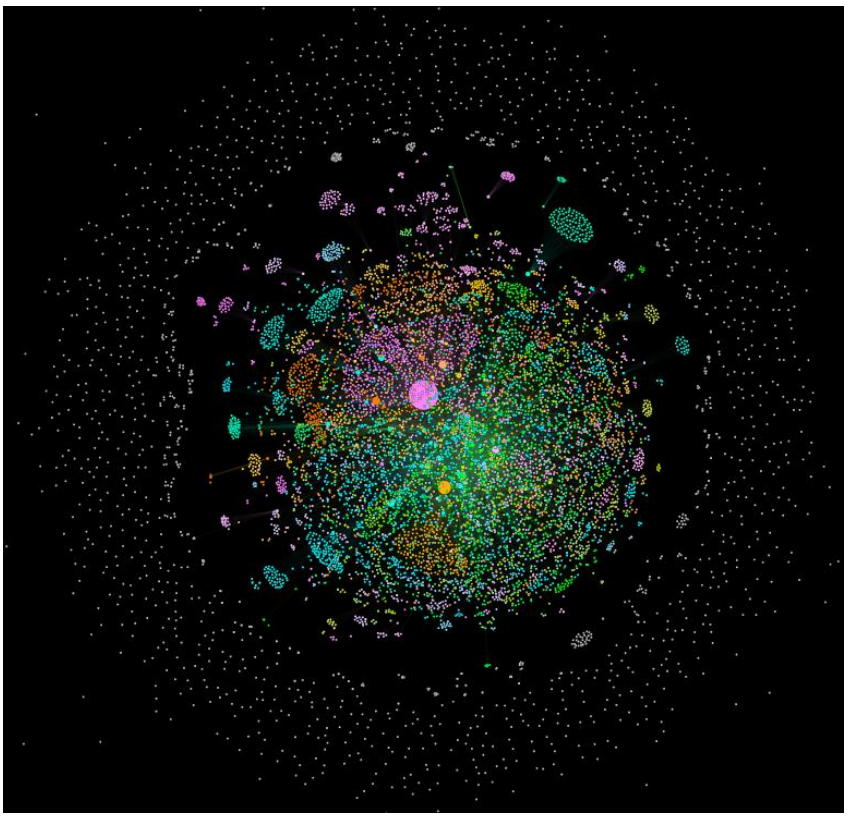

Рисунок 2.3 - Метод Yufan Hu. Оптимальное расстояние 200, относительная сила 0,2

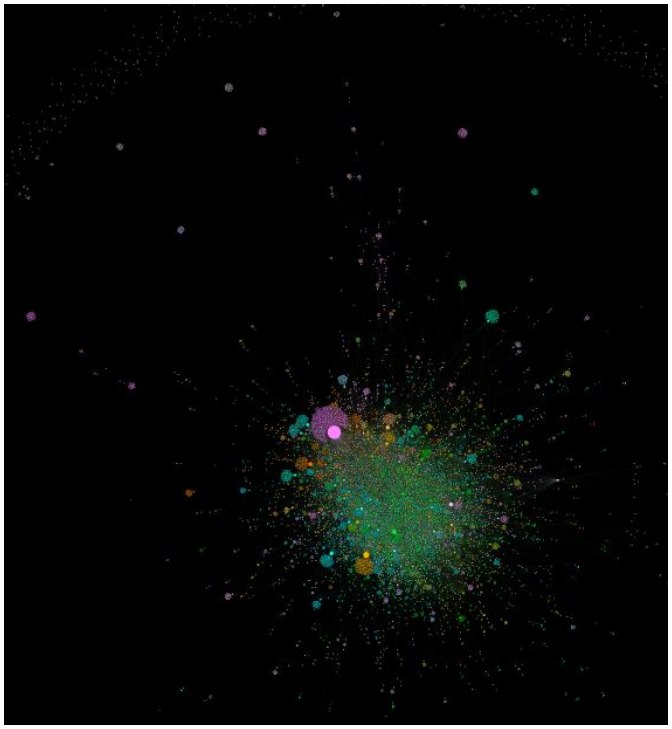

Рисунок 2.4 – Метод Force-Atlas 2, Scaling 10, Dissuade hubs true, Prevent overlap true, Gravity 1

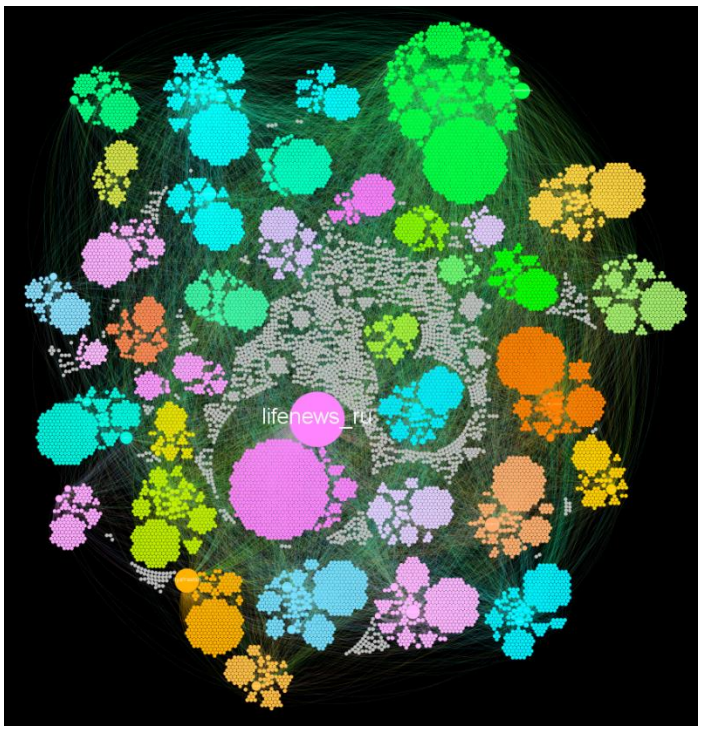

Рисунок 2.5 – Применение Circle pack layout, настроен по модулярности и степени узлов

Результаты:

- Основные аккаунты, из-за которых разрослась дискуссия - *lifenews_ru, IlyaYashin, izvestia_ru, mynameisphilipp, rodnyansky, polozovs, RT_russian, ForbesRussia*
- Основное число сообществ (по модулярности) – 8
- Всего сообществ – около 38.

Кельн – Германия

Беспорядки в Кельне – массовые новогодние нападения преимущественно на женщин в новогоднюю ночь 2016 года. В пострадавших оказалось около 1 000 женщин. Отмечается, что нападавшими были молодые мужчины «североафриканской» или «арабской» внешности, не говорящие по-немецки.

Нами будет рассмотрена дискуссия, развернувшаяся в сети Twitter по нападениям в Кельне.

Общее количество твиттов – 64 874;

Всего пользователей, опубликовавших свои твитты по заданному набору хэш-тэгов в период с 01.01.2016 по 31.01.2016 – 12 382;

Общее количество Likes у вышеуказанных пользователей – 304 931;

Общее количество Retwitts у вышеуказанных пользователей – 219 290;

Общее количество комментариев у вышеуказанных пользователей – 78 943;

Всего пользователей приняло участие в дискуссии (вершин ориентированного графа дискуссии) – 40 117;

Всего направленных связей между пользователями дискуссии (направленных ребер графа дискуссии) – 98 508.

С визуализацией данной дискуссии справились только методы OpenOrd, Force-Atlas 2 и Circle pack layout (из-за ее объёма), стоит отметить визуализацию Force-Atlas 2 и Circle pack layout (Рис. 2.6, 2.7, 2.8, 2.9).

