Федеральное государственное образовательное бюджетное учреждение
высшего образования
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»**

Факультет информационных технологий и анализа больших данных
Кафедра анализа данных и машинного обучения

Выпускная квалификационная работа

на тему: Построение модели определения отраслевой принадлежности компании на основе ее медийной активности

Направление подготовки: 01.03.02 Прикладная математика и информатика
Профиль: Анализ данных и принятие решений в экономике и финансах

Выполнил студент учебной группы
ПМ20-1
Кудряшов Никита Александрович _____

Научный руководитель работы
ассистент
Блохин Никита Владимирович _____

ВКР соответствует предъявляемым требованиям:
Заведующий кафедрой анализа данных и машинного обучения, к.т.н., доцент

_____ Д. А. Петросов

«____» _____ 2024 г.

Москва 2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Стремительное развитие информационных технологий и доступность информации сильно изменил модель ведения бизнеса. Компании активно используют различные медийные платформы для распространения информации о своей деятельности, достижениях, продуктах и услугах. Социальные сети, новостные сайты, блоги, форумы и другие онлайн-ресурсы стали неотъемлемой частью корпоративной коммуникации. Отображаемая там информация представляет компанию для конечного потребителя и возможных партнеров. Медийные платформы и хранящаяся там информация о компаниях выступают важным источником данных для аналитики и принятия стратегических решений внутри компании.

В условиях большого количества информации и конкурирующего рынка становится необходимым разрабатывать эффективные методы анализа, позволяющие определить отраслевую принадлежность компании. Понимание, к какой отрасли можно отнести компанию, дает возможность не только определить ее конкурентное преимущество и позиционирование на рынке, но и принимать обоснованные стратегические решения.

Анализ медийной активности компании становится ключевым фактором для выявления тенденций в поведении потребителей, оценки влияния маркетинговых кампаний, а также для прогнозирования ее развития в контексте отраслевой динамики. Разработка модели определения отраслевой принадлежности компаний на основе медийной активности является важным шагом к повышению качества аналитики и принятия обоснованных управленческих решений в условиях развития цифровой инфраструктуры. Актуальность работы заключается в том, что определение отраслевой принадлежности компании является нетривиальной задачей, которая требует привлечения современных методов и технологий: от сбора и формирования представления новостных данных до обучения глубокой нейронной сети.

Целью данной выпускной квалификационной работы является построение графой нейронной модели способной определить отраслевую принадлежность компании на основе ее медийной активности.

Основные задачи для достижения поставленной цели:

- Исследование основных источников медийной активности различных компаний.
- Составление собственного датасета с информацией о медийной активности компаний.
- Анализ существующих инструментов для обучения моделей на основе графов.
- Разработка алгоритма, учитывающего взаимосвязи между компаниями в одной отрасли.
- Техническая реализация и совершенствование сформулированных идей.
- Анализ полученных результатов.

Объектами исследования являются источники медийной активности компании, а также способы обучения графовой нейронной модели.

Предмет исследования – построение и дальнейшее применение модели классификации отраслевой принадлежности компании.

Методология работы опирается на классические и современные подходы в задачах обработки и анализа связанных данных, представимых в виде графов. Для реализации задач на практике был использован язык python 3.10.

Выпускная квалификационная работа состоит из трёх основных глав, в каждой из которых содержатся параграфы.

В первой главе описывается решаемая задача. Вводятся основные теоретические понятия, такие как граф и графовые нейронные сети.

Рассматриваются способы применения графов для реализации различных алгоритмов. Проводиться исследование метода по достижению поставленной цели выпускной квалификационной работы – предсказание связей в графах. Описываются различные математические подходы и подходы с применением глубокого обучения.

Во второй главе описываются основные идеи, особенности используемых данных, а также математический и алгоритмический аппарат для разработки алгоритма, решающего поставленную задачу.

Третья глава содержит описание данных, на которых будет происходить обучение модели. В ней также будут описаны основные этапы разработки. Глава заканчивается описанием результатов экспериментов и оценкой качества полученной модели.

ГЛАВА 1. ГРАФЫ И ГРАФОВЫЕ НЕЙРОННЫЕ СЕТИ. РАССМОТРЕНИЕ ЗАДАЧИ ПРЕДСКАЗАНИЕ СВЯЗЕЙ В ГРАФАХ

1.1 Граф, как структура данных

Граф – это математическая структура, состоящая из двух множеств: множества объектов или узлов ($V = \{v_1, \dots, v_2\}$) и множества связей или ребер между этими объектами ($E = \left(e_{ij}\right)^n, j = 1$) (1.1).

$$G = (V, E), V \in \{v_i | i \in N\}, E \subseteq \left\{\left(v_i, v_j\right) | \left(v_j, v_i\right) \in V^2\right\} \quad (1.1)$$

Графы являются одной из важнейших структур в математических и компьютерных науках, так как они лежат в основе многих процессов, моделей и алгоритмов [1]. При помощи графов возможно моделировать множество различных процессов: карты прокладки труб или дорог, социальные или компьютерные сети, такие как Интернет, сети компаний и финансовых операций, цепочки химических реакций и молекулярные взаимодействия, эпидемическое распространение болезней и многое другое.

Графы можно разделить на различные типы:

- Направленный – связи между вершинами в таком графе ориентированы и имеют определенное направление (1.2).

$$E \subseteq \left\{\left(v_i, v_j\right) | \left(v_i, v_j\right) \in V^2, \left(v_i, v_j\right) \neq \left(v_i, v_j\right), i \neq j\right\} \quad (1.2)$$

- Ненаправленный – связи между вершинами в таком графе не ориентированы и характеризуют отношение между узлами в обе стороны (1.3).

$$E \subseteq \left\{\left(v_i, v_j\right) | \left(v_i, v_j\right) \in V^2, \left(v_i, v_j\right) = \left(v_i, v_j\right), i \neq j\right\} \quad (1.3)$$

- Взвешенный – каждая из связей в таком графе имеет некоторый вес, который может быть выражен в различных характеристиках (1.4).

$$E \subseteq \left\{\left(v_i, v_j\right) | \left(v_i, v_j\right) \in V^2, W\left(v_i, v_j\right) \in R\right\} \quad (1.4)$$

Помимо этого, графы можно представить в виде матрицы смежности $A \in$

$\{0,1\}^{N\times N}$, в которых $A_{ij} = 1$, если вершины i и j соединены и $A_{ij} = 0$ в ином случае.

1.2 Графовые нейронные сети (GNN): особенности и области применения

Использование графов в качестве структуры данных получило широкое распространение в области машинного и глубокого обучения благодаря возможности графов эффективно моделировать отношения и зависимости между наборами объектов. Подобная структура позволяет более детально представить сложные нелинейные взаимосвязи по сравнению с традиционными табличными данными, которые предполагают линейную взаимосвязь между данными.

Впервые использование графовых нейронных сетей было предложено в работе [2]. Графовые нейронные сети или Graph Neural Network (GNN) представляет собой класс моделей глубокого обучения, взаимодействующих с данными, представленными в виде графов.

Представление вершин, связей или в целом всего графа в виде эмбеддингов [3] позволяет облегчить работу графовых нейронных сетей. Эмбеддинг – есть математическая структура, которая находится внутри другой структуры. Когда говорится, что некоторый объект X является эмбеддингом объекта Y, то эмбеддинг выступает в виде инъективного и сохраняющего структуру отображения. Иными словами, эмбеддинг - непрерывное отображение некоторого объекта в вектор в пространство другой размерности. Векторы, в которые преобразуется структура графа, позволяют сохранить топологию и семантику исходной сети. Здесь к топологии относится изначальный вид графа: типы или шаблоны связей, наличие кластеров и общее взаимное расположение узлов и ребер.

Базовая структура GNN включает в себя энкодер и декодер (рис. 1) [4]. Энкодер принимает на вход изначальный граф, генерирует и представляет эмбеддинги. Декодер в свою очередь обрабатывает и использует созданные

эмбеддинги для составления прогнозов. Принцип кодирования информации с одинаковым вектором, основанным на сходстве окрестностей в графе, является основополагающим для сбора структурных данных о графе. Благодаря такому подходу GNN применяет итеративные механизмы агрегации и распространения информации по сети, позволяя обучать узлы на основе их локального окружения. Конечные обученные эмбеддинги инкапсулируют информацию о структуре и связности графа, которую в дальнейшем можно использовать для решение привычных задач машинного обучения.

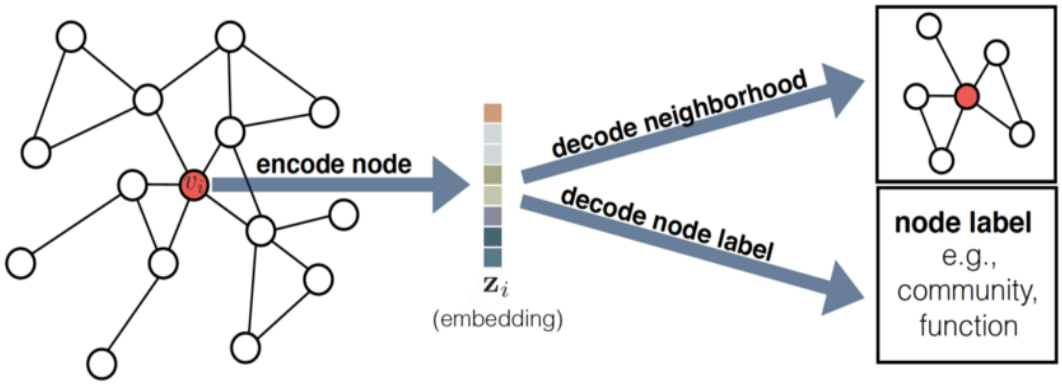


Рис. 1. Пример подхода к обработке графа.

Графовые нейронные сети – универсальные модели, которые могут выполнять различные задачи с использованием данных, структурированных в виде графа. Среди самых распространенных задач можно выделить следующие:

- Классификация узлов (Node Classification) - определение меток классов для каждого узла в графе на основе характеристик узла и топологии графа. Подобные модели могут быть использованы для определения категорий узлов в социальной сети [5], сети цитирований или фармакологических сетей [4].
- Предсказания связей в графах (Link Prediction) – задача заключается в прогнозировании установления связей между двумя узлами графа,

которые в данный момент не имеют прямой связи (матрица смежности для них не завершена). Таким образом, подобные модели могут быть использованы в качестве рекомендательных систем, например, при подборе фильмов в онлайн кинотеатрах [15, 16], прогнозировании потенциальных взаимосвязей в социальных сетях на основе общих интересов [6], прогнозирование соавторства в сетях цитирования [17] и многом другом.

- Классификация графов (Graph Classification) — это классификация целых сетей по предопределенным классам. К примерам решаемых задач можно отнести классификацию химических соединений на основе молекулярной структуры или лекарственных препаратов в целом [8], определение типов графов в социальных сетях, оценка безопасности графов на основе их общего статуса безопасности (на примере прогнозирования безопасности авиаперелетов на основе графов предыдущих полетов в определенных областях) [9], а также использование средств обработки естественного языка и графовых нейронных сетей для классификации текстов [10].
- Обнаружение сообществ (Community Detection) - идентификация групп узлов, плотно связанных внутри графа. Подобный подход может быть использован в обнаружении групп пользователей со схожими интересами в социальных сетях [11] или определения наиболее загруженных транспортных районов [12].
- Построение векторных представлений для графовых наборов данных (Graph Embedding) - создание и анализ низкоразмерных представлений для узлов, ребер или целых графов для сохранения информации о их структуре и топологии.
- Генерация графов (Graph Generation). Создание новых графов, которые имеют сходные структурные свойства с уже имеющимся набором данных, позволяет расширить изначальную выборку и тем самым

улучшить качество создаваемых моделей. Подобный подход может использоваться в множестве задач, упомянутых ранее. В качестве примера реализации подобного подхода стоит упомянуть графовую генеративную модель, целью которой является изучение эффективных взаимосвязей узлов графа в end-to-end режиме для решения сложных задач генерации архитектурных макетов с ограниченными графическими возможностями (Graph Transformer Generative Adversarial Network (GTGAN)) [13].

Из представленных подходов применения GNN подходящей для решения поставленных задач в рамках темы выпускной квалификационной работы является использование модели предсказаний связей в графах (Link Prediction).

1.3 Постановка решаемой задачи

Для достижения целей данной выпускной квалификационной работы необходимо построить модель, способную учитывать медийные взаимосвязи между компаниями, которые позволят провести точное и объективное определение отраслевой принадлежности. В таком случае различные организации и связи могут быть представлены в виде графа, где узлами выступают сами компании, а ребрами обозначается их медийная связь. Использование графовой структуры позволяет обобщать информацию о взаимоотношениях компаний и использовать ее для обучения модели.

Для построения рекомендательной системы на основе медийной активности компаний необходимо обучить модель с точностью предсказывать их отраслевую принадлежность. Организациям необходимо знать конкретную область ведения деятельности для построения успешного и прибыльного бизнеса. Используя графовую структуру, помимо создания узлов и связей между компаниями можно создавать вершины, представляющие собой уникальные идентификаторы отраслевой деятельности, связи между

которыми будет определяться на основе принадлежности к более крупным отраслевым группам.

Задача сводиться к тому, чтобы обучить модель находить взаимосвязи между компаниями и публикуемыми о них новостями и соединять их с идентификаторами отраслевой деятельности. Для решения этой проблемы из описанных в прошлом пункте задач, решаемых с помощью графовых нейронных сетей, наилучшим образом подходит предсказание связей в графах (Link Prediction).

1.4 Предсказания связей в графах (Link Prediction). Гетерогенность и гомофильность графов.

Как уже говорилось ранее, предсказание связей в графах – это задача прогнозирования существования связи между двумя узлами сети. Впервые эта задача была сформулирована в работе [14].

Link Prediction имеет множество названий в зависимости от области применения. Термин «предсказание связей» часто относится к предсказанию связей в однородных графах (homogeneous graphs), где узлы и связи между ними имеют только один тип. Примерами таких сетей можно назвать строение атома или граф остатков (рис. 2)

Сложнее бывает, когда сеть состоит из узлов, представляющих отдельные объекты с различными типами связей, соответствующих различным отношениям между объектами. Такие графы называются графами знаний. Иначе их можно назвать гетерогенными графами (heterogeneous graphs). В качестве примера можно привести сети регуляции генов или химических элементов (рис. 2) Различные вершины и связи в таких графах обладают собственными пространствами идентификаторов и набором характеристик (рис. 3), в отличие от однородных графов.

