**Анализ графа пользователей Хабрахабра**

Описание задачи: построить граф пользователей сообщества «Машинное обучение» сайта Хабрахабр.

Требования: получить граф пользователей (возможно несколько графов) и программу для получения графов других сообществ/расширения существующего графа.

Полученный граф будет иметь такую структуру: вершинами являются пользователи, хотя бы 1 раз проявлявшие активность в данном сообществе, а ребрами соединяться будут те пользователи, которые хотя бы 1 раз контактировали друг с другом (то есть в данном случае оставляли комментарии к комментариями или статьям друг друга). Также можно применять различные фильтры и усиления условий на наличие ребра, например, ребро существует только тогда, когда пользователи контактировали друг с другом не менее 5 раз.

Пути решения проблемы:

1. Собрать данные с помощью парсера или API
2. Объединить полученные данные в базу данных
3. (при необходимости) конвертировать полученные данные в нужный формат
4. Визуализировать : возможные решения – network, neo4j, gephi, yEd

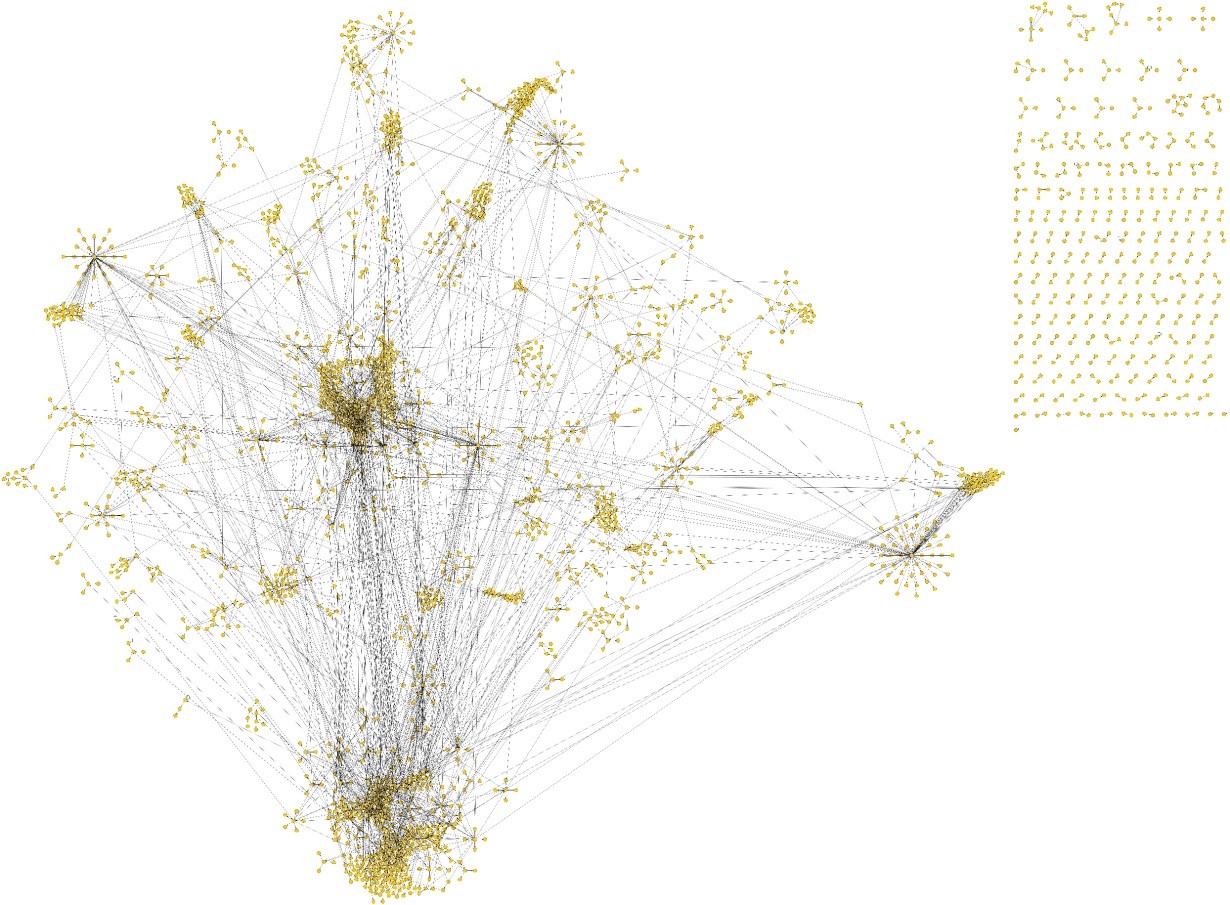
Алгоритмы отрисовки графов:

Силовой алгоритм (force-directed algorithm). В этом алгоритме каждой вершине назначается «заряд», аналогичный электрическому заряду в реальном мире. Ребрам назначается сила притяжения (т.е. они выступают в роли пружин, соединяющих тела (вершины) в пространстве). Получается, что вершины, соединенные ребрами, будут притягиваться друг к другу из-за упругости пружины между ними, а вершины, не соединенные ребром, будут отталкиваться друг от друга из-за силы отталкивания зарядов. Задача состоит в том, чтобы привести систему в состояние равновесия (когда все силы уравновешены) или, если равновесие невозможно, в состояние минимальной энергии. Силовой алгоритм работает примерно за O(n^3), где n – количество вершин в графе, так как алгоритм проходит все вершины за линейное время, и в каждой итерации необходимо вычислить отталкивающие силы для каждой пары вершин.