Рисунок 2.6 – Кельн, нулевое состояние

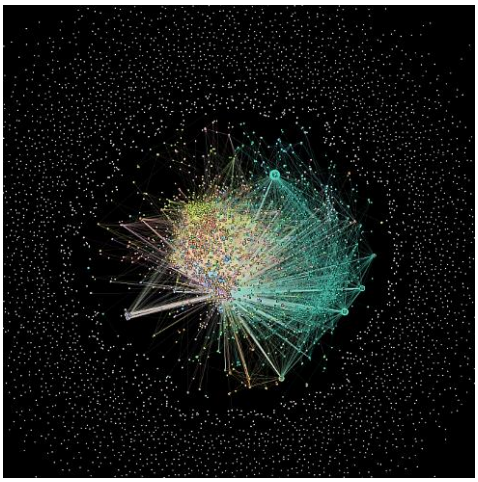

Рисунок 2.7 – Кельн. метод OpenOrd. Liquid 25%, Expansion 25%, Cooldown 25%, Crunch 10%, Simmer 15%

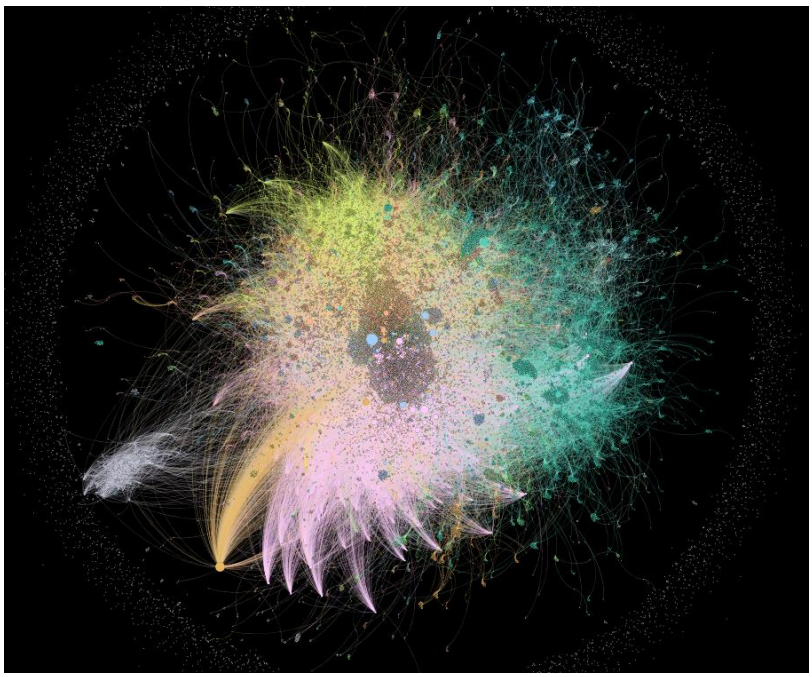

Рисунок 2.8 – Кельн, Метод Force-Atlas 2, Scaling 10, Dissuade hubs true, Prevent overlap true, Gravity 1

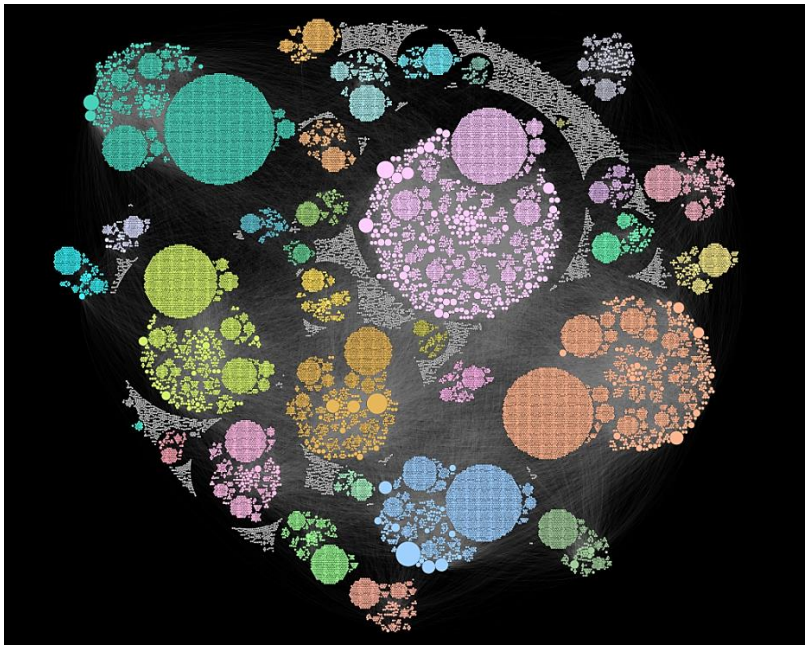

Рисунок 2.9 – Кельн, применен Circle pack layout, настроен по модулярности, степени узлов и количеству твиттов

Результаты:

- Основные аккаунты, из-за которых разрослась дискуссия - *chris65110, tagesschau, LarsWinter_, BukowskisNephew, MatthiasMeisner, aktuelle_stunde, _AltRight_, LadyAodh, gamergateblogde*
- Основное число сообществ (по модулярности) – 7
- Всего сообществ – около 21

Фергюсон – США

Беспорядки в Фергюсоне — вандализм, мародерство и поджоги, начавшиеся 9го августа 2014 года в городе Фергюсон штат Миссури.

Поводом к беспорядкам стало убийство безоружного чернокожего Майкла Брауна белокожим местным полицейским Дарреллом Уилсоном во время попытки ареста. Накал страстей возрос после того, как 24 ноября 2014 года большая часть жюри присяжных отказалась предъявлять обвинение Дарреллу Уилсону.

Нами будет рассмотрена дискуссия, развернувшаяся в сети Twitter по беспорядкам в Фергюсоне.

Всего пользователей приняло участие в дискуссии (вершин ориентированного графа дискуссии) – 169 677;

Всего направленных связей между пользователями дискуссии (направленных ребер графа дискуссии) – 334 050.

С таким большим объёмом дискуссии снова справились три метода: OpenOrd, Force-Atlas 2 и Circle pack layout, стоит отметить визуализацию Force-Atlas 2 и Circle pack layout, которые оказались достаточно точными (Рис. 2.10, 2.11, 2.12).

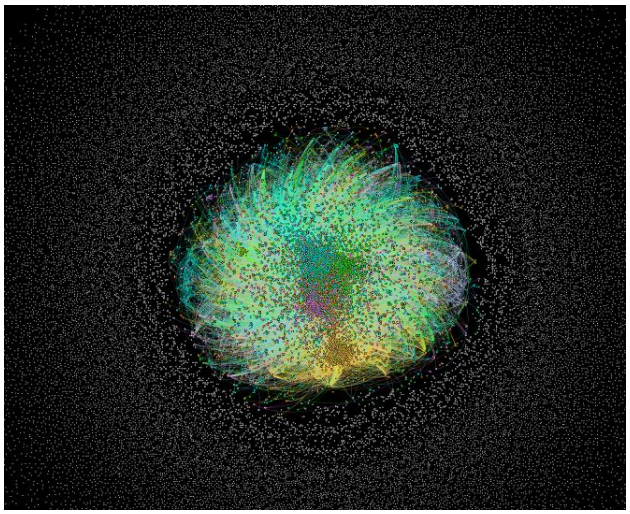

Рисунок 2.10 – Фергюсон, метод OpenOrd. Liquid 25%, Expansion 25%, Cooldown 25%, Crunch 10%, Simmer 15%

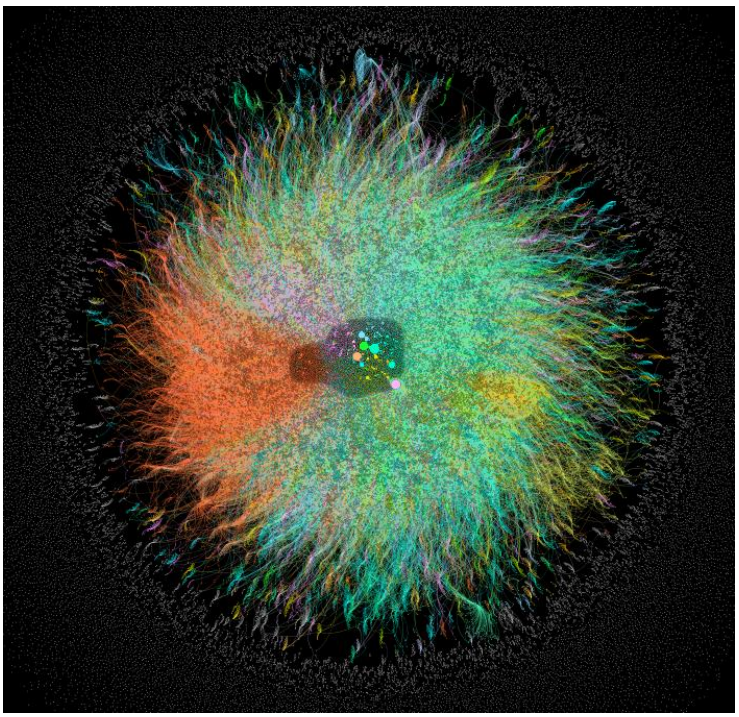

Рисунок 2.11 – Фергюсон, Метод Force-Atlas 2, Scaling 10, Dissuade hubs true, Prevent overlap true, Gravity 1