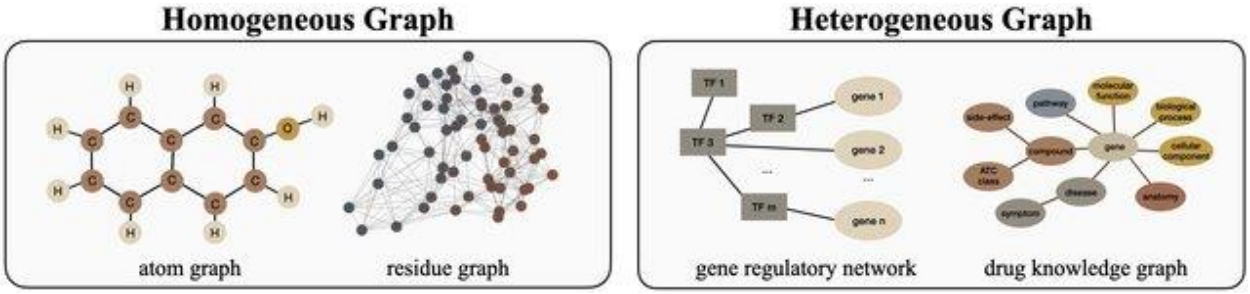


Рис 2. Примеры однородных и гетерогенных графов.

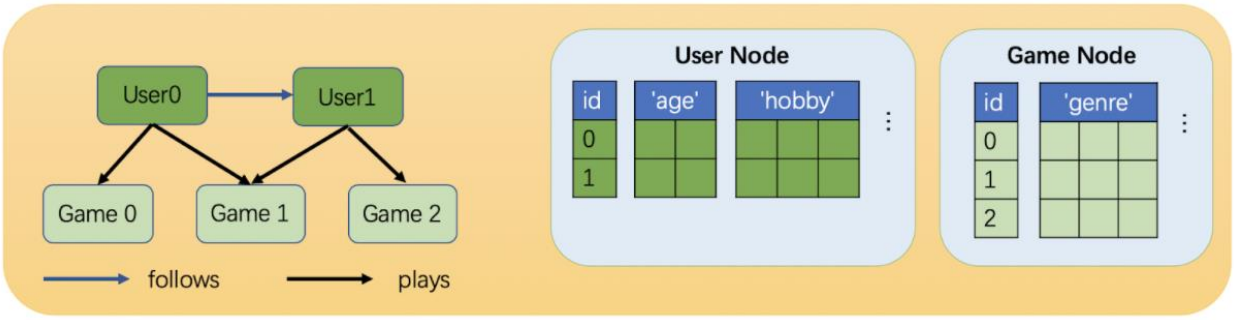


Рис. 3. Пример гетерогенного графа с двумя типами узлов («пользователь» и «игра») и двумя типами связей («подписан» и «играет»).

1.5 Классические методы предсказаний связей в графах

При рассмотрении различных методов решения задачи Link Prediction можно выделить два основных типа: классические методы и методы с использованием графовых нейронных сетей. Первые, в свою очередь, подразделяются на:

- Эвристические методы (Heuristic Methods) – методы используют простые, но эффективные оценки сходства узлов в качестве вероятности наличия связей [19].
- Методы изучения скрытых признаков (Latent-Feature Methods) в некоторой литературе данные методы также называются моделями скрытых факторов (latent-factor models) или эмбеддинговыми методами (embedding methods). Методы вычисляют скрытые свойства или представления узлов, часто получаемые путем разложения на множители определенной матрицы, полученной из сети, таких как

матрица смежности или матрица Лапласа. Эти скрытые характеристики узлов не являются явно наблюдаемыми — они вычислятся путем оптимизации.

- Методы, основанные на содержании (Content-Based Methods), используют имеющиеся характеристики данных, связанные с узлами.

1.5.1 Эвристические методы

Рассмотрим некоторые примеры эвристических методов предсказывания связей. Обозначим x и y в качестве исходного и целевого узла, между которыми можно спрогнозировать связь. Для обозначения множества соседей узла x будем использовать $\Gamma(x)$.

Первым и самым простым способом является метод общих соседей (common neighbors (CN)). Данный подход рассчитывает количество вершин, одновременно являющихся соседями двух узлов в качестве вероятности наличия между ними связи (1.5) (рис. 4).

$$f_{CN}(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{1.5}$$

CN широко используется в рекомендациях друзей в социальных сетях. Предполагается, что чем больше у двух людей общих друзей, тем больше вероятность, что они сами также являются друзьями.

Коэффициент Жаккара (Jaccard score) измеряет долю общих соседей (1.6).

$$f_{Jaccard}(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{1.6}$$

Метод предпочтительной привязанности (preferential attachment (PA)) использует произведение степеней узла[1] для измерения вероятности связи

[1] Степень узла – количество ребер графа, инцидентных в конкретной вершине.

(1.7).

$$f_{PA}(x,y) = |\Gamma(x)| \cdot |\Gamma(y)| \quad (1.7)$$

PA подразумевает, что x с большей вероятностью будет связан с y, если y имеет высокую степень (рис. 4). Подобный подход применим в сетях цитирования - новая статья с большей вероятностью будет цитировать те статьи, которые уже имеют много упоминаний. Сети, сформированные с помощью механизма PA, называются свободными от масштабирования (scalefree networks) [20].

Существующие эвристические методы могут быть классифицированы на основе максимального количества переходов между соседними узлами, необходимого для вычисления оценки. CN, Jaccard и PA считаются методами первого порядка, так как они используют только самых ближайших соседей (с одним переходом для двух целевых узлов). Далее рассмотрим два метода второго порядка.

Индекс Адамик-Адар (Adamic-Adar (AA)) [21] – метод, учитывающий веса общих соседей (1.8).

$$f_{AA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|} \quad (1.8)$$

Он определяется как сумма обратной логарифмической степени центральности соседей, разделяемых двумя узлами. Предполагается, что вершина с высокой степенью соединяется как с x, так и с y менее информативна, чем узел более низкой степенью (рис. 4).

Метод распределения ресурсов (Resource allocation (RA)) [22] использует более жесткий фактор снижения веса (1.9).

$$f_{RA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|} \quad (1.9)$$

Он определяется как сумма обратной степени центральности соседей, разделяемых двумя узлами. Предполагается, что в отличие от подхода Адамик и Адара, метод RA еще больше будет отдавать предпочтения узлам с низкими степенями.

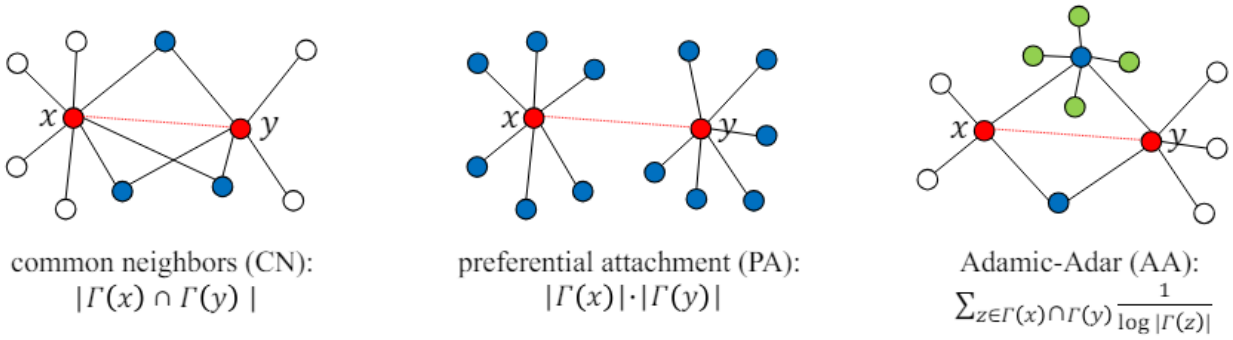


Рис. 4. Иллюстрация трех эвристических подходов в решении задачи предсказания связей: CN, PA и AA.

Способы первого и второго порядка являются локальными методами, поскольку все они высчитываться из локального подграфа ближайших соседей исходного и целевого узла, не учитывая при этом структуру всей сети. Существуют также методы, которые рассматривают всю сеть целиком. Такие подходы называться эвристическими методами высокого порядка (high-order heuristics) и, как правило, являются более производительными, в отличие он низкоуровневых методов. К ним можно отнести индекс Каца (Katz index) [23], Rooted PageRank (RPR) [24] и SimRank (SR) [25].

Индекс Каца использует взвешенную сумму всех переходов между узлами x и y, где более продолжительный переход оценивается ниже (1.10).

$$f_{Katz}(x,y) = \sum_{l=1}^{\infty} \beta^l \left| walks^{\langle l \rangle}(x,y) \right| \qquad (1.10)$$

Здесь β является убывающим коэффициентом между 0 и 1, а $\left| walks^{\langle l \rangle}(x,y) \right|$ подсчитывает l-длину переходов между x и y. Если же рассматривать переходы только длинной 2 ($l = 2$), то индекс Каца сводиться к методу CN.

Rooted PageRank является обобщением осинового алгоритма PageRank[2]. Сначала рассчитывается стационарное распределение π_x случайного блуждания[3], начиная с узла x, который двигается по текущим соседям с вероятностью α или возвращается в вершину x с вероятностью $1-\alpha$. Далее π_x рассчитывается от узла y (обозначается как $[\pi_x]_y$), чтобы предсказать связь (x, y). Если же граф неориентированный, алгоритм симметрично повторяется относительно узла y (1.11).

$$f_{RPR}(x, y) = [\pi_x]_y + \left[\pi_y\right]_x \tag{1.11}$$

Оценка SimRank предполагает, что два узла схожи, если их соседи также схожи. Она рассчитывается рекурсивно (1.12):

$$\begin{aligned} &if\ x = y: && f_{SR}(x, y) = 1 \\ &otherwise: && f_{SR}(x, y) = \gamma \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(x)} f_{SR}(a, b)}{|\Gamma(x)| \cdot |\Gamma(y)|} \end{aligned} \tag{1.12}$$

где γ – константное значение между 0 и 1.

Помимо описанных выше эвристических методов существуют множество других подходов, описанных в работах [14, 26]. Эвристические методы можно рассматривать как вычисление предопределенных особенностей графовой структуры относительно расположения узлов и связей между ними. Несмотря на эффективность во многих областях, эти характеристики графовой структуры охватывают лишь небольшое

---

[2] PageRank - один из алгоритмов ссылочного ранжирования. Алгоритм применяется к коллекции документов, связанных гиперссылками (таких, как веб-страницы из всемирной паутины), и назначает каждому из них некоторое численное значение, измеряющее его «важность» или «авторитетность» среди остальных документов.

[3] Случайное блуждение (random walk) - случайный процесс перехода между состояниями (в нашем случае - узлами сети / вершинами графа) определяемый матрицей перехода (row stochastic matrix), в которой фиксируется вероятность случайного перехода из узла в узел.

подмножество всех возможных структурных шаблонов и не могут в полном объеме отобразить характеристики сети. Большинство эвристических методов работают только с однородными графами. Кроме того, эвристические методы хорошо работают только тогда, когда механизм формирования сети согласуется с эвристикой. Могут существовать сети со сложными механизмами формирования, на которых не получиться использовать ни один из методов.

1.5.2 Методы, основанные на скрытых признаках

Для того, чтобы в полной мере учесть все структурные особенности сети следует рассмотреть методы, учитывающие скрытые признаки графов (Latent-Feature Methods). Среди такого подхода к решению задачи предсказания связей можно выделить два основных метода: матричная факторизация (Matrix Factorization) и вычисление эмбеддингов графов (Network Embedding).

Матричная факторизация (MF) была рассмотрена в работах по рекомендательным системам [27, 28]. MF преобразует разряженную матрицу смежности A графа в произведение двух плотных матриц эмбеддиннгов Z. Исходные признаки матрицы (в нашем случае связи в графах) выражаются через латентные признаки линейным образом (1.13).

$$\hat{A}_{i,j} = z_i^T \cdot z_j \tag{1.13}$$

Затем происходит минимизация среднеквадратичной ошибки между создаваемой матрицей $\hat{A}_{i,j}$ и обычной матрицей смежности A по рассматриваемым связям для того, чтобы определить эмбеддинги (1.14).

$$\mathcal{L} = \frac{1}{\mathcal{E}} \sum_{(i,j)\in\mathcal{E}} \left(A_{i,j} - \hat{A}_{i,j}\right)^2 \tag{1.14}$$

Таким образом мы можем предсказать новые связи по скалярному произведению двух эмбеддингов интересующих узлов. Вариации матричной факторизации включают использование степеней в матрице A [29] или

использовании матриц сходства узлов (Node similarity matrices) [30] для замены обычной матрицы смежности. Если мы заменим A на матрицу Лапласа L и определим потери следующим образом (1.15):

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{E}} \left\| z_i - z_j \right\|_2^2, \tag{1.15}$$

Затем строится нетривиальное решение вышеприведенной задачи с использованием собственных векторов, соответствующих k наименьшим ненулевым собственным значениям L, которые восстанавливают метод лапласовой карты собственных значений (Laplacian eigenmap technique) [31] и решение для спектральной кластеризации (spectral clustering) [32].

Применения сетевых эмбеддингов (Network embedding) получили большую популярность в последнее время благодаря работе DeepWalk [33]. Подобные методы изучают низкоразмерные представления (эмбеддинги) для узлов, как правило основанные на обученной модели скип-грам[4] (skipgram model) [34] на основе последовательностей узлов, генерируемых случайным блужданием, так что узлы, которые часто появляются рядом друг с другом при случайном блуждании (т. е. узлы, расположенные близко в сети), будут иметь схожие эмбеддинги. Затем эти представления попарно объединяются для прогнозирования связей. В работе [35] также упоминается, что многие другие методы Network Embeddings (такие как LINE, DeepWalk и node2vec).

Таким образом, они также могут быть отнесены к методам со скрытыми признаками. Например, в DeepWalk приблизительно факторизует (1.16):

[4] Изначально скип-грам (вариант модели word2vec) - один из методов обучения без учителя, который используется для поиска близких по тематике слов для заданного слова.

$$log\left(vol(G)\left(\frac{1}{w}\sum_{r=1}^{w}(D^{-1}A)^{r}\right)D^{-1}\right)-\log(b), \quad (1.16)$$

где:

- $vol(G)$ – сумма степеней узлов;
- D – диагональная степенная матрица;
- w – размерность окна скип-грам;
- b – некоторая константа.

Как мы можем видеть, DeepWalk, по сути, факторизует логарифм суммы некоторых нормализованных матриц смежности высокого порядка (вплоть до w). Это можно представить как случайное блуждание с расширением окрестности на w шагов, так что мы не только требуем, чтобы прямые соседи имели похожие эмбеддинги, но и требуем, чтобы узлы, находящиеся друг от друга через w шагов случайного блуждания, имели похожие вложения.

Аналогично, алгоритм LINE а своих формах второго порядка неявно факторизует (1.17):

$$log\big(vol(G)(D^{-1}AD^{-1})\big)-log(b). \quad (1.17)$$

Другой популярный метод node2vec, который является улучшенной версией DeepWalk с добавлением негативного семплирования[5] и смещенным случайным блужданием[6], также неявно факторизует матрицу.

Главным недостатком метода поиска скрытых признаков является

---

[5] Негативное сэмплирование — это способ создать для обучения векторной модели отрицательные примеры, то есть показать ей пары узлов, которые не являются соседями по контексту.

[6] Смещенное случайное блуждание (biased random walk) – отличается от обычного случайного блуждания тем, что переменная с некоторой непостоянной вероятностью в некоторый момент может сменить свое текущее состояние на любое другое потенциальное состояние.