Другой способ - круговая схема распределения вершин графа на плоскости: весь граф разбивается на кластеры, а затем вершины каждого кластера размещаются на окружности соответствующего диаметра. Далее необходимо оптимизировать расположение окружностей относительно друг друга, уменьшая количество пересечений ребер. Для алгоритма, отрисовывающего круговой граф, существуют алгоритмы, справляющиеся за квадратичное от количества вершин время (O(n^2)).

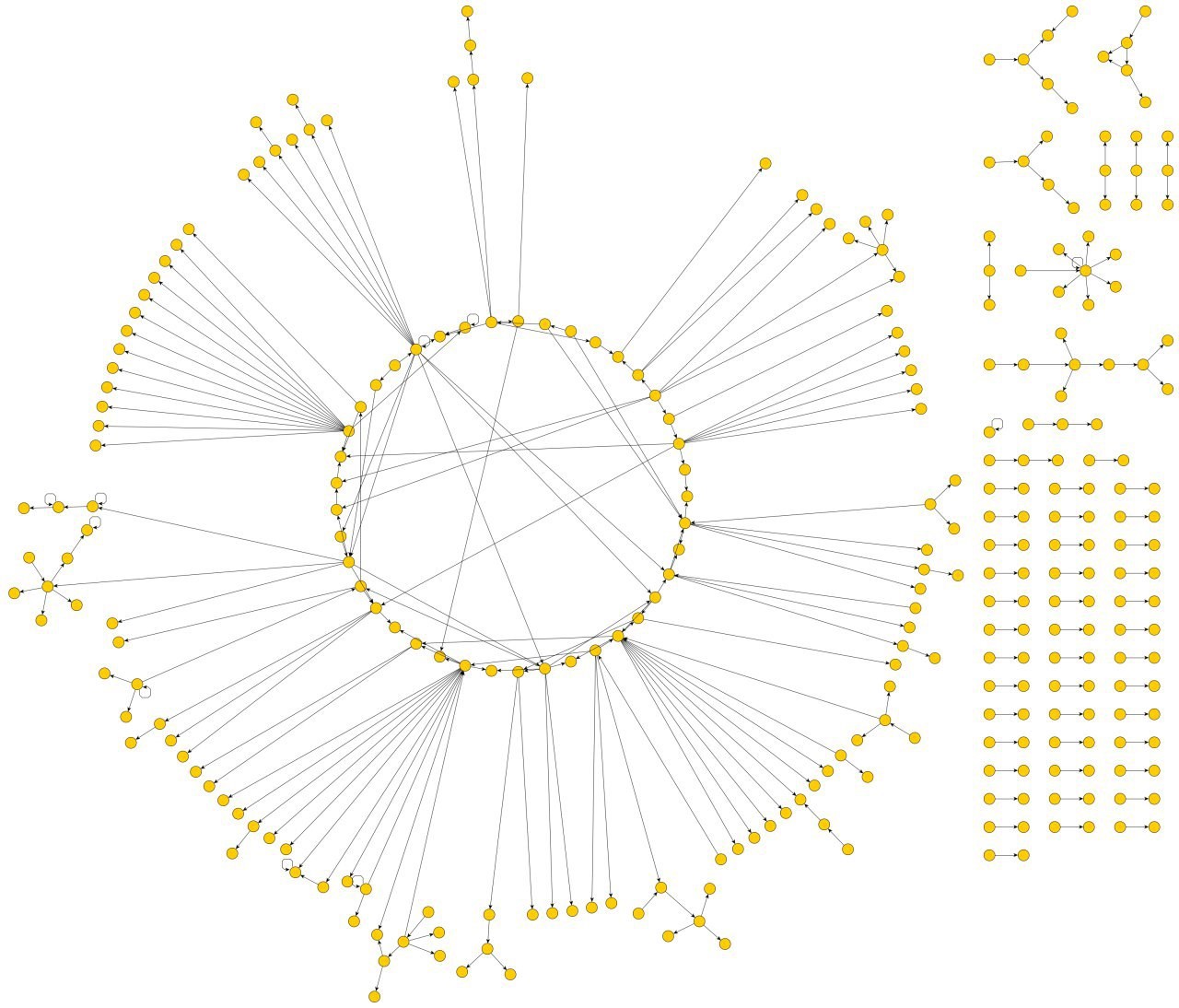
Полученные результаты:

С помощью парсера удалось собрать данные о 1868 уникальных пользователях и 33817 комментариях, оставленных этими пользователями.