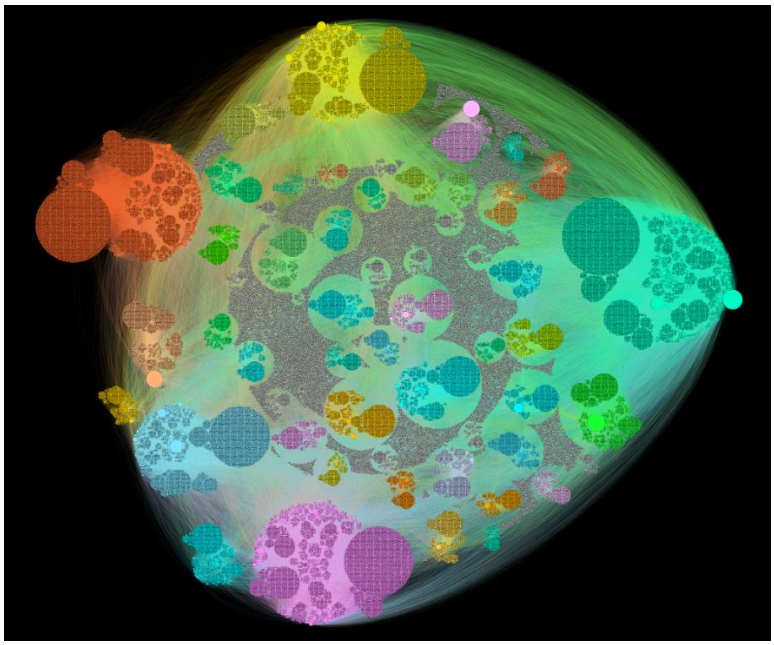

Рисунок 2.12 – Фергюсон, применен Circle pack layout, настроен по модулярности, степени узлов и количеству твиттов

Результаты:

- Основные аккаунты, из-за которых разрослась дискуссия – *disetv, deray, Nettaaaaaaaa, dolphin_LS, ryanjreilly, kvayozone, CassandraRules*
- Основное число больших групп (по модулярности) – 5
- Всего сообществ – около 42.

Шарли Эбдо – Франция

Трагедия в редакции журнала Charlie Hebdo случилась 7го января 2015 года в Париже. Террористический акт совершили братья Куаши: Саид и Шериф. Поводом для такого зверства стала карикатура на пророка Мухаммеда, нарисованная в редакции. По итогу погибло 12 человек, а также 11 оказались ранены.

Нами будет рассмотрена дискуссия, развернувшаяся в сети Twitter по террористической атаке в Шарли Эбдо.

Общее количество твиттов – 420 080;

Всего пользователей, опубликовавших свои твитты по заданному набору хэш-тэгов в период с 07.01.2015 по 10.01.2015 – 266 904;

Общее количество Likes у вышеуказанных пользователей – 1 330 728;

Общее количество Retwitts у вышеуказанных пользователей – 1 851 412;

Общее количество комментариев у вышеуказанных пользователей – 206 344;

Всего пользователей приняло участие в дискуссии (вершин ориентированного графа дискуссии) – 719 503;

Всего направленных связей между пользователями дискуссии (направленных ребер графа дискуссии) – 981 131.

С визуализацией самой большой дискуссии справились три метода: OpenOrd, Force-Atlas 2 и Circle pack layout, и снова стоит отметить работу Force-Atlas 2 и Circle pack layout (Рис. 2.13, 2.14, 2.15).

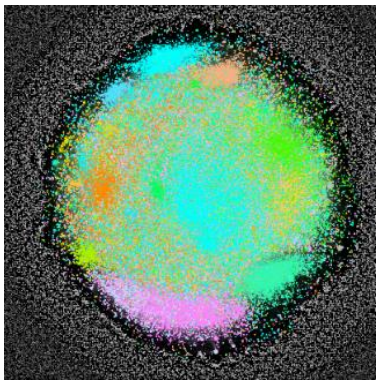

Рисунок 2.13 – Шарли Эбдо, метод OpenOrd. Liquid 25%, Expansion 25%, Cooldown 25%, Crunch 10%, Simmer 15%

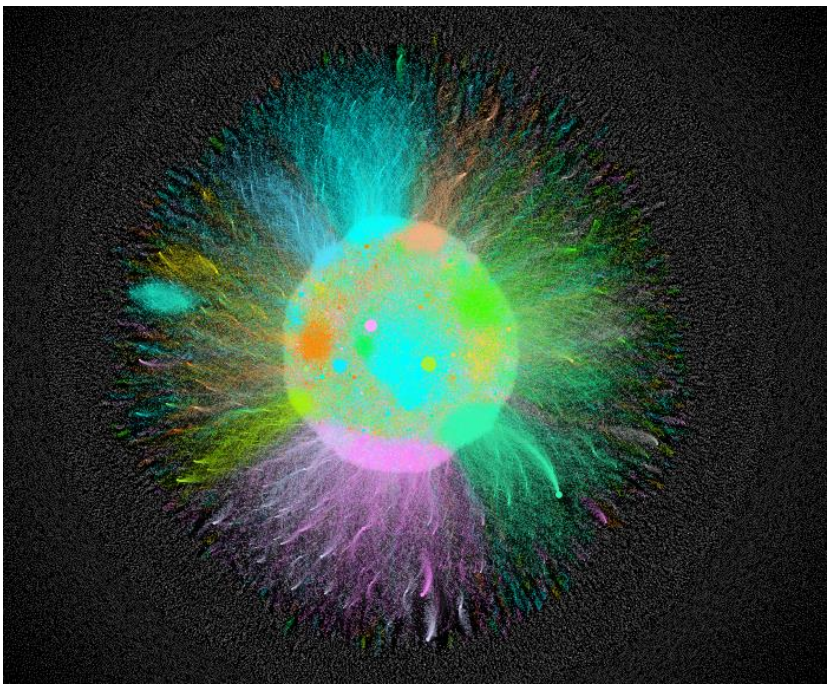

Рисунок 2.14 – Шарли Эбдо, Метод Force-Atlas 2, Scaling 10, Dissuade hubs true, Prevent overlap true, Gravity 1

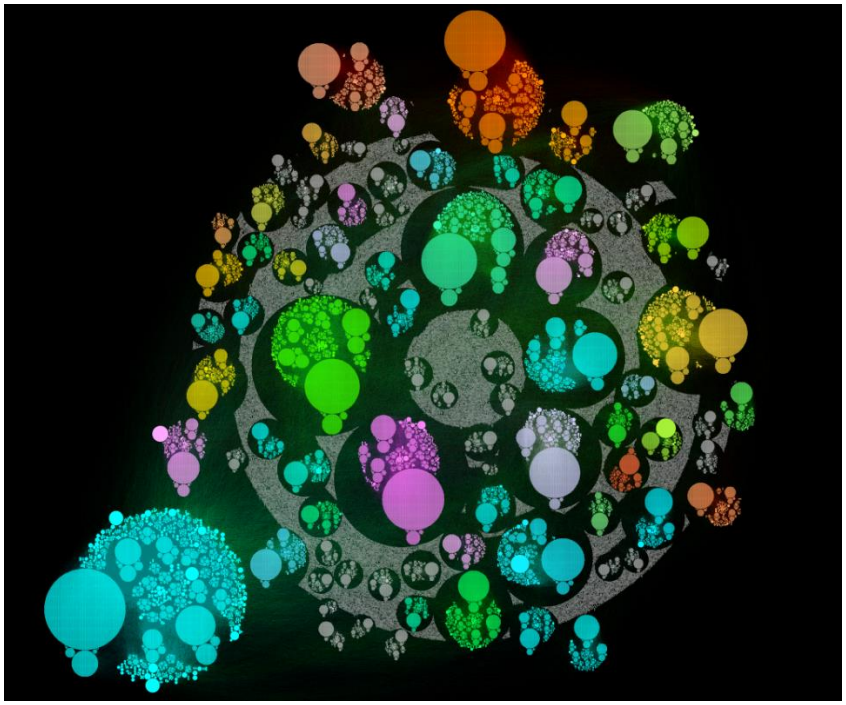

Рисунок 2.15 – Шарли Эбдо, применен Circle pack layout, настроен по модулярности, степени узлов и количеству твиттов

Результаты:

- Основные аккаунты, из-за которых разрослась дискуссия – *Le_Figaro, Bipartisanism, Limportant_fr, RMCinfo, afpfr, Serv_Pubblico*
- Основное число больших групп (по модулярности) – 11
- Всего сообществ – около 74.

2.3 Результаты

Программное обеспечение Gephi действительно очень удобное для визуализации графов и обладает рядом положительных свойств, однако стоит отметить один его существенный недостаток: отсутствие возможности работы с графом уже после укладки (перемещение узлов, возможность выделить группу узлов по каким-либо признакам и так далее).

В Таблице 2.1 собрана информация по времени визуализации каждого метода той или иной дискуссии.