отсутствие возможности выявлять структурное сходство между узлами, так же как и схожие модели в NLP не могут отражать контекстное значения некоторых слов. Методы со скрытыми признаками требуют чрезвычайно большой размерности для выражения некоторых простых эвристик, что иногда приводит к худшей производительности, чем эвристические методы. Наконец, методы с использованием скрытых признаков являются трансдуктивными методами обучения - эмбеддинги узлы не могут быть обобщены на новые узлы или сети.

Несмотря на то, что в большинстве случаев методы направлены на однородные сети, существует множество методов с использованием скрытых признаков, разработанных для гетерогенных графов. Например, модель RESCAL [36] обобщает матричную факторизацию на графы с множественными типами связей, которая, по сути, выполняют своего рода тензорную факторизацию. Metapath2vec [37] обобщает node2vec на гетерогенные графы.

1.5.2 Методы, основанные на содержании узлов

Как эвристические методы, так и методы со скрытыми признаками сталкиваются с проблемой "холодного запуска" (cold-start problem). То есть, когда к сети присоединяется новый узел, эвристические методы и методы со скрытыми признаками могут быть не в состоянии точно предсказать его связи, поскольку у него мало существующих связей или их нет. В этом случае могут помочь методы, основанные на содержании (content-based methods) [38]. Такие методы используют явные характеристики узла (данные хранящиеся в нем). Например, в сетях цитирования распределение слов в тесте может использоваться в качестве подходящих элементов данных узлов. В социальных сетях профиль пользователя, с информацией о человеке и его интересами, может использоваться в качестве элементов данных (однако информация о дружбе относится к элементам графической структуры).

Content-based методы, как правило менее эффективны, в отличие от эвристических методов и методов со скрытыми признаками, вследствие чего их часто используются совместно с для повышения общей производительности прогнозирования связей [39, 40].

1.6 Методы предсказаний связей в графах на основе моделей глубокого обучения на графах

В отличие от классических, методы GNN объединяют в себе использование и графической информации, и содержимого узлов одновременно для создания модели предсказания связей. Выделяют два типа подходов реализации подобных алгоритмов:

- Методы, основанные на узлах (Node-Based Methods), - используют представления о попарной связей узлов для построения модели.
- Методы, основанные на подграфах (Subgraph-Based Methods), - извлекают локальный подграф вокруг каждой связи и используют представление подграфа, полученное с помощью GNN, в качестве представления связей.

1.6.1 Методы, основанные на эмбеддингах узлов

Самый простой способ использования GNN для прогнозирования связей — это рассматривать GNN как индуктивные методы сетевых эмбеддингов, которые изучают эмбеддинги узлов из локальной окрестности, затем попарно объединяя их для построения представлений связей внутри графа (Node-Based Methods).

Первой работой, посвященной методам на узлах, была статья о Graph AutoEncoder (GAE) [41]. Для заданной матрицы смежности A и матрицы фичей узла X графа, GAE сначала использует модель GCN[7] [42] для расчета

[7] Графовая сверточная сеть (Graph Convolutional Network (GCN)). Архитектура

представлений z_i для каждого i-ого узла, затем применяя $\sigma(z_j, z_i)$ для предсказания связи (i, j) (1.18):

$$\hat{A}_{i,j} = \sigma(z_i, z_j), \qquad where\ z_i = Z_i; Z = GCN(X, A), \tag{1.18}$$

где:

- Z – матричное представление узла, являющееся результатом обработки GCN, в которой i-ая строка является представлением z_i i-ого узла;
- $\hat{A}_{i,j}$ – предсказываемая вероятность для связи (i, j);
- σ – сигмоидная функция.

Если X не задано, то GAE использует one-hot encoding матрицу I. Модель минимизирует кросс-энтропию между восстановленной и истинной матрицей смежности (1.19).

$$Loss = \sum_{i \in V, j \in V} \left(-A_{i,j} log \hat{A}_{i,j} - (1 - A_{i,j}) \log (1 - \hat{A}_{i,j})\right) \tag{1.19}$$

На практике функция потерь для положительных связей ($A_{i,j} = 1$) увеличивается на величину k, являющаяся соотношением между негативными связями ($A_{i,j} = 0$) и позитивными. Цель состоит в том, чтобы сбалансировать вклад положительных и отрицательных ребер в потери. В противном случае высокое значение функции потерь связано с отрицательными связями из-за общей разреженности используемого графа.

Вариантной версией модели GAE является модель VGAE (Variational Graph AutoEncoder) [41]. Вместо того, чтобы использовать детерминированные эмбеддинги узлов z_i, VGAE реализует сразу две модели GCN для расчета среднего μ_i и дисперсии σ_i^2 для z_i отдельно. VGAE предполагает, что матрица смежности A генерируется из скрытых представлений узлов Z через $p(A|Z)$, где Z следует априорному

распределению[8] $p(Z)$. Для предсказания связей используется модель на основе матричного произведения в виде $p(A|Z)$ (1.20):

$$p(A|Z) = \prod_{i \in V} \prod_{j \in V} p\left(A_{i,j}|z_i, z_j\right), \ \ where \ p\left(A_{i,j} = 1|z_i, z_j\right) = \sigma\left(z_i, z_j\right). \quad (1.20)$$

В свою очередь, предварительное распределение $p(Z)$ принимает стандартное нормальное распределение (1.21):

$$p(Z) = \prod_{i \in V} p(z_i) = \prod_{i \in V} N(z_i|0, I)\,. \quad (1.21)$$

Учитывая $p(A|Z)$ и $p(Z)$, мы можем вычислить апостериорное распределение[9] Z, используя теорему Байеса. Однако это распределение часто оказывается неразрешимым. Таким образом, учитывая матрицу смежности A и матрицу характеристик узлов X, VGAE использует графовые нейронные сети для аппроксимации апостериорного распределения матрицы эмбеддингов узлов Z (1.22):

$$q(Z|X, A) = \prod_{i \in V} q(z_i|X, A), \ \ where \ q(z_i|X, A) = N\left(z_i|\mu_i, diag\left(\sigma_i^2\right)\right). \quad (1.22)$$

Затем VGAE максимизирует нижнюю границу данных для изучения параметров GCN (1.23):

$$Loss = \mathbb{E}_{q(Z|X,A)}[\log p(A|Z)] - KL[q(Z|X, A)||p(Z)], \quad (1.22)$$

где:

- $KL[q(Z|X, A)||p(Z)]$ – это расстояние Кульбака-Лейблера между

[8] Априорное распределение (prior distribution) - распределение вероятностей, которое выражает предположения о p до учёта экспериментальных данных.

[9] Апостериорное распределение (posterior distribution) - распределение, вычисленное в результате проведения эксперимента с исследуемыми объектами.

аппроксимированным апостериорным и априорным распределением Z.

Оптимизация происходит за счет репараметризации (reparameterization trick) [43]. Суть такого метода заключается в следующем – пусть z является непрерывной случайной величиной и $z \sim g_\phi(z|x)$ – некоторое условное распределение. Тогда эту СВ можно выразить как детерминированную переменную $z \sim g_\phi(\epsilon|x)$, где ϵ – это вспомогательная переменная с независимым предельным значением $p(\epsilon)$ и $g_\phi(.)$ – некоторая векторнозначная (vector-valued) функция, параметризованная через ϕ. Наконец эмбеддинги средних μ_i и μ_j используются для прогнозирования связи (i, j) через $\hat{A}_{i,j} = \sigma(\mu_i^T, \mu_j)$.

На основе двух этих подходов к решению задачи предсказания связей были разработаны множество других алгоритмов, которые сегодня используется во множестве разрабатываемых GNN моделей (PGNN, GC-MC, R-GCN, SACN и другие).

1.6.2 Методы, основанные на эмбеддингах подграфов

Методы, основанные на подграфах, извлекают локальный подграф вокруг каждой целевой связи и изучают представление подграфа с помощью GNN для прогнозирования ссылок.

Основоположником таких методов был SEAL (learning from Subgraphs, Embeddings, and Attributes for Link prediction) [44]. SEAL сначала извлекает окружающий подграф для каждой целевой связи для прогнозирования, а затем применяет GNN на уровне графа (с пуллингом), чтобы определить, соответствует ли подграф существованию соединения.

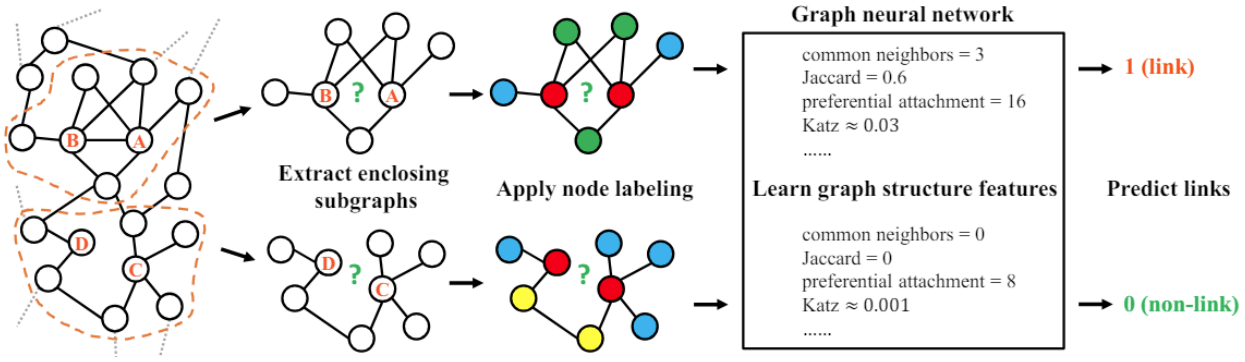


Рис. 5. Иллюстрация фреймворка SEAL.

Для начала необходимо дать определение понятию вложенного графа (enclosing subgraph). Для графа $G = (V, E)$ при заданном наборе узлов $S \subseteq V$, вложенный подграф, охватывающий h-переход для S, есть подграф G_S^h, индуцированный из G набором узлов $\cup_{j \in S} \{i | d(i,j) \leq h\}$, где $d(i,j)$ – кратчайшее расстояние между вершинами i и j. Иными словами, вложенный подграф, охватывающий h-переход вокруг набора узлов S, содержит узлы в пределах h-переходов любого узла в S, а также все ребра между этими узлами. В задачах предсказания связей набор узлов S обозначает два узла, между которыми необходимо предсказать связь. Например, при прогнозировании связи между x и y, $S = \{x, y\}$ и $G_{x,y}^h$ обозначает подграф, охватывающий h переходов для связи (x, y). Цель извлечения вложенного подграфа для каждой связи заключается в том, что SEAL стремится автоматически изучать особенности структуры графа из сети. Используя GNN, алгоритм пытается сформировать представление о полноценном графе на основе характеристик вложенных подграфов с h переходами.

После извлечения вложенного подграфа $G_{x,y}^h$ следующим шагом является маркировка узлов (node labeling). SEAL применяет маркировку узлов с двойным радиусом (Double Radius Node Labeling (DRNL)), чтобы присвоить целочисленную метку каждому узлу в подграфе в качестве дополнительной фичи. Цель состоит в том, чтобы использовать разные метки для различения

узлов с разными ролями во вложенном подграфе. Например, узлы x и y, между которыми мы хотим определить связь, должны отличаться от остальных и иметь одинаковое обозначение. Аналогично, узлы на разных переходах относительно x и y должны иметь собственную метку, чтобы мы могли их явно идентифицировать. Правильная маркировка узлов позволяет методам, основанным на подграфах, лучше усваивать представление связей, чем методам, основанным на узлах.

После получение меток узлов SEAL преобразует их в эмбеддинги. Эти новые векторы фичей объединяются с исходными фичами содержимого узла (если таковые имеются) для формирования новых объектов узла. SEAL также позволяет объединять некоторые предварительно обученные эмбеддинги узлов, такие как эмбеддинги node2vec, с фичами узлов. Однако, как показывают результаты экспериментов, добавление предварительно обученных эмбеддингов узлов не дает явных преимуществ для конечной производительности [44].

Наконец, SEAL загружает эти вложенные подграфы, а также их новые векторы фичей узлов в GNN на уровне графа (DGCNN [44]) для обучения классификатора. Чтобы обучить эту нейронную сеть, SEAL случайным образом выберет N существующих соединений из сети в качестве положительных обучающих связей и равное количество ненаблюдаемых связей (случайных пар узлов) в качестве отрицательных обучающих связей. После обучения SEAL применяет обученный GNN к вложенным подграфам новых ненаблюдаемых пар узлов, чтобы предсказать их связи. Вся структура SEAL показана на (рис. 5). SEAL обеспечивает высокую производительность при прогнозировании связей, демонстрируя более высокую производительность, чем при использовании предопределенных эвристик.

Подход для решения задачи предсказания связей, описанный в статье [44] и реализованный в SEAL, вдохновил множество последующих работ,

которые модернизировали отдельные этапы основного алгоритма.

1.6.3 Сравнительная характеристика двух подходов

На первый взгляд, как методы, основанные на узлах, так и методы, основанные на подграфах, изучают особенности структуры графа вокруг предполагаемых связей на основе GNN. Однако, Subgraph-Based методы на практике более эффективны.

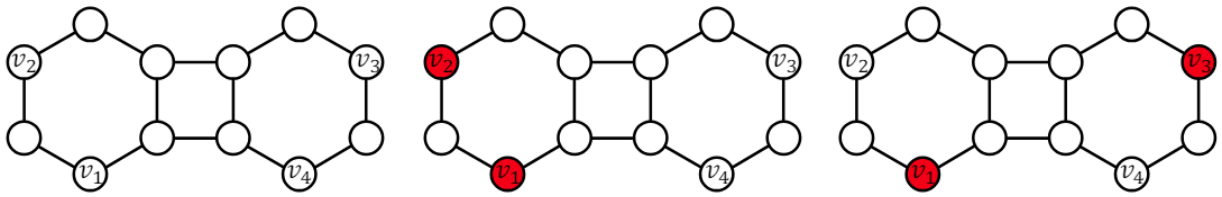


Рис. 6. Различная способность представления связей между методами, основанными на узлах и подграфах.

Для начала опишем на примере ситуацию, в которой явно можно увидеть превосходство моделей на основе подграфов. На (рис. 6) слева изображен пример графа, на котором мы хотим выполнить прогнозирование связей. Узлы v_2 и v_3 изоморфны (симметричны друг относительно друга), как и связи (v_1, v_2) и (v_4, v_3), в то время как (v_1, v_2) и (v_1, v_3) не изоморфны. Так узел v_1 близок к v_2 и делит с ним больше общих соседей, чем с узлом v_3, поэтому интуитивно мы не можем одинаково представлять узлы v_2 и v_3. Однако, так как v_2 и v_3 изоморфны, методы, основанные на узлах, будут обучены одинаковыми представлениями для этих узлов, соответственно вероятность связей (v_1, v_2) и (v_1, v_3) в таком случае будет одинакова, что некорректно.

С использованием node-based методов, GNN не может рассчитывать количество ближайших соседей между двумя узлами (что равно 1 для (v_1, v_2) и 0 для (v_1, v_3)), что является одной из фундаментальных структурных особенностей для прогнозирования связей.