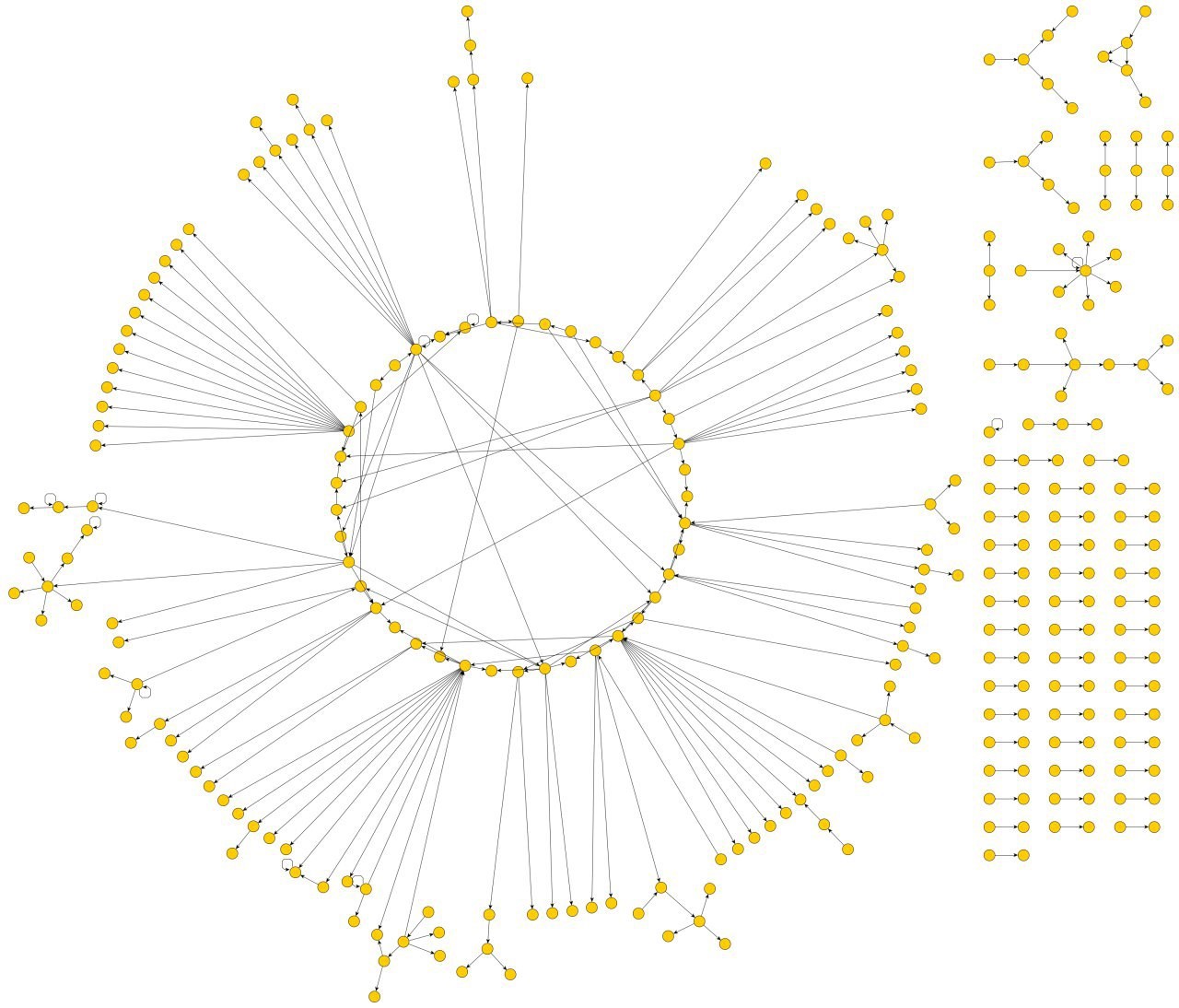


Граф пользователей, полученный с помощью применения силового алгоритма в yEd.

Здесь ребром соединяются пользователи, контактировавшие друг с другом хотя бы 1 раз.



Данный граф получается при усилении условия на ребра (теперь ребром соединяются только пользователи, контактировавшие друг с другом не менее 5 раз) и дополнительно отброшены изолированные вершины.



Вывод:

С помощью использованных инструментов (python для сбора данных и yEd для визуализации). Полученные визуализации позволяют сделать выводы о структуре сообщества. В дальнейшем можно применять другие фильтры и другие алгоритмы отрисовки графов.