Таблица 2.1 - Сводная таблица по экспериментальным исследованиям

Набор данных	**Количество узлов**	**Количество ребер**	**Имя алгоритма**	**Время выполнения**
Бирюлёво – Россия	11 429	20 106	OpenOrd	3 мин
			Yifan Hu	4 мин
			Fruchterman-Reingold	5 мин
			Force Atlas 2	10 мин
			Circle pack layout	3 с
Кельн – Германия	40 117	98 508	OpenOrd	11 мин
			Yifan Hu	–
			Fruchterman-Reingold	–
			Force Atlas 2	43 мин
			Circle pack layout	5 с
Фергюсон – США	169 677	334 050	OpenOrd	46 мин
			Yifan Hu	–
			Fruchterman-Reingold	–
			Force Atlas 2	2 ч 27 м
			Circle pack layout	10 с
Шарли Эбдо – Франция	719 503	981 131	OpenOrd	3 ч 24 мин
			Yifan Hu	–

			Fruchterman-Reingold	–
			Force Atlas 2	Остановка после 4 ч
			Circle pack layout	20 с

Для небольших графов отлично подходит любая force-directed укладка, однако стоит отметить Yufan Hu и Fruchterman-Reingold – являются достаточно быстрыми и точными.

Для средних и больших графов хорошим решением будут OpenOrd и ForceAtlas 2. OpenOrd достаточно быстро наглядно представит структуру сети, а ForceAtlas 2 даст большую точность, правда, с потерей времени.

Также лучшим среди укладок стал пакет Circle pack layout, настроенный по модулярности (объединение узлов в сообщества) и по степени узлов. Данный пакет способен за считанные секунды представить даже самый большой граф в пригодный для анализа вид.

Глава 3. Разработка программного комплекса для визуализации пользовательских дискуссий

В работе поставлена задача по разработке программного комплекса, состоящего из конкретных методов и инструментов, для работы с дискуссией в виде веб-графа, а также адаптации этих инструментов к анализу пользовательских дискуссий в социальных сетях.

На основании найденных недостатков у Gephi было разработано и реализовано собственное клиент-серверное приложение, решающее их.

В рамках разработки приложения был выбран метод раскладки Force Atlas 2. Так как он является одним из самых точных методов визуализации графов, а также хорошо себя зарекомендовал на всех четырех дата-сетах.

Тестирование выбранного метода проводилось на дата-сете Бирюлёво – Россия.

Инструменты, созданные в рамках реализации приложения:

- Интерактивная карта по сообществам (мера – модулярность)
- Интерактивная карта по степеням вершин

Поданные на вход данные: дата-сет Бирюлёво – Россия в формате json, координаты вершин графа получены с помощью укладки Circle pack layout в Gephi.

Стек технологий:

- Языки: java-script, html, css;
- Платформа: node.js;
- Использованные библиотеки: sigma.js, g6.js.

Подробное описание функционала приложения представлено в пункте 3.1.

3.1 Функциональные характеристики приложения

Меню приложения дает возможность навигации (Рис. 3.1):

- Переход к интерактивной укладке методом Force Atlas 2
- Переход к интерактивной карте по сообществам (на основе метрики – модулярность) [10]
- Переход к интерактивной карте по степеням

Рисунок 3.1 – Интерактивное меню приложения

Кнопка *Layout Force Atlas 2* переключает нас на страницу с интерактивной укладкой алгоритмом Force Atlas 2. Данный метод реализован с помощью библиотеки g6.js [13].

В начале мы можем увидеть вершины графа, сбившиеся в одну точку (начальное состояние) (Рис. 3.2), затем граф интерактивно распадается в укладку (Рис. 3.3). При наведении мыши на узел высвечивается имя пользователя в сети, его входящая и исходящая степень (Рис. 3.4). Размер узла соответствует степени вершины. Также после завершения алгоритма, узлы можно перемещать, и посредством этого настраивать вид укладки по своему усмотрению – работать с дискуссией.

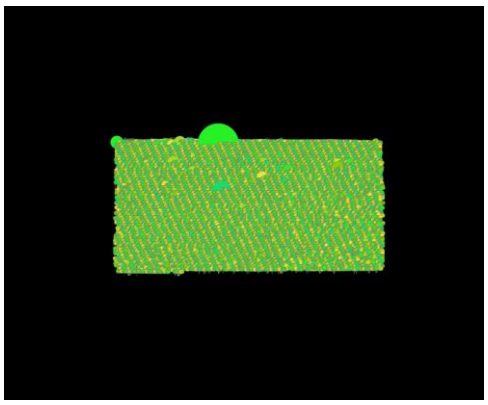

Рисунок 3.2 – Начальное состояние графа

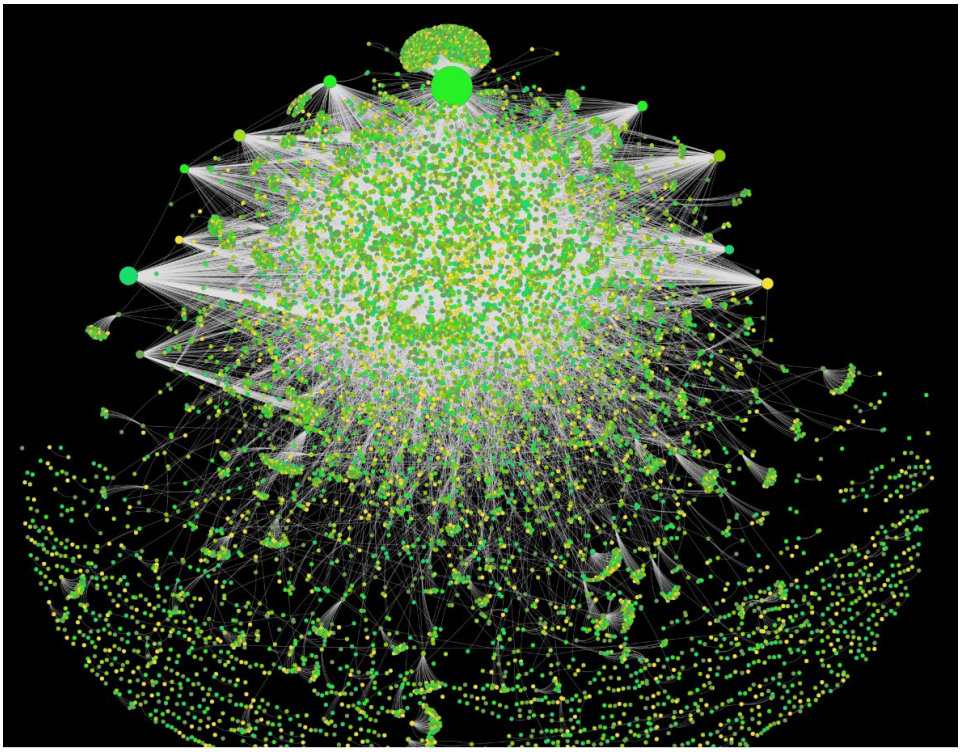

Рисунок 3.3 – Визуализации дискуссии Бирюлёво, методом Force Atlas 2

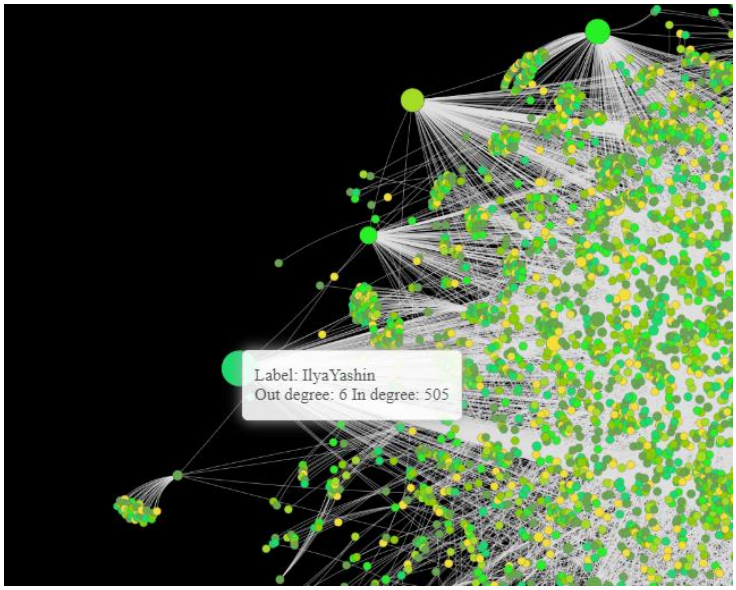

Рисунок 3.4 – Информационная панель при наведении мыши на узел

Кнопка *Interactive map by communities* переключает нас на интерактивную карту по сообществам (Рис. 3.5). Данная карта реализована с помощью библиотеки sigma.js [2].