Если же мы извлекаем вложенные подграфы с одним переходом как для (v_1, v_2), так и для (v_1, v_3), то они сразу же становятся различимыми из-за их различных структур охватывающих подграфов. Кроме того, этап маркировки узлов в методах, основанных на подграфах, также помогает моделировать связи между двумя целевыми узлами более эффективно.

1.7 Вывод

Для решения задачи предсказания связи в графах существует много различных алгоритмов и готовых инструментов. Применяя методы на практике с уникальным набором данных, необходимо учитывать все структурные особенности и характеристики графа, чтобы выбрать оптимальный вариант, который будет показывать достойный результат.

Цели второй и третьей главы заключаются в том, чтобы реализовать, применить и оценить результаты работы различных моделей прогнозирования связей для предсказания отраслевой принадлежности компаний на основе их медийной активности.

ГЛАВА 2. РАССМОТРЕНИЕ ИСПОЛЬЗУЕМЫХ ДАННЫХ И МОДЕЛЕЙ

2.1 Формирование датасета

Для составления графа и последующего обучения модели необходимо сформировать датасет, включающий в себя основную информацию о компании, а также данные о ее медийной активности. Помимо этого, необходимо ввести определенный идентификатор отраслевой принадлежности, который необходимо будет предсказывать. Таким образом, итоговый граф можно разделить на два отдельных подграфа:

- Граф медийной активности компаний.
- Граф идентификаторов отраслевой принадлежности компаний.

Первый граф будет представлять из себя сеть с узлами, являющимися компаниями и обладающими набором фичей, и соединениями, определяющими связь организаций на основании медийной активности. Второй граф будет представлять собой иерархию кодов ОКВЭД[10], структура которых представлена на (рис. 1). Граф будет иметь древовидную структуру, связи в котором будут обозначать «потомственность» каждой группы и подгруппы идентификаторов.

[10] ОКВЭД – Общероссийский классификатор видов экономической деятельности ОК 029-2014 (КДЕС Ред. 2), утвержденный приказом Росстандарта N 14-ст от 31.01.2014г. Представляет из себя

Рис. 1. Структура кода ОКВЭД

В результате должен получиться гетерогенный граф, определяющий различные связи между компаниями, компаниями и идентификаторами отраслевой принадлежности и потомками в кодах ОКВЭД.

2.1.1 Рассмотрение типов медийной информации

Для начала необходимо определить, какие аспекты медийной деятельности компании нам были бы интересны. Классифицировать их можно по следующим критериям:

- Финансовые новости: новости о финансовом состоянии компании, отчеты о прибылях и убытках, инвестиционные анонсы, выплаты дивидендов и т. д. Как упоминалось в статье [45], использование такой информации и построение на ее основе сети с последующим обучением графовой нейронной модели позволяет более точно выявить отраслевую принадлежность для групп компаний.
- Стратегические новости: Объявления о стратегических партнерствах, слияниях и поглощениях, запусках новых продуктов или услуг, развитие новых рынков и т. д.
- Кадровые новости: Информация о назначениях на ключевые должности,

об отставках или увольнениях топ-менеджмента, изменениях в руководстве компании и т. п.

- Технологические новости: о новых технологиях, патентах, разработках, инновациях, обновлениях продуктов и услуг компании, а также об участии компании в индустриальных событиях и конференциях.
- Сообщения о социальной ответственности: новости о деятельности компании в области социальной ответственности, участие в благотворительных программах и акциях, экологические проекты и другие инициативы.
- Кризисные новости: объявления об инцидентах, скандалах, конфликтах с заинтересованными сторонами, судебных процессах или других кризисных ситуациях, в которых оказалась компания.

Эти типы новостей могут варьироваться в зависимости от сферы деятельности компании, ее размера, стратегии взаимодействия со СМИ и других факторов.

2.1.2 Рассмотрение возможных источников информации

Следующей задачей было рассмотрение источников новостей, которые бы включали в себя различные типы медийной активности компаний, описанных в предыдущем разделе.

Были изучены следующие новостные сайты:

- «РБК Компании» [46] – сервис для управления репутацией компаний и экспертов в деловом СМИ, а также PR-инструмент для создания публикаций, новостей бизнеса, брендов и корпораций. Организации используют данный сервис для продвижения своих продуктов и отслеживания заинтересованности пользователей.
- «РБК Отрасли» [47] – подраздел новостного портала, в котором публикуется информация о различных отраслях российской экономики и о том, как бизнес функционирует в условиях динамично меняющейся

экономической и геополитической обстановки.

- «Тинькофф журнал» [48] – медиа о личных финансах и жизни в России. В одном из потоков[11] на сайте присутствует инвестиционная информация и публикации о деятельности различных компаний.
- «Хабр» [49] – веб-сайт в формате системы тематических коллективных блогов (хабами) с элементами новостного сайта. В разделах «Администрирование», «Менеджмент» и «Маркетинг» ежедневно публикуется информация о деятельности различных компаний.
- «Лента.ru» [50] – новостное интернет-издание. В потоке «Экономика» публикуются краткие новостные сводки про организации на российском рынке.
- «Коммерсантъ» [51] – новостной портал популярной советской и российской газеты с усиленным деловым блоком, в котором публикуется информация о финансах и бизнесе.
- «МФД-ИнфоЦентр» [52] – информационное агентство, специализирующееся на финансовой информации и создании современных программных продуктов для банков, инвестиционных компаний, корпоративных и индивидуальных инвесторов. На сайте публикуется актуальные котировки акций, курсы валют, а также присутствует форум трейдоров.
- «Московская биржа» [53] – новостной портал при единственной в России многофункциональной биржевой площадкой по торговле акциями, облигациями, производными инструментами, валютой, инструментами денежного рынка и товарами.
- «ТАСС» [54] – российское государственное федеральное

[11] Поток – подраздел на новостном сайте с определенной категорией или темой, публикуемой там информации.

информационное агентство с публикацией наиболее актуальной информации с открытым архивом новостей.

Все указанный интернет-издания выкладывают подходящую для рассмотрения информацию в рамках темы выпускной квалификационной работы. Однако необходимо было правильно агрегировать публикации об активности компании с таких сайтов или найти готовое решение, способное в одном месте предоставить полную сводку новостей и взаимодействий компании для дальнейшего использования.

2.1.3 СПАРК-Интерфакс

СПАРК-Интерфакс — это аналитическая система, предоставляющая пользователям информацию о российских компаниях и социально-экономическом состоянии регионов и отраслей в России. Данный программный комплекс является универсальным решением, не привязанным к какой-либо конкретной корпоративной модели работы с контрагентами [55].

Информационный ресурс содержит сведения о юридических лицах и индивидуальных предпринимателях, зарегистрированных на территории Российской Федерации и ряда стран СНГ.

Глубина информационного архива системы СПАРК превышает 15 лет, а сам архив сформирован на основе сведений, полученных из официальных источников.

На сайте помимо общей информации о конкретной компании присутствует отдельная вкладка, посвященная медийной активности организации – «Публикации в СМИ». Раздел включает в себя результаты поиска по наименованию компании или ее руководителю из более чем 40 000 открытых источников сети Интернет и СМИ, в том числе закрытых новостных лент агентства «Интерфакс». В разделе отражается график изменения числа публикаций, отдельно показано число рисковых публикаций. Также в разделе размещены риск-факторы и деловые темы, обнаруженные за рассматриваемый

период и топ наиболее заметных публикаций за рассматриваемый период. Для компаний, у которых рассчитан индекс деловой репутации (ИРР), отображается график его изменения.

Благодаря удобному функционалу и возможности выгружать отчеты по компаниям сбор данных об организациях и их активности будет реализован с помощью данной аналитической системы.

2.2 Рассмотрение общего пайплайна моделей GNN

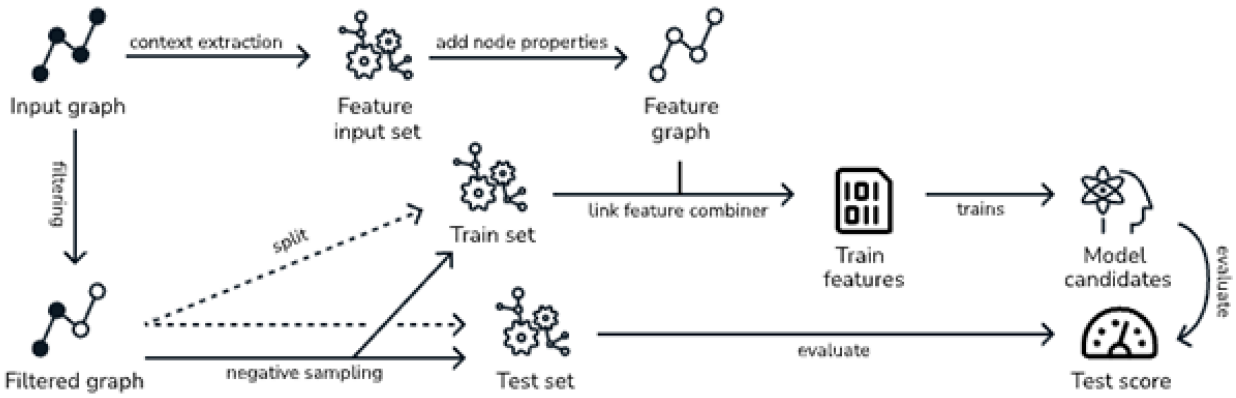


Рис. 2. Пайплан обучения графовых нейронных сетей.

На (рис. 2) представлен общий подход к созданию моделей глубокого обучения, основанных на графах. Его можно разделить на следующие этапы:

1. Фильтрация и определение типов связей в гетерогенном графе;
2. Извлечение контекста и фичей узлов из изначального графа;
3. Разбиение графа на обучающую и валидационную выборки;
4. Негативное семплирование графа для последующего обучения;
5. Обучение и оценка качества модели.

Если же более подобно рассматривать этап обучения модели, то можно выделить три основных его ступени (рис. 3):

- Кодирование (Encoding phase) – создание численного представления исходного графа;

- Обмен сообщениями (Message Passing) – вычисление текущего состояния каждого узла, в соответствии с состояниями вершин в окрестности;
- Декодирование (Decoding phase) – вычисление итогового прогноза.

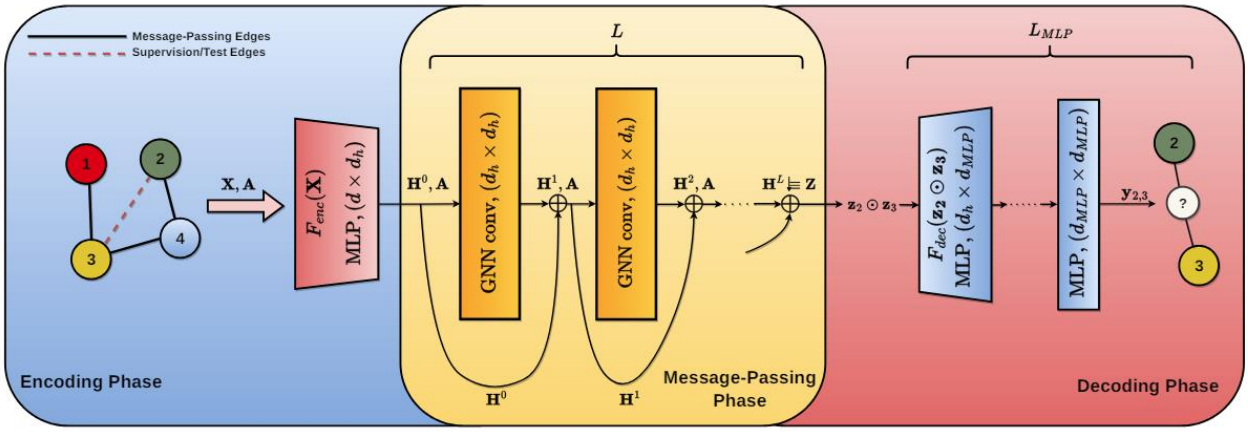


Рис. 3. Пример последовательности обучения модели предсказания связей в графе.

2.2.1 Энкодер: создание векторного представления графа

Как уже упоминалось в первой главе, задача энкодера заключается в создании эмбеддингов графа на основе его структурных и внутренних характеристиках. Существует два основных подхода к формированию эмбеддингов графов:

1. для вершин графа
2. для вершин и рёбер

Применение эмбеддингов для графа позволяет получить следующие преимущества, по сравнению с использованием «сырой» структуры исходного графа:

- Становится возможным использование математического аппарата машинного обучения, для работы с векторами.
- Экономия использования памяти, как оперативной, так и накопителя.

Так как векторизация графа позволяет получить его отображение в пространство меньшей размерности.

- Ускорение вычислений, по сравнению с традиционными графовыми моделями.

Для построения эмбеддингов исходный граф знаний представляется в виде набора троек HRT: head (объект), relation (связь), tail (второй объект).

Получение векторного представления графа можно разделить на следующие этапы:

1. Отбор метода представления узлов и вершин графа в непрерывное векторное пространство, на основе ожидаемого результата: вектор, матрица, тензор и на основе того, какие данные необходимо векторизовать: узлы или связи.
2. Выбор модели эмбеддинга, а именно – функции скоринга (правдоподобия отображения – scoring function) для каждого набора HRT.
3. Непосредственно обучение выбранной модели. Задача минимизации функции скоринга для всех наборов триплетов в исходных данных.

2.2.2 Генерация случайных примеров при обучении

Для обучения моделей необходимо составить полный набор отношений между объектами. Для узлов V и связей E набор всех возможных троек T (HRT) получается с помощью декартового произведения $T = V \times E \times V$. Гетерогенный граф представляет собой подмножество $K \in T$. Есть два метода создания отрицательных наборов троек: предположение о закрытом и открытом[12] мире [56]. В первом случае, все тройки, которые, не входя в K

[12] Предположение об открытости мира, (ПОМ) — предположение в формальной логике о том, что истинность утверждения не зависит от того, «известно» ли какому-либо наблюдателю или агенту о верности данного утверждения. Оно противоположно предположению о замкнутости мира, из

считаются неверными, во втором – неизвестными.

Техники по генерации отрицательных троек обычно создают их с помощью изменения в уже имеющихся тройках head, relation или tail на неправильные.

2.2.3 Message Passing

Концепт передачи сообщений (message passing) заключается в следующем - каждая вершина графа имеет внутреннее состояние. Каждую итерацию это внутреннее состояние пересчитывается, основываясь на внутренних состояниях соседей в окрестности. Каждый сосед влияет на состояние вершины, так же, как и вершина влияет на состояния соседей (рис. 4).

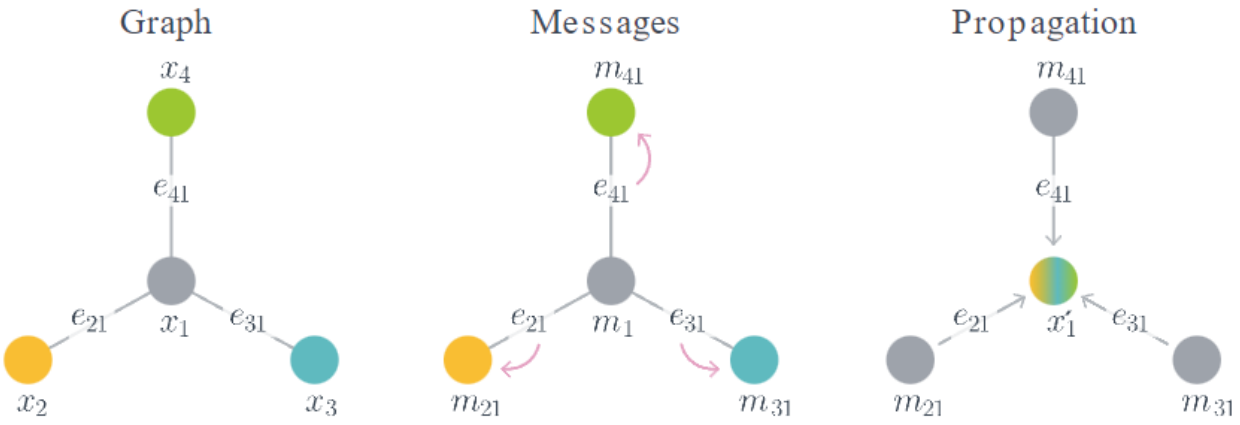


Рис. 4. Графическое представление парадигмы message passing.

Пусть $x_v \in \mathbb{R}^{d_1}$ будет фичей для узла v, и $w_e \in \mathbb{R}^{d_2}$ является фичей связи (u, v). Парадигма передачи сообщений определяет следующие вычисления по узлам (Node-wise) и ребрам (Edge-wise) на шаге обучения $t + 1$ (2.1, 2.2):

которого следует, что ложно любое утверждение, о котором не известно, что оно верно.

$$Edge-wise:\ \ m_e^{(t+1)} = \phi\left(x_v^{(t)}, x_u^{(t)}, w_e^{(t)}\right), (u,v,e) \in \mathcal{E} \qquad (2.1)$$

$$Node-wise:\ \ x_v^{(t+1)} = \psi\left(x_v^{(t)}, \rho\left(\left\{m_e^{(t+1)}: (u,v,e) \in \mathcal{E}\right\}\right)\right) \qquad (2.2)$$

В приведенных выше уравнениях для каждого ребра определена функция сообщения ϕ (message function), которая генерирует сообщение путем объединения фичи ребра с фичами входящих в него узлов. ψ это функция обновления (update function), определенная на каждом узле для обновления фичей узла путем агрегирования входящих сообщений с помощью функции уменьшения ρ (reduce function) [57].

2.3 Используемые модели для предсказания связей в графах

Для обучения моделей предсказания связей необходимо было рассмотреть различные технические реализации к формированию глубоких нейронных сетей и адаптировать их под структуру нашего гетерогенного графа. Далее будут описаны структурные части и полноценные реализации GNN для построения итоговой рекомендательной системы.

2.3.1 GraphSage

GraphSage, представленный в работе [64], представляет из-себя фреймфорк для индуктивного изучения представлений на больших графах. Эта модель используется для создания низкоразмерных эмбеддингов для узлов.

Основной подход заключается в минимизации каждого узлового эмбеддинга z_i после расчета, чтобы явно отобразить схожесть соседних вершин в графе (2.3)

$$L(z_i) = -log\left(\sigma\left(z_i^T z_j\right)\right) - k_n \cdot \mathbb{E}_{j' \sim p_n} log\left(1 - \sigma\left(z_i^T z_{j'}\right)\right), \qquad (2.3)$$

где:

- j – узел, находящийся с i узлом при некотором случайном блуждании;

- p_n – распределение негативного семплирования;
- k_n – количество негативных примеров.

Если мы сосредоточимся на случайных блужданиях длиной 2, то вышеупомянутая потеря сведется к цели предсказания связи. GraphSage не рассматривает все отрицательные связи, а использует отрицательную выборку, чтобы рассматривать только k_n отрицательных пар (i, j') для каждой положительной пары (i, j), что больше подходит для больших графов. Сам же слой, который мы будем использовать в нескольких из моделей, можно описать следующими уравнениями (2.4, 2.5, 2.6):

$$h_{N(i)}^{l+1} = aggregate\left(\left\{h_j^l, \forall j \in N(i)\right\}\right) \tag{2.4}$$

$$h_i^{l+1} = \sigma\left(W \cdot concat\left(h_i^l, h_{N(i)}^{l+1}\right)\right) \tag{2.5}$$

$$h_i^{l+1} = norm\left(h_i^{l+1}\right) \tag{2.6}$$

2.3.2 Graph Convolutional Network (GCN)

Графовая сверточная сеть [42], так же как и GraphSage, используется для создание улучшенных моделей эмбеддингов узлов совместно с внутренними характеристиками этих вершин. Математически GCN следует следующей формуле (2.7):

$$h^{l+1} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}h^l W^l\right), \tag{2.7}$$

где:

- h^l – l-слой сети;
- $\sigma(\cdot)$ – нелинейная функция активации;
- W – матрица весов для этого слоя;
- $\widetilde{D}$ – матрица степеней узлов графа;
- $\tilde{A}$ – матрица смежности графа.

Так, входная размерность h^0 представляет собой $N \times D$, где N – количество

узлов, а D – количество используемых фичей. Таким образом, мы можем объединить несколько слоев в цепочку, чтобы получить представление на уровне узла с формой $N \times F$, где F - размер вектора характеристик выходного узла.

Так как далее мы будем использовать две модели, основанные на GraphSage и GCN, необходимо описать их основное отличие. GCN по своей сути являются преобразовательными, т. е. они могут генерировать вложения только для узлов, присутствующих в фиксированном графе, во время обучения. Это означает, что если в будущем граф эволюционирует и в него будут добавлены новые узлы (невидимые во время обучения), то нам нужно будет переобучить весь граф, чтобы вычислить вложения для нового узла. В то же время алгоритм GraphSage использует богатые возможности узлов и топологическую структуру окрестности каждого узла одновременно, чтобы эффективно генерировать представления для новых узлов без переобучения. В дополнение к этому GraphSage выполняет выборку окрестностей, что обеспечивает алгоритму уникальную способность масштабировать граф.

2.3.3 Network in Graph Neural Network (NGNN)

NGNN была представлена в работе [62]. Главной особенностью является то, что вместо того, чтобы добавлять или расширять имеющиеся слои графовой нейронной сети, NGNN углубляет модель GNN, добавляя слои нелинейной нейронной сети прямого распространения в каждый слой GNN (рис. 5). Подобный слой можно описать как нелинейную функцию (2.8):

$$h^{(l+1)} = \sigma\left(f_w(G, h^l)\right), \tag{2.8}$$

где:

- $h^{(0)} = X$ – входящие характеристики узла;
- G – входящий граф;
- h^l – эмбеддинги узлов;

- L – количество слоев в нейронной сети;
- $\sigma(\cdot)$ – нелинейная функция активации

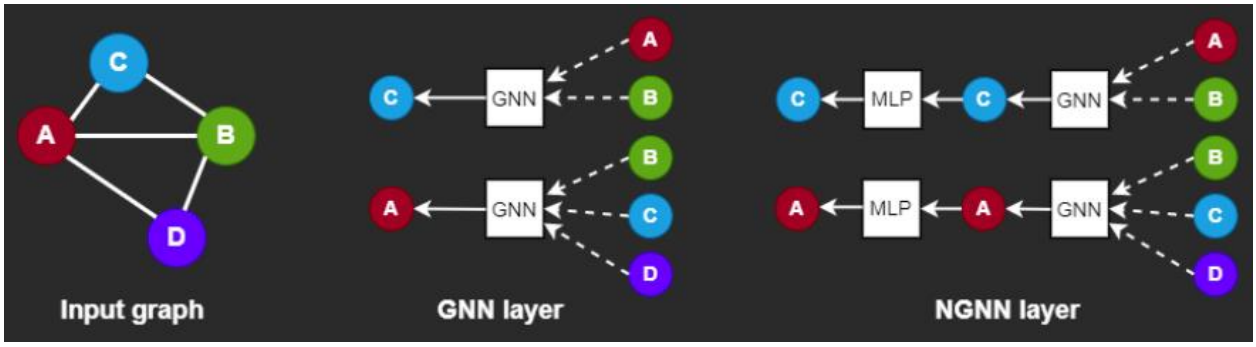


Рис. 5. Структура слоев NGNN.

По сути, NGNN это нелинейное преобразование исходных эмбеддингов узлов l-го слоя. Несмотря на свою простоту, метод NGNN весьма эффективен. Кроме того, он не требует больших затрат памяти и может работать с различными методами обучения.

По мере увеличения количества слоев GNN и количества итераций обучения представления узлов внутри одного и того же связного компонента будут стремиться сходиться к одному и тому же значению. NGNN использует простой нелинейное преобразование после определенных слоев GNN для решения проблемы чрезмерного сглаживания.

Данный подход будет реализован совместно с моделями построения эмбеддингов GraphSage и GCN для достижения более точных предсказаний, так как их совместное использование позволяет достичь лучших результатов на больших графах [63].

2.3.4 Heterogeneous Graph Transformer (HGT)

Heterogeneous Graph Transformer (HGT) [65] - это графовая архитектура нейронной сети, которая может работать с достаточно большими гетерогенными и динамическими графами за счет применения

трансформерной[13] структуры.

На (рис. 6) представлена общая структура модели HGT. Для заданного гетерогенного подграфа в качестве таргет-узла t, s_1 и s_2 - узлы источники, модель HGT использует связи $e_1 = (s_1, t)$ и $e_2 = (s_2, t)$ и соответствующие им мета отношения $< \tau(s_1), \phi(s_1), \tau(t) >$ и $< \tau(s_2), \phi(s_2), \tau(t) >$ в качестве входных данных для изучения контекстуализированного представления h^l для каждого узла, который может быть использован для последующих задач. HGT состоит из трех основных операторов:

- Attention - который оценивает взаимную важность каждого исходного узла относительно других;
- Message – для распространения сообщений (изменений весов) между узлами;
- Aggregate - агрегация разнородных сообщений для конкретной цели.

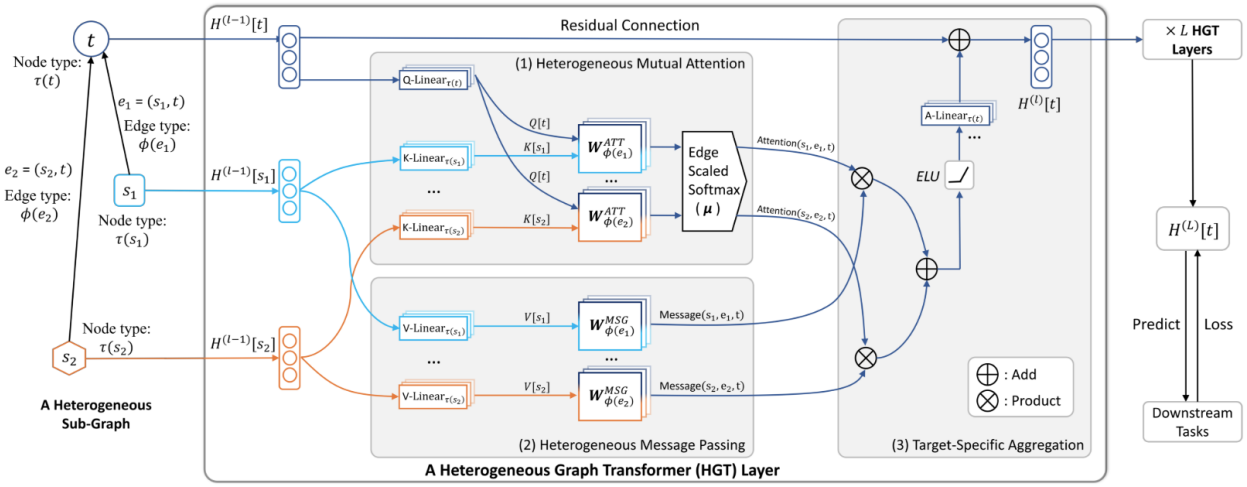


Рис. 6. Общая структура нейронной сети HGT.

Для расчета Attention для набора триплета (s, e, t) используется следующий алгоритм (2.9, 2.10, 2.11, 2.12):

[13] Трансформер – структура глубоких нейронных сетей, основанная на механизме внимания без использования рекуррентных нейронных сетей.

$$Attention(s,e,t) = Softmax\left(||_{i\in[1,h]} ATT - head^i(s,e,t)\right), \quad (2.9)$$

$$ATT - head^i(s,e,t) = \left(K^i(s) W^{ATT}_{\phi(e)} Q^i(t)^T\right) \cdot \frac{\mu_{\tau(s),\phi(s),\tau(t)}}{\sqrt{d}}, \quad (2.10)$$

$$K^i(s) = KLinear^i_{\tau(s)}(H^{l-1}[s]), \quad (2.11)$$

$$Q^i(t) = QLinear^i_{\tau(t)}(H^{l-1}[t]). \quad (2.12)$$

Для расчета сообщения (Message) на передачу для триплета (s,e,t) (2.13, 2.14):

$$Message(s,e,t) = ||_{i\in[1,h]} MSG - head^i(s,e,t), \quad (2.13)$$

$$MSG - head^i(s,e,t) = MLinear^i_{\tau(s)}(H^{l-1}[s]) W^{MSG}_{\phi(e)} \quad (2.14)$$

Для агрегирования (Aggregate) сообщений в таргет-узле t (2.15):

$$\widetilde{H}^l[t] = \sum_{\forall s\in N(t)} (Attention(s,e,t) \cdot Message(s,e,t)) \quad (2.15)$$

2.3.5 GANTE-T

Трансиндуктивная модель GATNE-T [66] – нейронная сеть, способная к восприятию мультиплексных гетерогенных сетей. Иными словами, данная GNN приспособлена к использованию на гетерогенных графах с множеством типов связей и узлов (рис. 7).

Основной ее особенностью является отельное рассмотрение эмбеддингов связей и узлов для их различных типов. Для создания этих представлений используется улучшенная модель GraphSage с применением трансформенной логики.