Возможности карты:

- Просмотр легенды карты
- Интерактивное перемещение
- Просмотр информации о любом узле (список входящих/исходящих ребер, характеристики узла) (Рис. 3.6)
- Нахождение узла по имени (Рис. 3.7)
- Возможность выделения группы узлов по модулярности (Рис. 3.8), а также просмотр состава данного сообщества (Рис. 3.9)

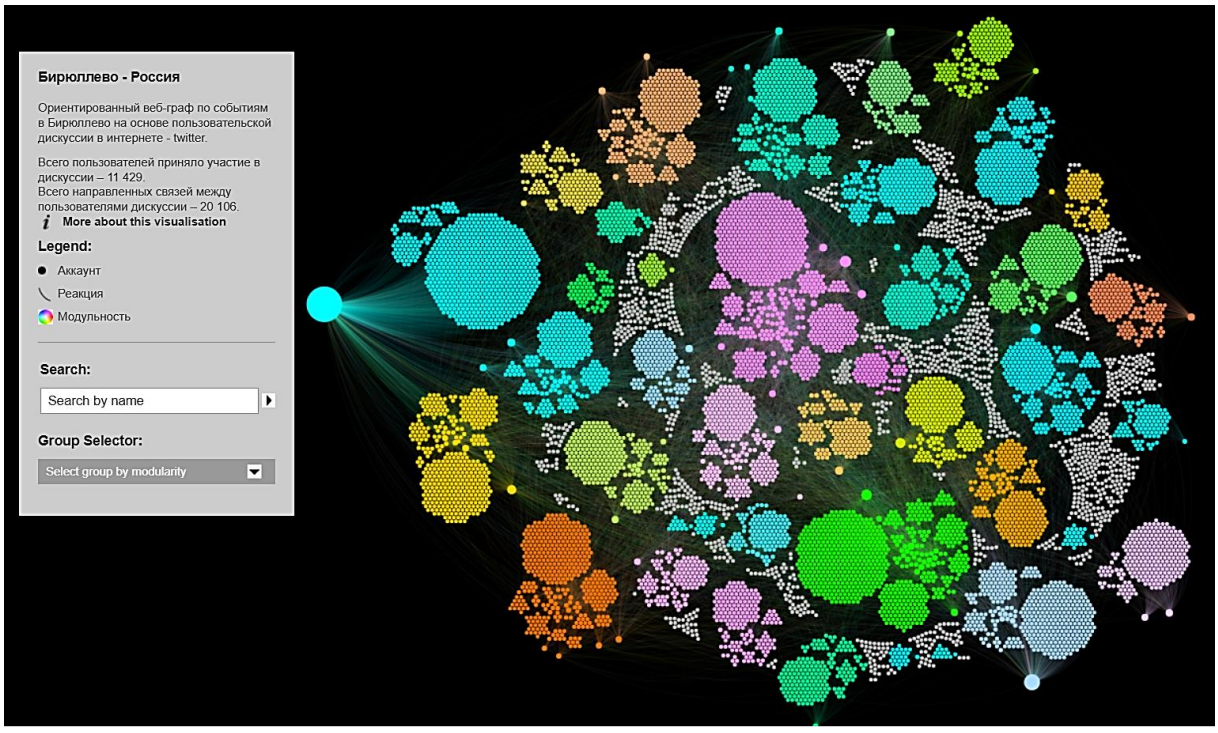

Рисунок 3.5 – Интерактивная карта по сообществам

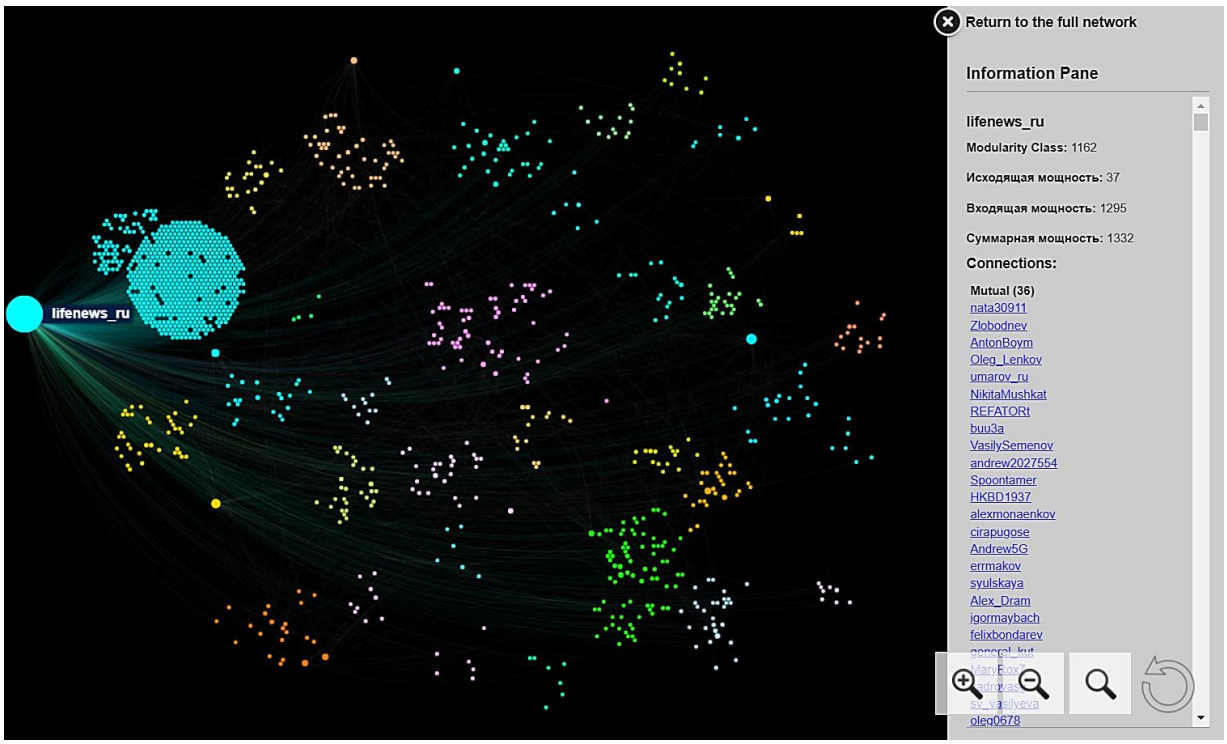

Рисунок 3.6 – Просмотр информации об узле lifenews_ru

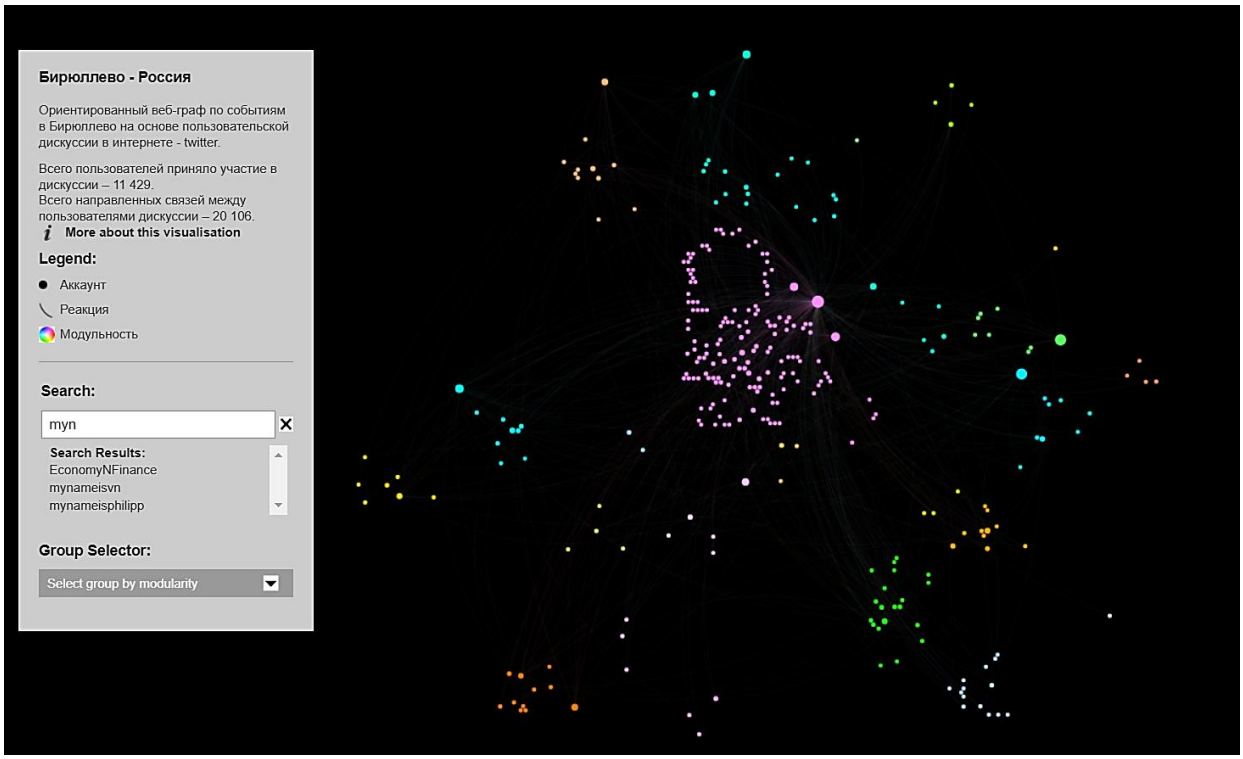

Рисунок 3.7 – Поиск узла по имени