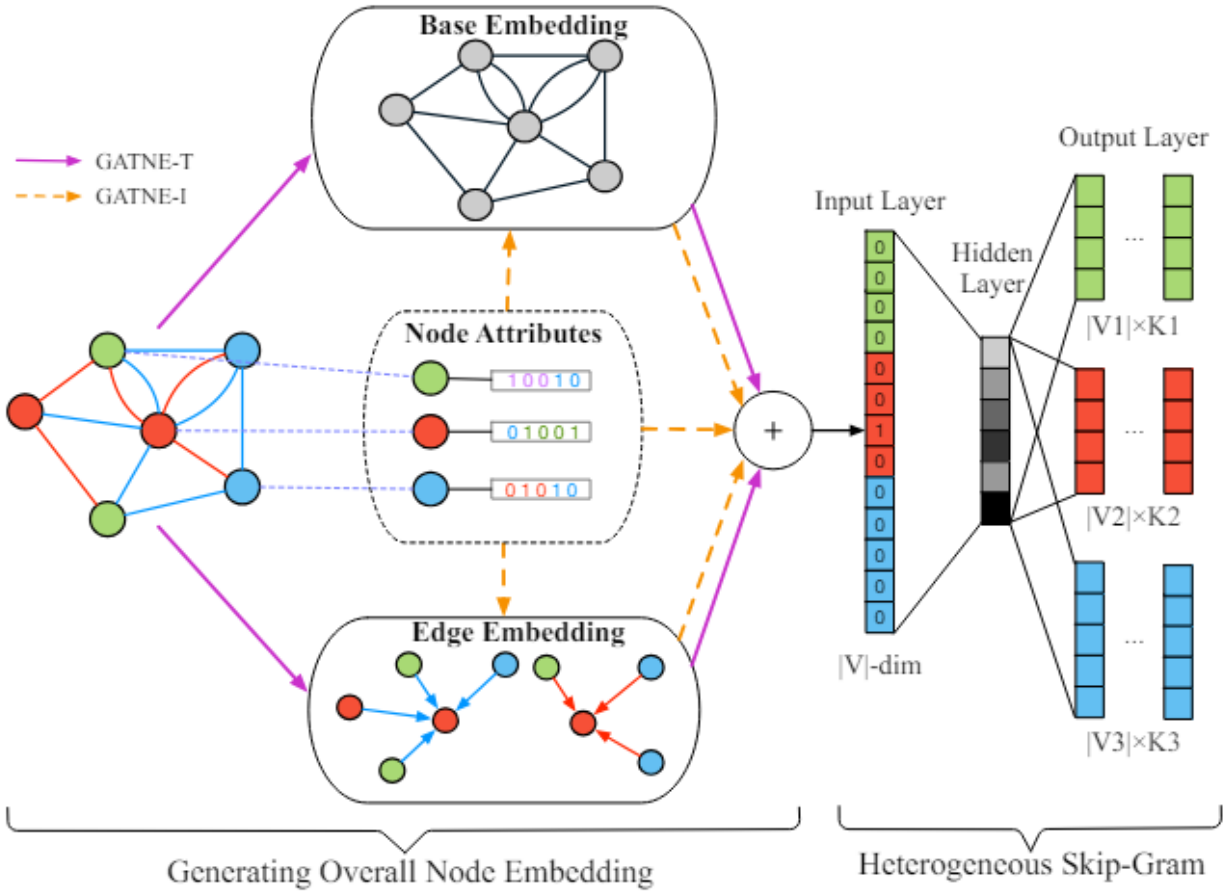


Рис. 7. Общая структура нейронной сети GATNE-T.

Основные компоненты GATNE-T включают в себя (рис. 8):

- Трансформерные блоки: Эти блоки используются для изучения зависимостей между узлами в графе. Они позволяют модели извлекать и использовать контекстуальные признаки узлов и связей.
- Генеративно-состязательная сеть (GAN): GATNE-T включает генератор, который генерирует фиктивные представления узлов, а также дискриминатор, который пытается различать реальные и сгенерированные узлы. Этот процесс приводит к тому, что генератор улучшает свои навыки создания реалистичных представлений узлов.
- Функция потерь: для обучения GATNE-T используется комбинация различных функций потерь, включая функцию потерь для генератора и функцию потерь для дискриминатора. Цель состоит в том, чтобы

обучить генератор создавать представления узлов, которые могут обмануть дискриминатор, а также обучить дискриминатор различать реальные и сгенерированные представления узлов.

Algorithm 1: *GATNE*

Input: network $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, embedding dimension d, edge embedding dimension s, learning rate η, negative samples L, coefficient α, β.

Output: overall embeddings $\mathbf{v}_{i,r}$ for all nodes on every edge type r

Initialize all the model parameters θ.
Generate random walks on each edge type r as $\mathcal{P}_r$.
Generate training samples $\{(v_i, v_j, r)\}$ from random walks $\mathcal{P}_r$ on each edge type r.
while *not converged* **do**
    foreach $(v_i, v_j, r) \in$ *training samples* **do**
        Calculate $\mathbf{v}_{i,r}$ using Equation (6) or (13)
        Sample L negative samples and calculate objective function E using Equation (17)
        Update model parameters θ by $\frac{\partial E}{\partial \theta}$.

Рис. 8. Псевдокод, описывающий логику работы алгоритма GATNE.

2.3.6 Position-aware Graph Neural Networks (P-GNN)

Position-aware Graph Neural Networks (P-GNN) [67] - это класс GNN для вычисления эмбеддингов узлов, которые включают информацию о местоположении узла относительно всех других узлов в сети, сохраняя при этом индуктивные возможности и используя внутренние характеристики узлов. Ключевая особенность подхода заключается в том, что положение узла может быть зафиксировано с помощью эмбеддинга с низким уровнем искажений путем количественной оценки расстояния между данным узлом и набором опорных узлов.

P-GNN содержат следующие ключевые компоненты (рис. 9):

- k наборов опорных узлов размера S_i.
- Функция вычисления сообщений F, которая объединяет информацию о

характеристиках двух узлов с их сетевым расстоянием.

- Матрица M сообщениями опорных узлов, где каждая строка i представляет собой сообщение M_i с установленными привязками, вычисленное с помощью F.
- Обученные функции агрегации AGG_M, AGG_S, которые агрегируют или преобразуют информацию об элементах узлов из опорного набора, далее агрегируя ее по всем опорным набором.
- Обучаемый вектор w, который проецирует матрицу сообщений M в низкоразмерное пространство эмбеддинга $z \in R_k$.

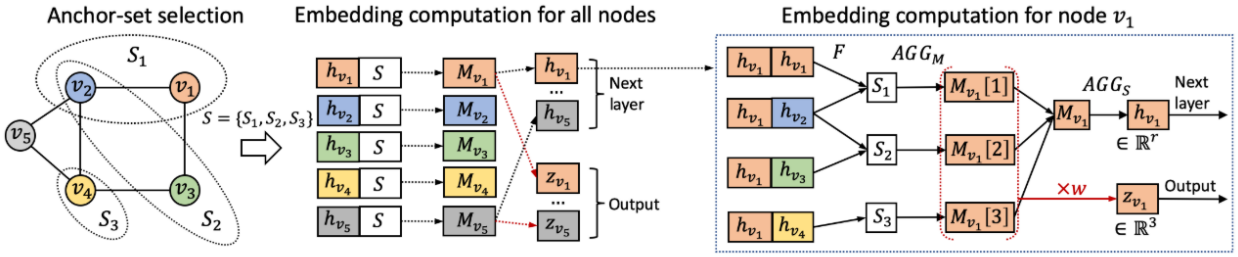


Рис. 9. Общая структура нейронной сети P-GNN.

Сам же алгоритм работы сетей на основе идеи P-GNN описан на (рис. 10).

Algorithm 1 The framework of P-GNNs

Input: Graph $G = (\mathcal{V}, \mathcal{E})$; Set S of k anchor-sets $\{S_i\}$; Node input features $\{\mathbf{x}_v\}$; Message computation function F that outputs an r dimensional message; Message aggregation functions AGG_M, AGG_S; Trainable weight vector $\mathbf{w} \in \mathbb{R}^r$; Non-linearity σ; Layer $l \in [1, L]$
Output: Position-aware embedding $\mathbf{z}_v$ for every node v

$\mathbf{h}_v \leftarrow \mathbf{x}_v$
for $l = 1, \ldots, L$ **do**
  $S_i \sim \mathcal{V}$ for $i = 1, \ldots, k$
  for $v \in \mathcal{V}$ **do**
    $\mathbf{M}_v = \mathbf{0} \in \mathbb{R}^{k \times r}$
    for $i = 1 \ldots, k$ **do**
      $\mathcal{M}_i \leftarrow \{F(v, u, \mathbf{h}_v, \mathbf{h}_u), \forall u \in S_i\}$
      $\mathbf{M}_v[i] \leftarrow \text{AGG}_M(\mathcal{M}_i)$
    end for
    $\mathbf{z}_v \leftarrow \sigma(\mathbf{M}_v \cdot \mathbf{w})$
    $\mathbf{h}_v \leftarrow \text{AGG}_S(\{\mathbf{M}_v[i], \forall i \in [1, k]\})$
  end for
end for
$\mathbf{z}_v \in \mathbb{R}^k, \forall v \in \mathcal{V}$

Рис. 10. Псевдокод, описывающий логику работы алгоритма P-GNN.

2.4 Метрики для оценки качества моделей

Hits@K [61] - описывает отношение правильно предсказанных объектов, которые появляются среди первых K объектов в списке всех K предсказаний, отсортированному по функции скоринга. Она означает, что для n процентов объектов из одного графа знаний эквивалентный объект из второго графа знаний находится среди ближайший K соседей в векторном пространстве (2.16).

$$Hits@K = \frac{|\{t \in K_{test} \mid rank(t) \leq k\}|}{|K_{test}|} \tag{2.16}$$

Mean Rank (MR) описывает арифметическое среднее по всем индивидуальным рангам (результат функции скоринга). Это среднее положение первого корректного ответа в задаче предсказания (2.17).

$$MR = \frac{1}{|L|} \sum_{r \in L} r \tag{2.17}$$

Mean Reciprocal Rank аналогична Mean Rank, за тем исключением что берётся обратное значение для ранга (2.18):

$$MRR = \frac{1}{|L|}\sum_{r \in L}\frac{1}{r} \tag{2.18}$$

2.4 Вывод

Таким образом, нам удалось найти подходящий источник данных о медийной активности компаний, который впоследствии будет использован для формирования датасета.

Мы рассмотрели общую структуру графовых нейронных моделей. На основе приведенных во главе примеров технологических реализации тех или иных алгоритмов GNN будет сформирован набор нейронных сетей, на котором будет обучен гетерогенный граф компаний и уникальных идентификаторов отраслевой принадлежности.

ГЛАВА 3. АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

3.1 Создание поискового робота (crawler) для выгрузки данных

3.1.1 Формирование изначального списка интересующих компаний

Помимо поиска источников данных также было необходимо найти первоначальный список компаний, информация о которых будет выгружена в первую очередь.

Предполагается, что такие компании не должны быть крупными (про такие организации больше различных новостей и публикаций, следовательно, связаны они будет с множеством других компаний, для которых восстановить отраслевую принадлежность будет затруднительно из-за «перенасыщения» информацией), но при этом какая-либо медийная информация о них должна присутствовать. Идеальным вариантом является средняя по цитируемости организация.

В качестве отправной точки были взяты компании с сайта рейтингового агентства RAEX («РАЭКС-Аналитика»), которые присутствовали в списке крупнейших компаний России по объему реализации продукции (рис 1).

Рис. 1. Отрасли компаний, представленных в списке RAEX

3.1.2 Алгоритм работы поискового робота

Поисковой робот или же веб-краулер (Web crawler) – алгоритм автоматического интернет-серфинга, цель которого заключается в поиске

интересующей информации с веб-ресурсов. Краулер анализирует содержимое страницы, сохраняет его в специальном виде, и отправляется по ссылкам на следующие страницы. Порядок обхода страниц, защита от зацикливания, а также критерии выделения значимой информации определяются алгоритмами информационного поиска, заранее прописанными разработчиком. В большинстве случаев переход от одной страницы к другой осуществляется по ссылкам, содержащимся на первой и последующих страницах (рис. 2).

Рис. 2. Архитектура веб-кравлера

Для реализации поискового робота и выгрузки информации о медийной активности компании из «СПАРК-Интерфакс» воспользуемся библиотекой selenium [58]. Selenium WebDriver – это программная библиотека для управления браузерами. В рамках проекта разрабатывается специальный драйвер для браузера Chrome – Chromium. Благодаря этому драйверу и реализованному функционалу на языке программирования python пользователи могут автоматизировать действия, производимые в браузере, и, как следствие, настроить собственного поискового робота для выгрузки информации.

Поиск внутри сайта будет реализован по ИНН[14] и ОГРН[15] компаний. Для каждой такой организации будет выгружаться две отдельных страницы:

- «Карточка компании» - страница с основами данными о компании: полным наименованием, адресом, КПП, ОКПО и др.
- «Публикации в СМИ» - страница с примерами новостей о компании, а также организациями, с которыми она совместно упоминается в таких публикациях.

Алгоритм созданного поискового робота схож с алгоритмом, описанном на (рис. 1):

- Поиск страницы компании по ее уникальному идентификатору внутри «СПАРК-Интерфакс»;
- Поиск страницы «Карточка компании» по ее ссылке и последующая выгрузка основной информации из таблиц;
- Поиск страницы «Публикации в СМИ», фильтрация новостей (поиск публикации о компании за прошедший год) и последующая выгрузка примеров публикаций и списка упоминаемых организаций.

Таким образом была собрана информация о пятистах компаниях, входящих в список RAEX. После составления общей выборки совместно упоминаемых компаний (11304 организации) для всех организаций из этого списка повторно была выгружена основная информация, для полного покрытия данными итогового графа.

3.2 Предорбработка данных

После выгрузки данных с сайта «СПАРК-Интерфакс», а также

---

[14] ИНН - идентификационный номер налогоплательщика.

[15] ОГРН - основной государственный регистрационный номер, который содержит основную информацию о юридическом лице или ИП.

составлении и выгрузки кодов ОКВЭД необходимо было сформировать датафреймы с очищенной и полезной информацией:

- Датасет с полной информацией о любой компании из упомянутых в источнике с кодом ОКВЭД для связи с графом отраслевой иерархии;
- Датасет с расшифровками кодов ОКВЭД и информацией о входящих в них группах и подгруппах;
- Датасет со связями между компаниями на основе их медийной активности и весами (количеством взаимных упоминаний между организациями).

3.2.1 Рассмотрение используемых фичей

Для собранных компаний или узлов графа организаций сохранялись следующие данные:

- ИНН и ОГРН компании в качестве уникального ключа для последующих объединений с остальными таблицами;
- ОКОПФ — это общероссийский классификатор организационно-правовых форм собственности, на основании которого субъектам предпринимательства после их регистрации присваиваются коды, позволяющие идентифицировать их принадлежность к организационно-правовой форме;
- Адрес, где компания была зарегистрирована;
- Полное название организации;
- ОКАТО — это общероссийский классификатор административно-территориальных образований. Он необходим, чтобы хранить информацию о географических объектах Российской Федерации;
- Примеры новостей, публикуемых о компании.

Отдельно был сформирован json файл, содержащий в себе все соединения между компаниями и степень взаимной цитируемости.

Для узлов, представляющих коды ОКВЭД использовались:

- Уникальный идентификатор кода для соединения с графом компаний;
- Описание конкретного кода ОКВЭД;
- Раздел ОКВЭД – латинская буква, обозначающая принадлежность древа кодов к определенному общему отраслевому сектору.

Текстовые переменные были преобразованы в эмбеддинги посредством предобученной модели BERT [59].

3.2.2 Создание гетерогенного графа

Для создания гетерогенного графа и последующего обучения моделей использовалась библиотека deep graph library (DGL) [60].

В итоговой сети присутствует:

- Два типа узлов:
  - «Companies» - компании,
  - «OKVED» - коды ОКВЭД,
- Три типа связей между узлами:
  - «related» - медийная связь между компаниями,
  - «industry» - отраслевая принадлежность компаний по ОКВЕДу,
  - «child/parent» - иерархическая связь кодов ОКВЕД.