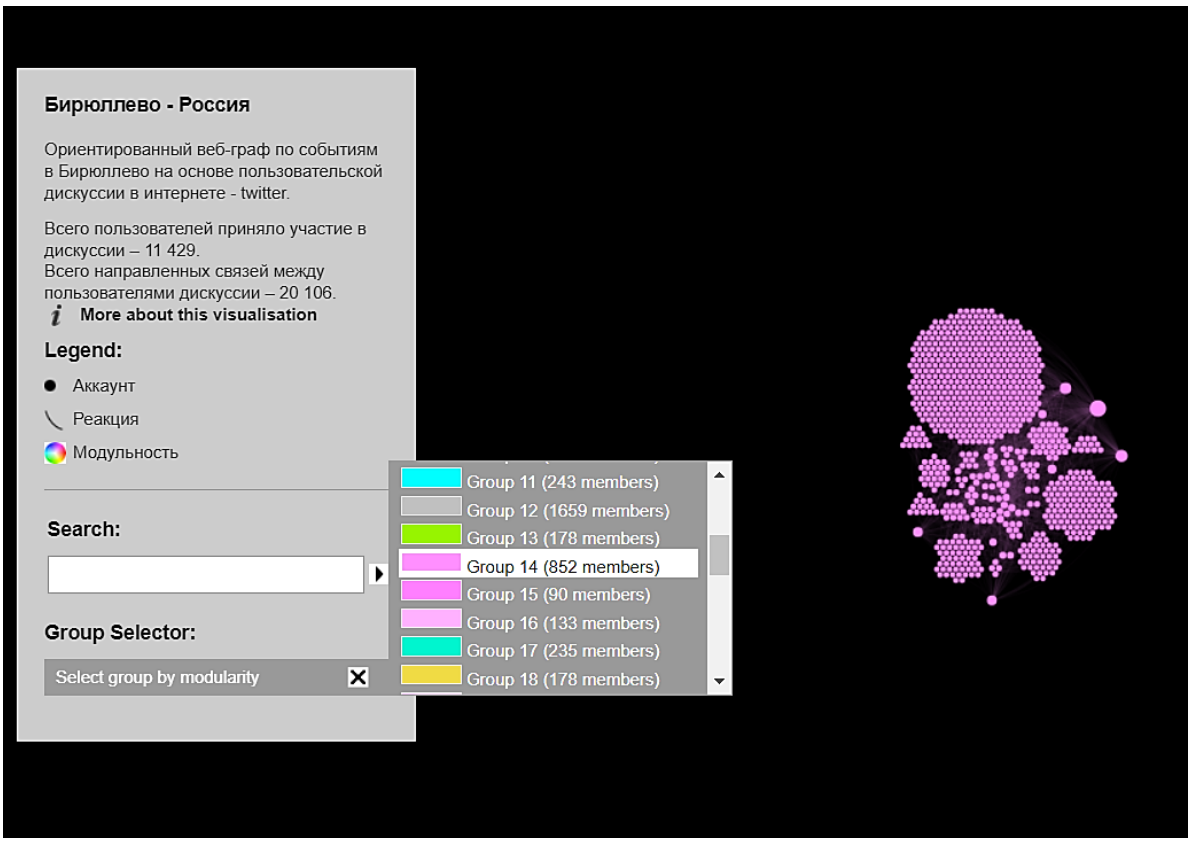

Рисунок 3.8 – Выделение на карте определенного сообщества

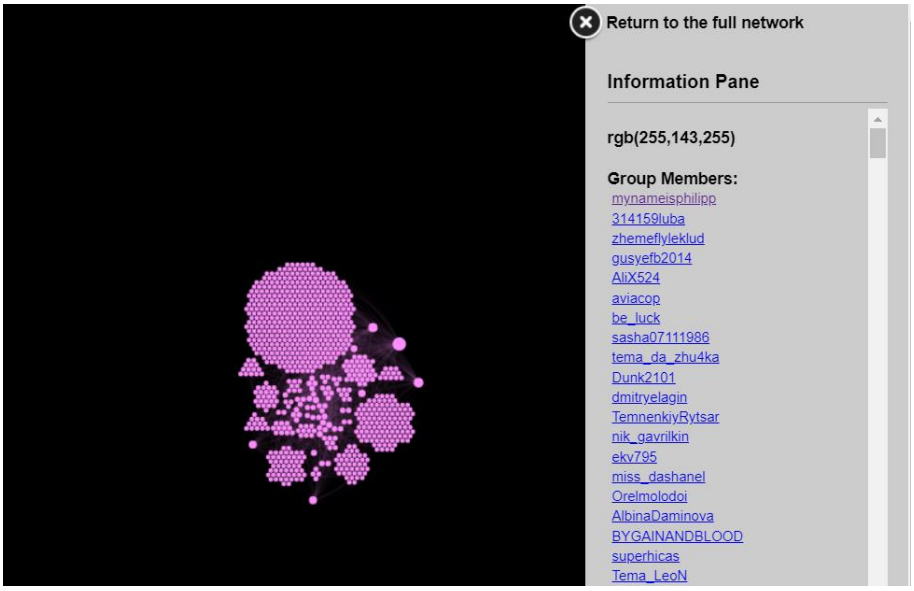

Рисунок 3.9 – Просмотр состава сообщества

Кнопка *Interactive map by gegree* переключает нас на интерактивную карту по степеням. Данная карта так же реализована с помощью библиотеки sigma.js [2].

Возможности карты идентичны предыдущей, однако присутствует единственное отличие:

- Возможность выделить группу узлов по степеням (Топ 10 (Рис. 3.10), пользователи с нулевой степенью, с единичной (Рис. 3.11) и так далее).

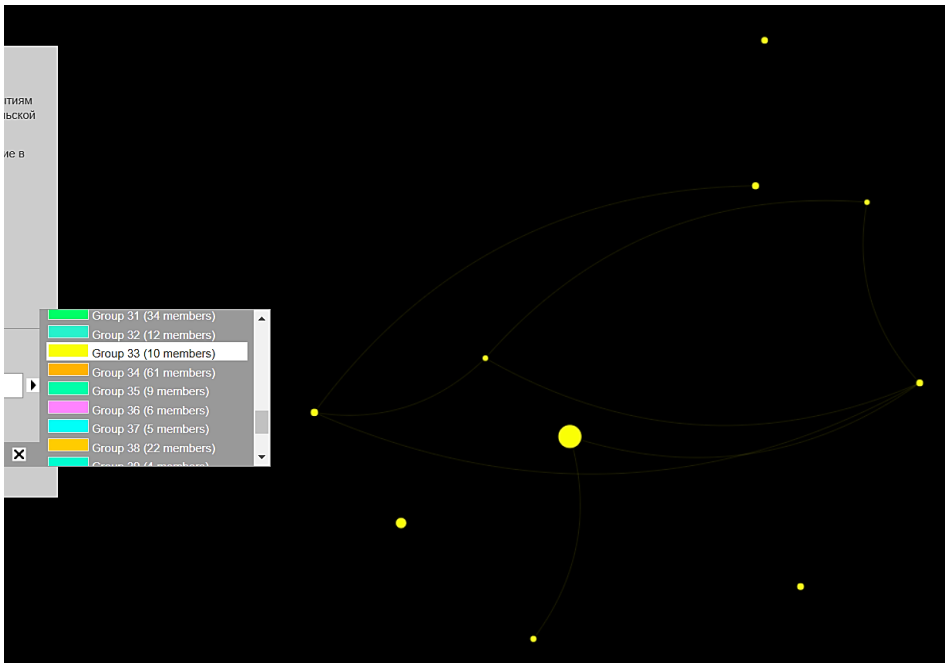

Рисунок 3.10 – Топ 10 пользователей в сети по популярности

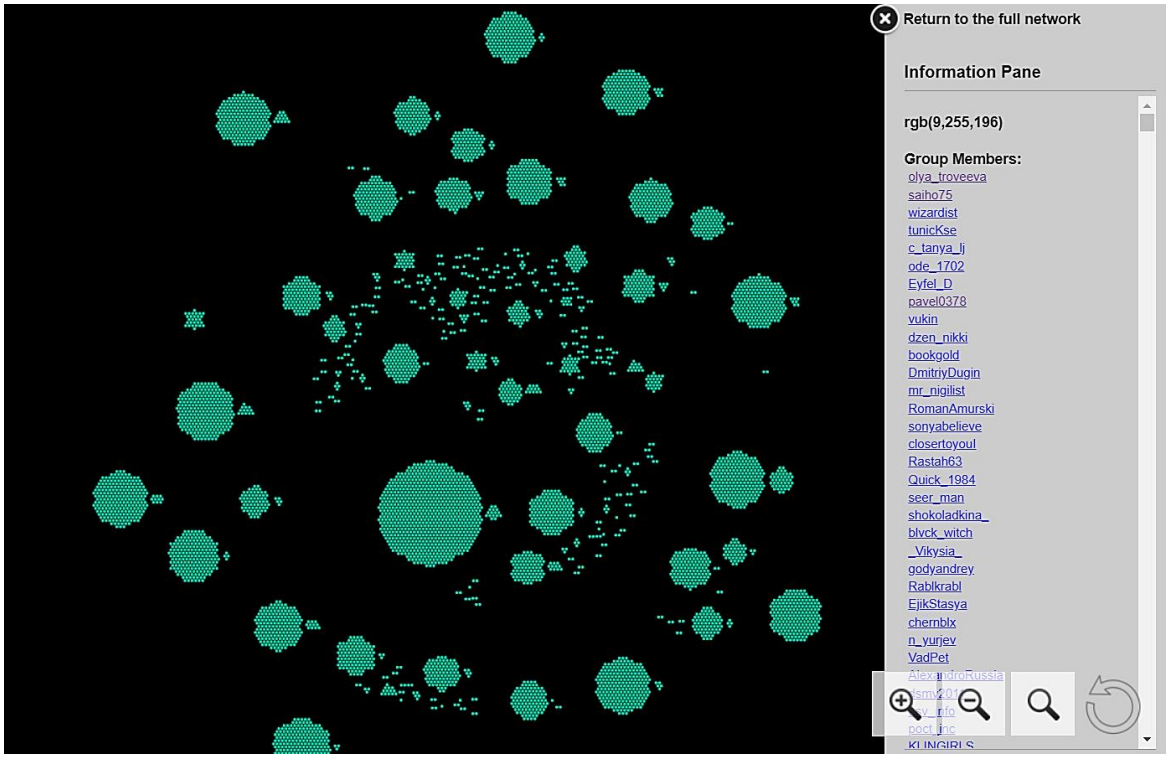

Рисунок 3.11 – Все пользователи с единичной степенью

В процессе реализации приложения, были также созданы интерактивные карты по сообществам для оставшихся трех дата-сетов. Кельн (Рис. 3.12), Фергюсон (Рис. 3.13), Шарли Эбдо (Рис. 3.14).