Каждому узлу были переданы фичи, а связи получили свои веса.

Для визуализации данных был использован GNNLens2 [60] — это инструмент интерактивной визуализации для графических нейронных сетей. Он обеспечивает плавную интеграцию с библиотекой (DGL). На (рис. 3) отображен пример одного перехода от узла из графа компаний, (рис. 4) представляет собой сеть иерархии кодов ОКВЭД и (рис. 5) показывает итоговый гетерогенный граф перед обучением.

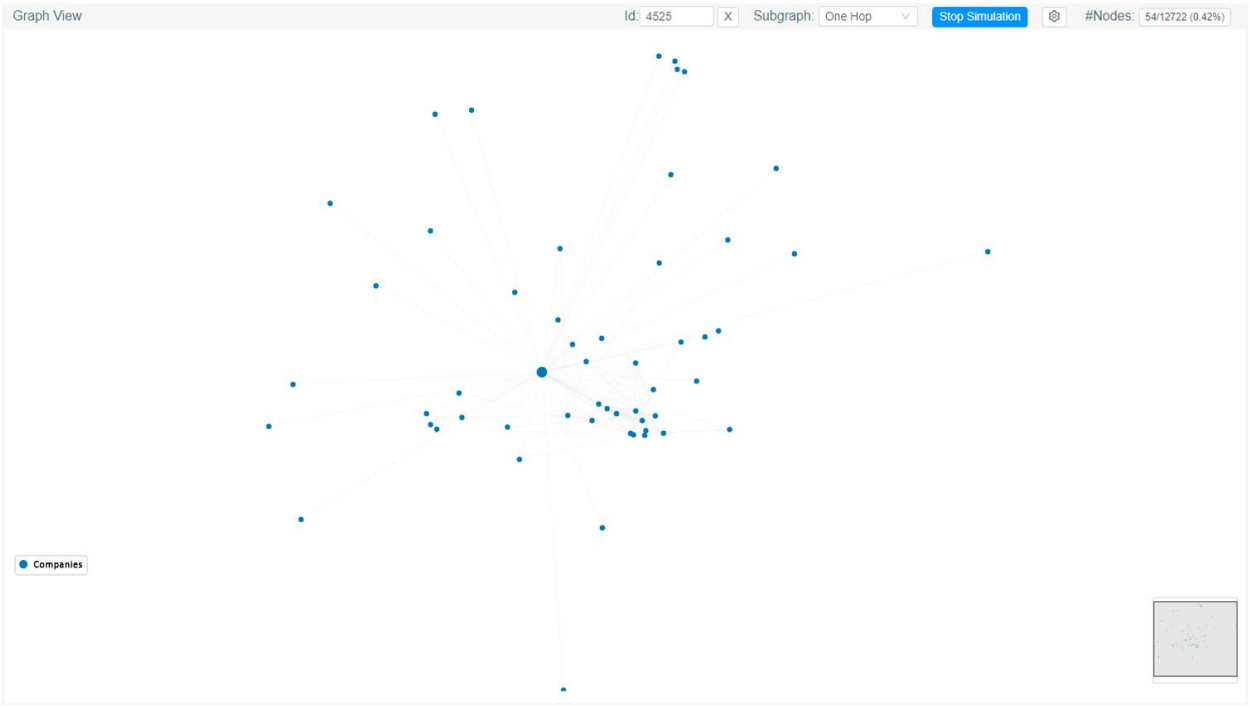


Рис. 3. Пример подграфа компаний с одним переходом от одного узла.

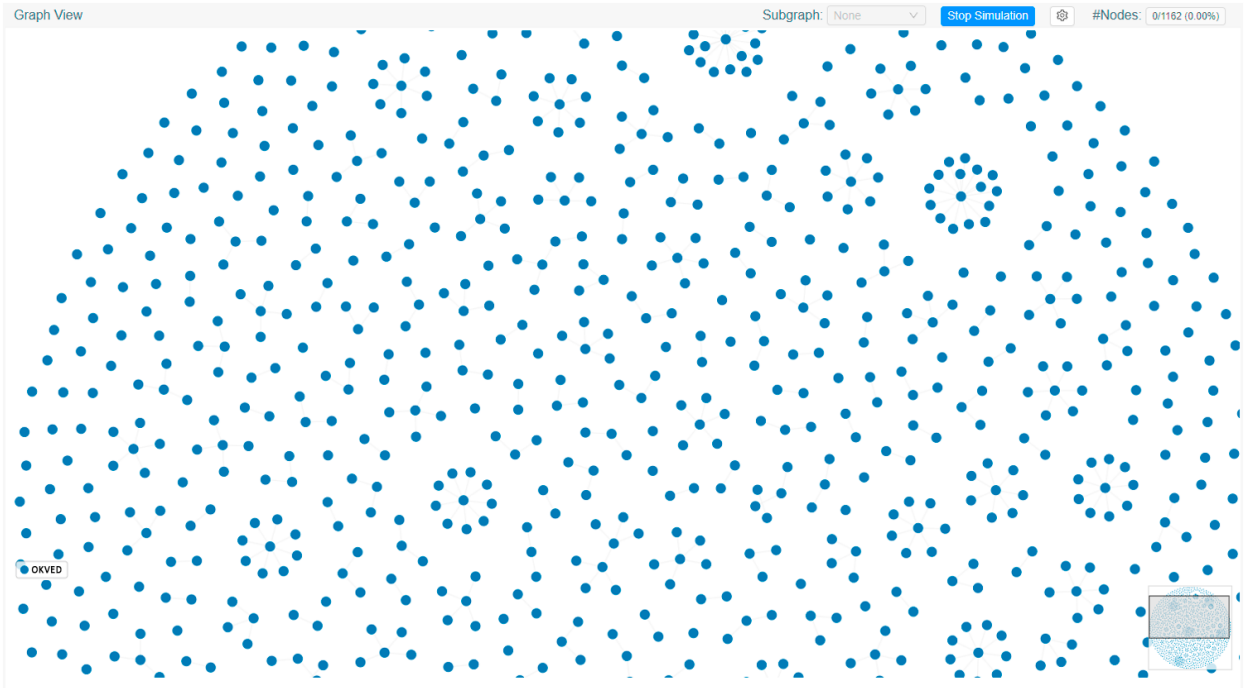


Рис. 4. Граф иерархии кодов ОКВЭД.

с

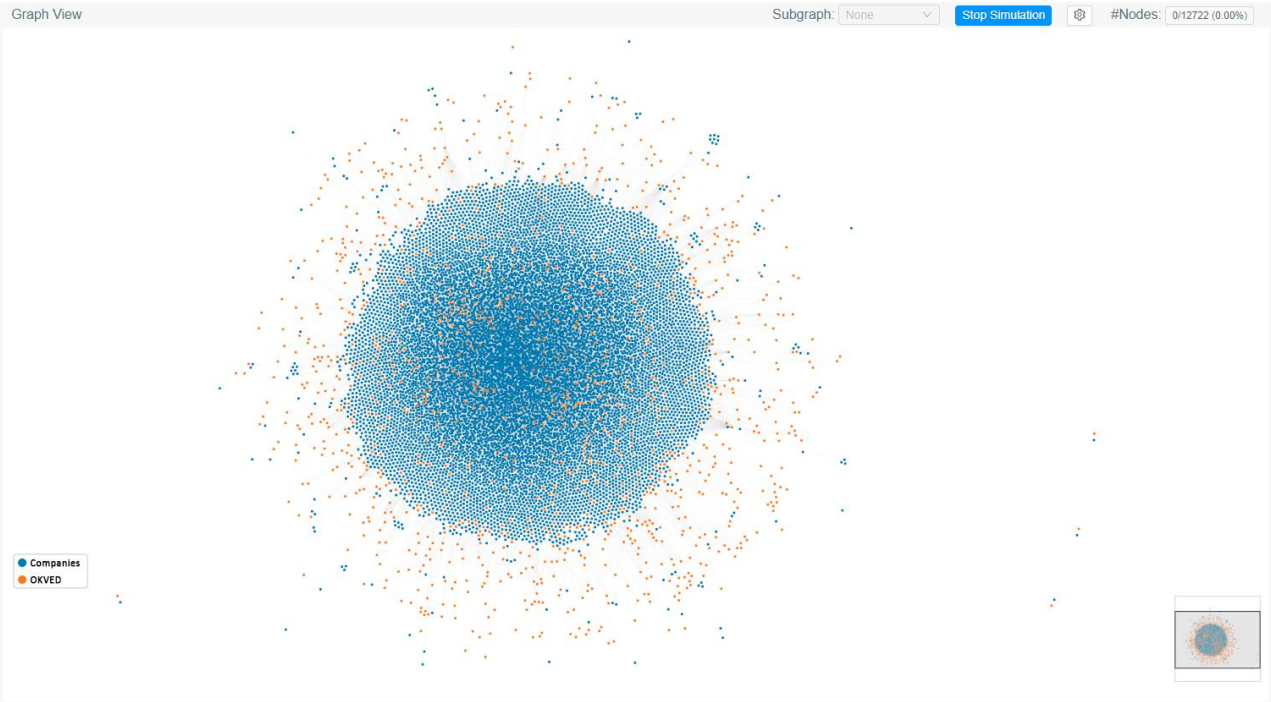


Рис. 5. Визуализация итогового гетерогенного графа.

3.3 Обучение моделей

Для собранных в предыдущей главе графовых данных были построены векторные представления и далее использованные в моделях глубокого обучения.

Для обучения GNN моделей необходимо настроить их гиперпараметры:

- Скорость обучения (learning rate) – гиперпараметр, который определяет, как быстро происходит обучение весов модели. Если его значение слишком маленькое, процесс обучения будет занимать много времени, так как будет дольше решение задачи на поиск локального минимума, но меньше вероятность его пропустить. Наоборот, если значение learning rate будет слишком большим, то задача поиска локального минимума не будет достигнута вовсе. Таким образом, правильная настройка данного гиперпараметра уменьшает затраты на время обучения модели и влияет на полученное качество.
- Количество эпох – данный гиперпараметр задаёт количество полных

проходов по обучающей выборке. При заданном малом количестве эпох модель может не достичь оптимальной точности, а при заданном излишнем количестве эпох обучения может произойти переобучение модели, а именно - модель утратит обобщающую способность и станет давать почти идеальные результаты на обучающей выборке, но не будет способна к работе с новыми данными.

3.4 Сравнительный анализ алгоритмов

Для первичного анализа качества моделей эмбеддинга исходный датасет был разделён на обучающую и валидационную выборку.

После обучения моделей был проведен расчет статистических метрик на тестовой выборке (таб. 1).

	Hits@1	Hits@2	Hits@3	Hits@5	Hits@10	MRR
NGNN + GraphSage	51,47%	60,34%	63,11%	74,21%	81,18%	0,41
NGNN + GCN	21,96%	51,53%	57,43%	63,04%	68,07%	0,14
HGF	70,02%	72,40%	76,92%	82,39%	89,23%	0,65
SEAL	35,08%	39,49%	42,16%	50,34%	59,90%	0,21
GANTE	65,60%	69,15%	71,03%	78,14%	82,30%	0,51
P-GNN	68,06%	71,58%	74,48%	81,33%	87,66%	0,54

Таб. 1. Результаты работы GNN моделей.

Визуализируем полученные значения метрик (рис. 5, 6).

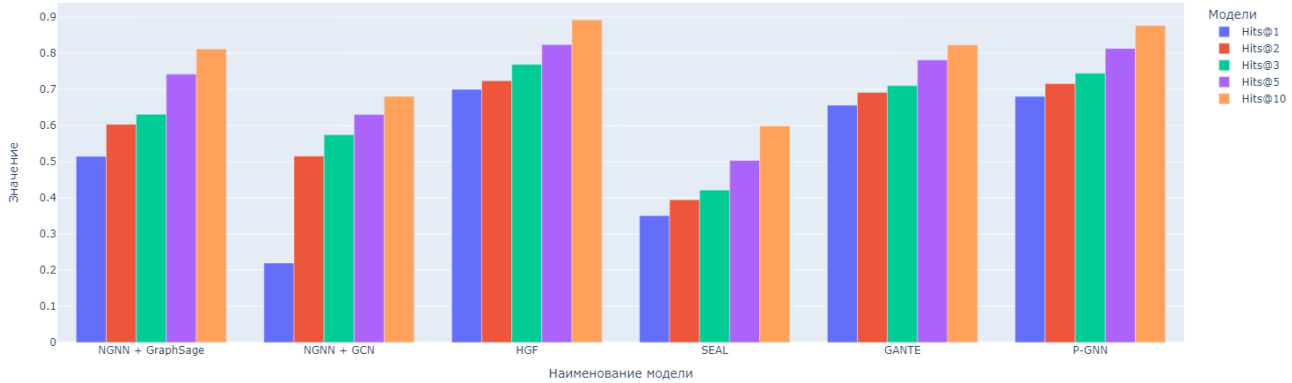


Рис. 5. Метрики Hits@1, Hits@2, Hits@3, Hits@5, Hits@10 для собранного

гетерогенного графа.

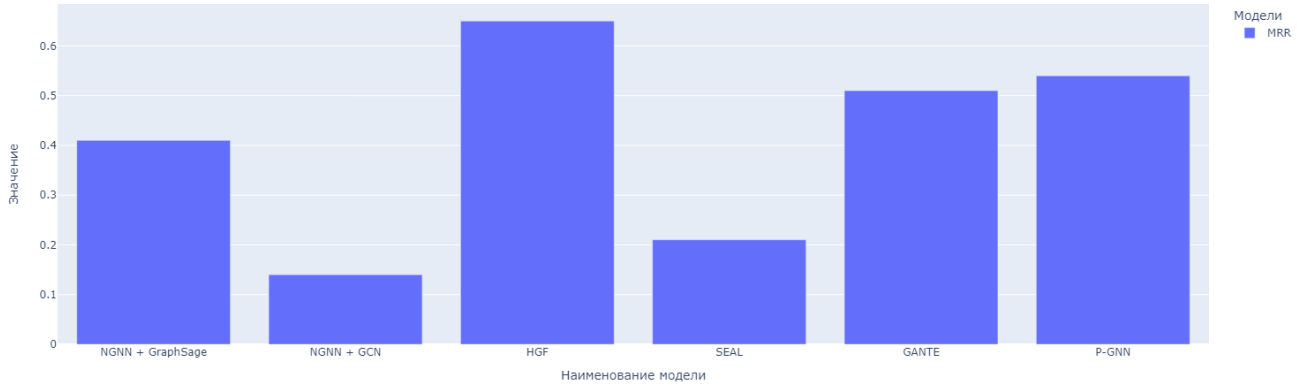


Рис. 6. Метрика MRR для собранного гетерогенного графа.

По значениям полученных метрик видно, что для данных о собранных данных получилось относительно хорошие результаты, однако датасет нуждается в расширении для достижения более точных предсказаний.

Также на основе значений метрики Mean Reciprocal Rank заметно, что лучше всего справились модели HGF, P-GNN и GANTE (в порядке убывания качества метрики), однако по метрике Hits@K лучшими являются HGF и P-GNN. Так как наилучшую результативность показала модель HGF, для дальнейшего анализа было принято решение взять именно ее.

3.5 Улучшение датасета с целью повышения качества модели

Для увеличения количества связей в графе между компаниями и улучшения качества предсказаний был произведен повторный сбор информации с помощью краулера из списка компаний, упоминаемых в изначальной выборке.

3.6 Анализ полученных результатов

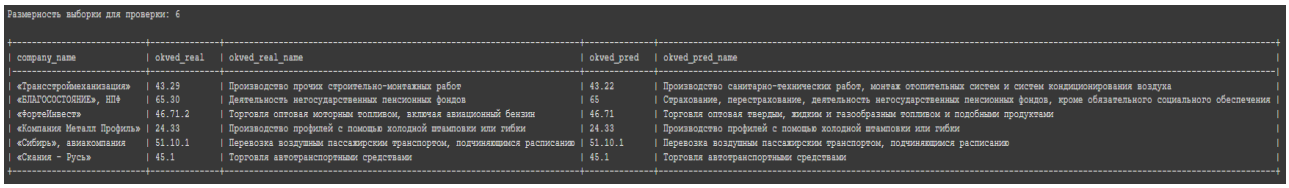
После дополнения графа и повторного обучения модели HGF были получены следующие классификационные метрики предсказания наличия связей между узлами в графе (таб. 2):

accuracy	precision	recall	f1-score
0,82868	0,79445	0,78690	0,79066

Таб. 2. Взвешенные метрики на валидационной выборке.

Для построенной модели был создан метод, который демонстрирует её работу в виде таблицы, состоящей из следующих полей (рис. 7):

- Наименование компании;
- Реальный основной код ОКВЭД компании;
- Расшифровка реального кода ОКВЭД;
- Предсказанный код ОКВЭД компании;
- Расшифровка предсказанного кода ОКВЭД.



Размерность выборки для проверки: 6

company_name	okved_real	okved_real_name	okved_pred	okved_pred_name
«Трансстроймеханизация»	43.29	Производство прочих строительно-монтажных работ	43.22	Производство санитарно-технических работ, монтаж отопительных систем и систем кондиционирования воздуха
«БЛАГОСОСТОЯНИЕ», НПФ	65.30	Деятельность негосударственных пенсионных фондов	65	Страхование, перестрахование, деятельность негосударственных пенсионных фондов, кроме обязательного социального обеспечения
«ФорвейИнвест»	46.71.2	Торговля оптовая моторным топливом, включая авиационный бензин	46.71	Торговля оптовая твердым, жидким и газообразным топливом и подобными продуктами
«Компания Металл Профиль»	24.33	Производство профилей с помощью холодной штамповки или гибки	24.33	Производство профилей с помощью холодной штамповки или гибки
«Сибирь», авиакомпания	51.10.1	Перевозка воздушным пассажирским транспортом, подчиняющимся расписанию	51.10.1	Перевозка воздушным пассажирским транспортом, подчиняющимся расписанию
«Скания - Русь»	45.1	Торговля автотранспортными средствами	45.1	Торговля автотранспортными средствами

Рис 7. Результаты предсказания модели.

Как видно по таблице (рис. 7) на представленной выборке модель показывают достаточно точные предсказания. Несмотря на некоторые неточности в предсказании подгрупп кодов ОКВЭД, она довольно часто попадает в итоговую группу, что также является удовлетворим результатом. Это говорит нам о том, что полученных данных уже достаточно для построения базовой модели рекомендаций. Однако итоговая точность модели указывает на то, что гетерогенный граф все еще нуждается в доработке: сеть все еще является недостаточно связной, а также необходимо рассмотреть некоторые варианты по расширению используемых фичей каждого узла.

ЗАКЛЮЧЕНИЕ

В ходе выполнения целей и задач данной выпускной квалификационной работы был произведён анализ предметной области для решаемой задачи. Были рассмотрены различные подходы в построении рекомендательных систем на основе задачи предсказания связей в графах.

В первой граве в теоретической справке были формализованы такие понятия как гетерогенный граф и задача link prediction. Выполнен сравнительный анализ методик реализации предсказания связей в графах. Их способность и скорость работы с большими объёмами данных.

Во второй главе было описаны особенности сбора датасета медийной активности компаний – проанализированы возможные источники новостных публикаций и отобран наиболее подходящий вариант. Рассмотрена общая структура графовых нейронных сетей. Описаны модели для дальнейшего обучения GNN на собственном наборе данных. Приведены используемые метрики в данном типе задач.

В третьей главе описана работа с моделями и приведён анализ полученных результатов, основанный на сравнении полученных метрик моделей. Выбрана наилучшая из моделей и дообучена до удовлетворимого результата.

Итоговая модель показывает достаточно хорошие метрики на представленном наборе данных, однако, существует несколько возможных путей улучшения ее репрезентации:

- Добавление оценки тональности новостей – необходимо научиться фильтровать положительные и отрицательные новости о компаниях.
- Добавить возможность определения дополнительных ОКВЭД компаний с возможностью их ранжирования.

- Улучшение работы модели при добавлении новых узлов (компаний) в набор данных.

Были применены современные средства программирования и технологии, в частности, python – один из самых популярных языков программирования в сферах анализа данных, включая широкий спектр библиотек и фреймворков.

На базе рассмотренных в данной работе алгоритмов можно создавать эффективные прикладные решения, в частности, в сфере рекомендательных систем.

Таким образом, задачи и цели выпускной квалификационной работы можно считать выполненными.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Fabian Beck, Michael Burch, Stephan Diehl, Daniel Weiskopf: «The State of the Art in Visualizing Dynamic Graphs». – 2014 г.
2. Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, Gabriele Monfardini: «The Graph Neural Network Model». – 2009 г.
3. Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang: «A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications». – 2018 г.
4. Ritwik Raj Saxena, Ritcha Saxena: «Applying Graph Neural Networks in Pharmacology». – 2024 г.
5. Aikta Arya, Pradumn Pandey, Akrati Saxena: «Node Classification Using Deep Learning in Social Networks». – 2022 г.
6. Mohamed Badiy, Fatima Amounas, Ahmad El allaoui, Younes Bayane: «Neural Network for Link Prediction in Social Network». – 2024 г.
7. Samarth Khanna, Sree Bhattacharyya, Sudipto Ghosh, Kushagra Agarwal, Asit Kumar Das: «Link Prediction for Social Networks using Representation Learning and Heuristic-based Features». – 2024 г.
8. Hoyeon Jeong, Young-Rae Cho, Jungsoo Gim, Seung-Kuy Cha, Maengsup Kim, Dae Ryong Kang: «GraphMHC: Neoantigen prediction model applying the graph neural network to molecular structure». – 2024 г.
9. Wenfang Zhang, Hongwei Shi, Yang Yang, Yonghui Luo: «Research on the Classification of Aviation Safety Reports Based on Text and Knowledge Graph». – 2020 г.
10. Ximing Li, Bing Wangm, Yang Wang, Meng Wang: «Graph-based Text Classification by Contrastive Learning with Text-level Graph Augmentation». – 2023 г.

11. Partha Basuchowdhuri, Satyaki Sikdar, Varsha Nagarajan, Khusbu Mishra, Surabhi Gupta, Subhashis Majumder: «Fast detection of community structures using graph traversal in social networks». – 2019 г.
12. Roman Sarkar, S M Sojib Ahamed, Md. Arman Hossain: «Possible Causes and Solutions of the Traffic Jam in Dhaka». – 2024 г.
13. Hao Tang, Ling Shao, Nicu Sebe, Luc Van Gool: «Graph Transformer GANs with Graph Masked Modeling for Architectural Layout Generation». – 2024 г.
14. David Liben-Nowell, Jon Kleinberg: «The link-prediction problem for social networks». – 2007 г.
15. James Bennett, Stan Lanning: «The Netflix Prize». – 2007 г.
16. Koppadi Bhavani, Kottu Aslesha, Lakshmi Sai: «Netflix Movies Recommendation System». – 2024 г.
17. Mohammad Rezwanul Huq, Sanjeda Sara Jennifer, Shafiul Mahmud Partho, Fariha Fairuz: «A Comparative Study between Graph Database and Traditional Approach to forecast Coauthor Link Prediction based on Machine Learning Models». – 2022 г.
18. DGL: Heterogeneous Graphs [Электронный ресурс]. – URL: https://docs.dgl.ai/guide/graph-heterogeneous.html#guide-graph-heterogeneous (дата обращения: 13.02.2024).
19. David Liben-nowell, Jon Kleinberg: «The Link Prediction Problem for Social Networks». – 2003 г.
20. Albert-Laszlo Barabasi, Reka Albert: «Emergence of Scaling in Random Networks». – 1999 г.
21. Lada A Adamic, Eytan Adar: «Friends and neighbors on the Web». – 2003 г.
22. Jing Zhou, Shung Jae Shin, Daniel Brass, Jaepil Choi, Zhi-Xue Zhang: «Social Networks, Personal Values, and Creativity: Evidence for Curvilinear and Interaction Effects». – 2009 г.
23. Leo Katz: «A new status index derived from sociometric analysis». – 1953 г.

24. Sergey Brin, Lawrence Page: «The anatomy of a large-scale hypertextual Web search engine». – 1998 г.
25. Glen Jeh Jennifer Widom Scaling: «Personalized Web Search». – 2002 г.
26. Linyuan Lu, Tao Zhou: «Link Prediction in Complex Networks: A Survey». – 2011 г.
27. Yehuda Koren, Robert Bell, Chris Volinsky: «Matrix Factorization techniques for recommender systems». – 2009 г.
28. Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, Alexander J. Smola: «Distributed Large-scale Natural Graph Factorization». – 2013 г.
29. Catalina Cangea, Petar Velickovic, Nikola Jovanovic, Thomas Kipf, Pietro Liò: «Towards Sparse Hierarchical Graph Classifiers». – 2018 г.
30. Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, Wenwu Zhu: «Asymmetric Transitivity Preserving Graph Embedding». – 2016 г.
31. Mikhail Belkin, Partha Niyogi: «Laplacian Eigenmaps for Dimensionality Reduction and Data Representation». – 2002 г.
32. Ulrike von Luxburg: «A Tutorial on Spectral Clustering». – 2007 г.
33. Bryan Perozzi, Rami Al-Rfou, Steven Skiena: «DeepWalk: Online Learning of Social Representations». - 2014 г.
34. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: «Efficient Estimation of Word Representations in Vector Space». – 2013 г.
35. Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, Jie Tang: «Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec». – 2018 г.
36. Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel: «A Three-Way Model for Collective Learning on Multi-Relational Data». – 2011 г.
37. Yuxiao Dong, Nitesh V. Chawla, Ananthram Swami: «metapath2vec: Scalable Representation Learning for Heterogeneous Networks». – 2017 г.

38. Pasquale Lops, Marco de Gemmis, Giovanni Semeraro: «Content-based Recommender Systems: State of the Art and Trends». – 2011 г.
39. Steffen Rendle: «Factorization Machines». – 2010 г.
40. Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Gunnemann, Neil Shah, Meng Jiang: «Graph Data Augmentation for Graph Machine Learning: A Survey». – 2023 г.
41. Thomas N. Kipf, Max Welling: «Variational Graph Auto-Encoders». – 2016 г.
42. Thomas N. Kipf, Max Welling: «Semi-Supervised Classification with Graph Convolutional Networks». – 2017 г.
43. Diederik P. Kingma, Max Welling: «Auto-Encoding Variational Bayes». – 2022 г.
44. Muhan Zhang, Yixin Chen: «Link Prediction Based on Graph Neural Networks». – 2018 г.
45. Rian Dolphin, Barry Smyth, Ruihai Dong: «A Machine Learning Approach to Industry Classification in Financial Markets». – 2023 г.
46. РБК Компании. [Электронный ресурс]. – URL: https://companies.rbc.ru/?utm_source=topline (дата обращения: 03.03.2024)
47. РБК Отрасли. [Электронный ресурс]. – URL: https://www.rbc.ru/industries?utm_source=topline (дата обращения: 03.03.2024).
48. Тинькоф Журнал. [Электронный ресурс]. – URL: https://journal.tinkoff.ru/ (дата обращения: 03.03.2024).
49. Хабр. [Электронный ресурс]. – URL: https://habr.com/ru/feed/ (дата обращения: 03.03.2024).
50. Lenta.ru. [Электронный ресурс]. – URL: https://lenta.ru/ (дата обращения: 03.03.2024).
51. Коммерсантъ. [Электронный ресурс]. – URL: https://www.kommersant.ru/ (дата обращения: 03.03.2024).

52. MFD.ru. [Электронный ресурс]. – URL: https://mfd.ru/about/ (дата обращения: 03.03.2024).
53. Московская биржа. [Электронный ресурс]. – URL: https://www.moex.com/ru/news/ (дата обращения: 03.03.2024).
54. ТАСС. [Электронный ресурс]. – URL: https://tass.ru/ (дата обращения: 03.03.2024).
55. СПАРК-Интерфакс. [Электронный ресурс]. – URL: https://spark-interfax.ru/ (дата обращения: 10.03.2024).
56. Yue Deng: «Recommender systems based on graph embedding techniques: A comprehensive review». – 2022 г.
57. DGL: Message Passing. [Электронный ресурс]. – URL: https://docs.dgl.ai/guide/message.html (дата обращения: 14.04.2024).
58. Selenium with Python. [Электронный ресурс]. – URL: https://selenium-python.readthedocs.io/ (дата обращения: 14.04.2024).
59. Hugging Face. all-MiniLM-L6-v2. [Электронный ресурс]. – URL: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (дата обращения: 24.04.2024).
60. GNNLens2. [Электронный ресурс]. – URL: https://github.com/dmlc/GNNLens2/tree/main (дата обращения: 24.04.2024).
61. Mehdi Ali, Max Berrendorf, Mikhail Galkin, Veronika Thost, Tengfei Ma, Volker Tresp, Jens Lehmann: «Improving Inductive Link Prediction Using Hyper-Relational Facts». – 2021 г.
62. Xiang Song, Runjie Ma, Jiahang Li, Muhan Zhang, David Paul Wipf: «Network In Graph Neural Network». – 2021 г.
63. DGL: Improving Graph Neural Networks Via Network-In-Network Architecture [Электронный ресурс]. – URL: https://www.dgl.ai/blog/2022/11/28/ngnn.html (дата обращения: 24.04.2024).
64. William L. Hamilton, Rex Ying, Jure Leskovec: «Inductive Representation Learning on Large Graphs». – 2018 г.

65. Ziniu Hu, Yuxiao Dong, Kuansan Wang, Yizhou Sun: « Heterogeneous Graph Transformer». – 2020 г.
66. Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, Jie Tang: «Representation Learning for Attributed Multiplex Heterogeneous Network». – 2019 г.
67. Jiaxuan You, Rex Ying, Jure Leskovec: «Position-aware Graph Neural Networks». – 2019 г.

ПРИЛОЖЕНИЕ

GitHub репозиторий выпускной квалификационной работы: https://github.com/meoskis/Diploma