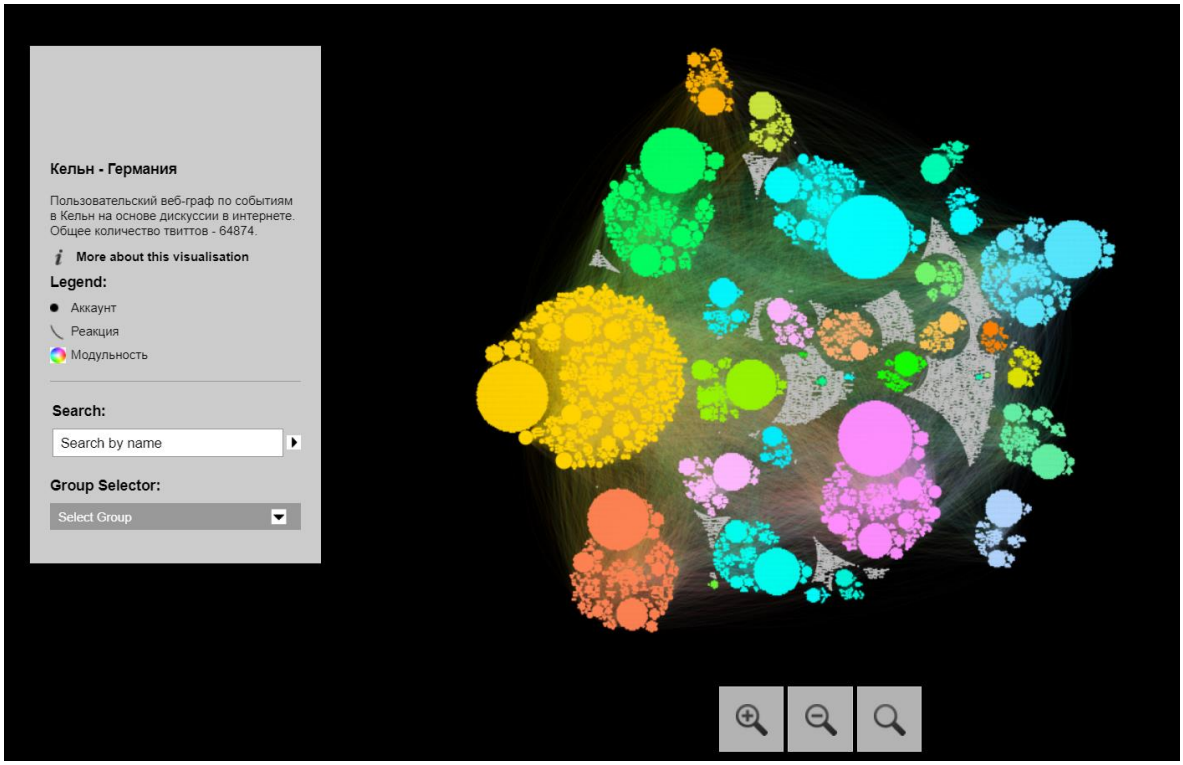

Рисунок 3.12 – Интерактивная карта веб-графа Кельна

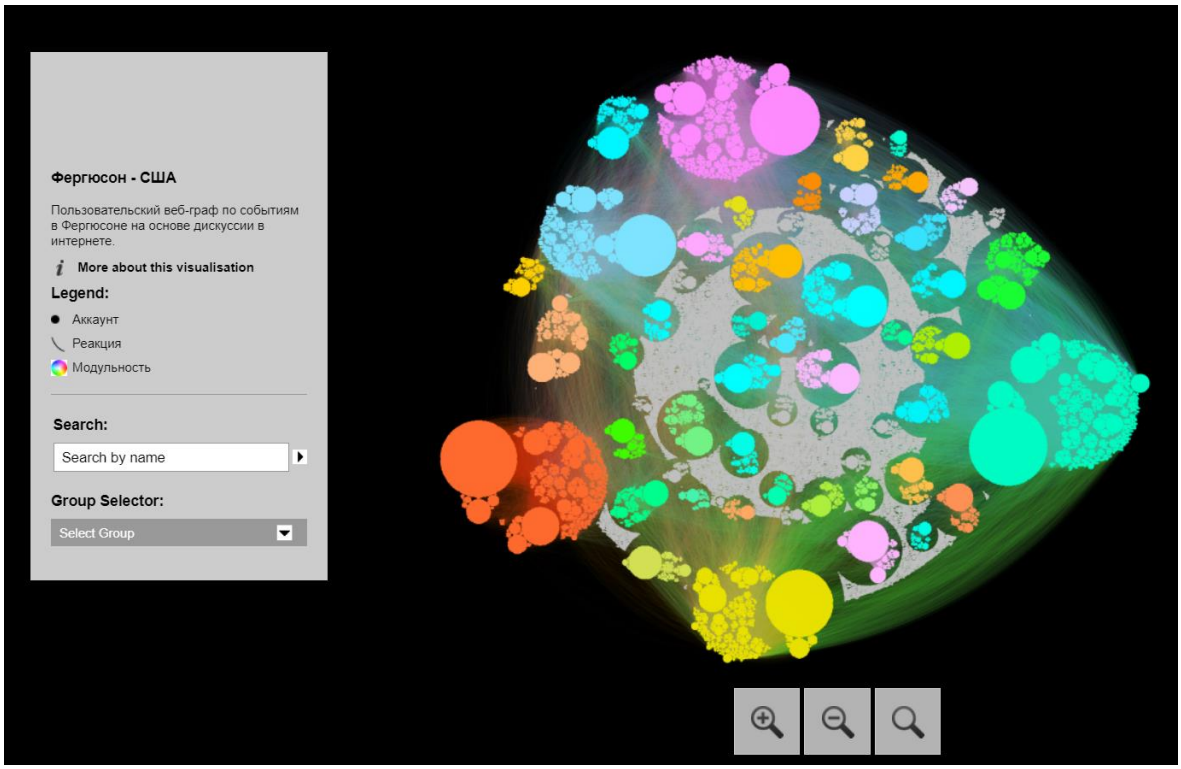

Рисунок 3.13 – Интерактивная карта веб-графа Фергюсона

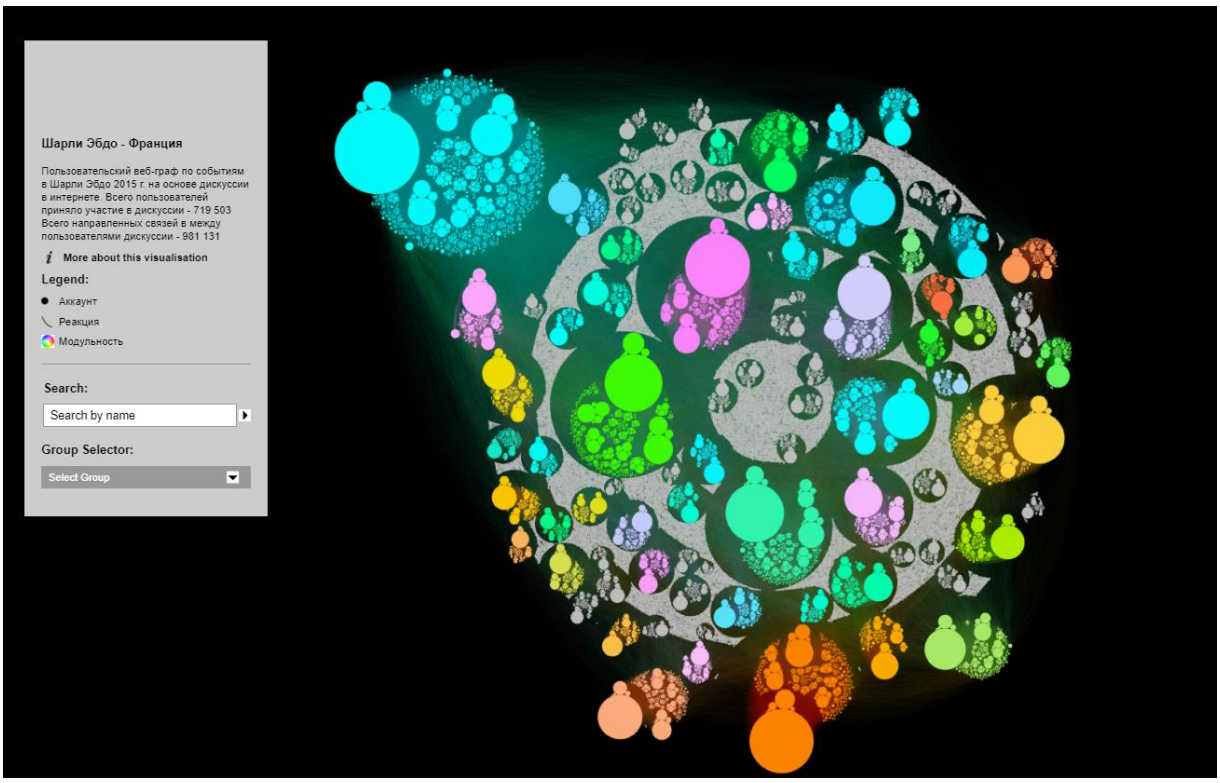

Рисунок 3.14 – Интерактивная карта веб-графа Шарли Эбдо

Заключение

Результаты работы

В данной выпускной квалификационной работе были выполнены следующие задачи:

- Проведен обзор научной литературы по теме исследования
- Проведен обзор алгоритмов раскладки графов
- Проведен обзор существующих решений для визуализации графов
- Проведено тестирование и апробация существующих алгоритмов раскладки графов на четырех реальных дискуссиях разного объёма, а также выявлены эффективные методы визуализации графов – Force Atlas 2 и Circle pack layout
- Разработан и реализован программный комплекс на основе клиент-серверного приложения, состоящий из алгоритма Force Atlas 2 и двух интерактивных карт (по сообществам и по степеням вершин), позволяющий эффективно взаимодействовать со структурой дискуссии в виде пользовательского веб-графа, а также имеющий возможность просмотра дискуссии
    - Кодовая база приложения доступна по данной ссылке: https://github.com/IrinaS-77/Analysis_of_the_Biryulyovo-Russia_discussion
- Созданы интерактивные карты по сообществам для четырех дискуссий. Ознакомиться с ними можно перейдя по следующим ссылкам:
    - Бирюлево: https://irinas-77.github.io/Biryulyovo_web-graph/network/
    - Кельн: https://irinas-77.github.io/Cologne_web-graph/network/
    - Фергюсон: https://irinas-77.github.io/Ferguson_web-graph/network/
    - Шарли Эбдо: https://irinas-77.github.io/Charlie_Hebdo_web-graph/network/

Перспективы развития

Данная дипломная работа несет следующие перспективы развития:

- Расширение функционала приложения
    - Добавление не менее эффективных методов визуализации графов в программный компонент
    - Принятие на вход различных форматов данных: csv, json, xml, gexf и другие
    - Добавление возможностей в интерактивные карты: скрытие слабо активных пользователей (число связей меньше 3, например)
- Тестирование работы приложения на большем числе дискуссий разного объёма

Список литературы

[1] Yifan Hu. Efficient and High-Quality Force-Directed Graph Drawing. – Wolfram Research Inc, 2005.

[2] Applying Graph Layout Techniques to Web Information Visualization and Navigation / Wei Lai, Xiaodi Huang, Quang Vinh Nguyen, Mao Lin Huang. – CGIV, 2007.

[3] Носов, В.И. Элементы теории графов. – Новосибирск: СГУТИ, 2008.

[4] Richard Klavans. OpenOrd: An Open-Source Toolbox for Large Graph Layout / Richard Klavans, Kevin Boyack, Shawn Martin. – Proceedings of SPIE – The International Society for Optical Engineering, January 2011.

[5] ForceAtlas2, A Continuous Graph Layout Algorithm for Handy Network Visualization / Mathieu Jacomy, Sebastien Heymann, Tommaso Venturini, Mathieu Bastian. – August 1, 2012.

[6] Jie Hua. Drawing Large Weighted Graphs Using Clustered Force-Directed Algorithm / Jie Hua, Mao Lin Huang, Quang Vinh Nguyen. – 18th International Conference on Information Visualisation, 2014.

[7] Ivan Blekanov. Comparing influencers: activity vs. connectivity measures in defining key actors in twitter ad hoc discussions on migrants in Germany and Russia / Blekanov I.S., Bodrunova S.S, Litvinenko A.A. – Springer Verlag, 2017.

[8] Ivan Blekanov. Measuring influencers in twitter ad-hoc discussions: Active users vs. internal networks in the discourse on biryuliovo bashings in 2013 / Blekanov I.S., Bodrunova S.S, Maksimov A. – Institute of Electrical and Electronics Engineers Inc., 2017.

[9] Se-Hang Cheong. Snapshot Visualization of Complex Graphs with Force-Directed Algorithms / Se-Hang Cheong, Yain-Whar Si. – IEEE International Conference on Big Knowledge, 2018.

[10] Zhenhua Huang. Visualizing complex networks by leveraging community structures / Zhenhua Huang, Junxian Wu, Yangyang Zhao. – Physica A: Statistical Mechanics and its Applications, 2020.

[11] Derek L. Hansen. Installation, orientation, and layout / Derek L. Hansen, Itai Himelboim. – Analyzing Social Media Networks with NodeXL (Second Edition), 2020.

[12] Ivan Blekanov. Detection of Hidden Communities in Twitter Discussions of Varying Volumes / Ivan Blekanov, Svetlana S. Bodrunova, Askar Akhmetov. – Selected Papers from the 9th Annual Conference "Comparative Media Studies in Today's World" (CMSTW'2021)), 2021.

[13] Yanyan Wang. G6: A web-based library for graph visualization / Yanyan Wang, Zhanning Bai, Wei Chen. – Visual Informatics Available online, 2021.