Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ОБУЧЕНИЕ МЕТРИКИ ПОХОЖЕСТИ СООБЩЕСТВ С ПОМОЩЬЮ ВЫДЕЛЕНИЯ ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ

Автор: Зернов Глеб Сергеевич _____

Направление подготовки: 01.03.02 Прикладная математика и информатика

Квалификация: Бакалавр

Руководитель: Сметанников И.Б., к.техн.н. _____

К защите допустить

Руководитель ОП Парфенов В.Г., проф., д.т.н. _____

«____» _____ 20___ г.

Санкт-Петербург, 2019 г.

Студент Зернов Г.С. Группа М3439 Факультет ИТиП

Направленность (профиль), специализация
Математические модели и алгоритмы в разработке программного обеспечения

Консультанты:

а) Попов А.Л., магистр _____

ВКР принята «_____» _____ 20___ г.

Оригинальность ВКР _____%

ВКР выполнена с оценкой _____

Дата защиты «_____» _____ 20___ г.

Секретарь ГЭК Павлова О.Н. _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

УТВЕРЖДАЮ
Руководитель ОП
проф., д.т.н. Парфенов В.Г. _____
«____» _____ 20___ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студент Зернов Г.С. **Группа** М3439 **Факультет** ИТиП

Руководитель Сметанников И.Б., к.техн.н., ассистент, факультет информационных технологий и программирования, Университет ИТМО

1 Наименование темы: Обучение метрики похожести сообществ с помощью выделения векторного представления

Направление подготовки (специальность): 01.03.02 Прикладная математика и информатика

Направленность (профиль): Математические модели и алгоритмы в разработке программного обеспечения

Квалификация: Бакалавр

2 Срок сдачи студентом законченной работы: «____» _____ 20___ г.

3 Техническое задание и исходные данные к работе

Разработать модель, которая позволит представить сообщества социальной сети в виде векторов. Модель необходимо обучить и протестировать на анонимных неразмеченных сессионных данных из социальной сети «Вконтакте». Необходимо проанализировать результаты и сравнить предлагаемое решение с альтернативными методами.

4 Содержание выпускной работы (перечень подлежащих разработке вопросов)

1. Описание предметной области. Обзор существующих подходов.
2. Описание моделей векторного представления сообществ
3. Анализ результатов, сравнение с существующими решениями.

5 Перечень графического материала (с указанием обязательного материала)

Графические материалы и чертежи работой не предусмотрены

6 Исходные материалы и пособия

а) T. Mikolov [и др.] Distributed representations of words and phrases and their compositionality // Advances in neural information processing systems. — 2013. — С. 3111–3119.

б) Grbovic M., Cheng H. Real-time Personalization using Embeddings for Search Ranking at Airbnb // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2018. — С. 311–320.

в) Vasile F., Smirnova E., Conneau A. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation // Proceedings of the 10th ACM Conference on Recommender Systems. — ACM, 2016. — С. 225–232.

7 Дата выдачи задания «____» _____ 20___ г.

Руководитель ВКР _____

Задание принял к исполнению _____

«____» _____ 20___ г.

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студент: Зернов Глеб Сергеевич

Наименование темы ВКР: Обучение метрики похожести сообществ с помощью выделения векторного представления

Наименование организации, в которой выполнена ВКР: Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Получение моделей представления сообществ социальной сети в виде векторов.

2 Задачи, решаемые в ВКР:

а) Обзор существующих подходов.

б) Разработка моделей выделения векторного представления сообществ по сессионным анонимным неразмеченным данным.

в) Анализ полученных результатов.

3 Число источников, использованных при составлении обзора: 22

4 Полное число источников, использованных в работе: 22

5 В том числе источников по годам:

Отечественных			**Иностранных**		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
0	0	0	8	7	7

6 Использование информационных ресурсов Internet: нет

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	**Раздел работы**
Язык программирования `Python`	2.2 - 3.5
Программное средство `Jupyter Notebook`	2.2 - 3.5
Библиотека `Pytorch`	2.2 - 2.4
Библиотека `Sklearn`	3.2, 3.4, 3.5
Библиотека `Matplotlib`	3.5
Библиотека `Spark` и язык программирования `Skala`	3.2

8 Краткая характеристика полученных результатов

Исследованы существующие решения данной задачи. Получены модели, позволяющие представить сообщества социальной сети в виде векторов. Проведено сравнение предложенного решения с существующими путем решения дополнительных задач классификации с использованием результатов работы моделей в качестве входных данных, а так же задачи предсказания следующего действия пользователя. Произведена иллюстрация векторов, полученных в результаты работы обучения одной из предложенных моделей.

9 Гранты, полученные при выполнении работы

При выполнении работы грантов получено не было.

10 Наличие публикаций и выступлений на конференциях по теме работы

Нет

Студент Зернов Г.С. _____

Руководитель Сметанников И.Б. _____

«____» _____ 20__ г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Социальные сети стали неотъемлемой частью современной жизни. Большая часть самых популярных сайтов являются социальными сетями [1], а число их пользователей стремительно растет каждый год. Около 80% всех пользователей сети Интернет — активные пользователи социальных сетей. [2]

Особенностью социальных сетей является огромное число контента, производимого пользователями. Текст, видео, фото, взаимодействия пользователей между собой — все это создает большие объемы информации. Анализ этой информации является сложной, но необходимой для развития платформы, задачей.

Например, для улучшения пользовательского опыта социальные сети используют множество алгоритмов машинного обучения. Персонализированная реклама и рекомендации контента, предложения добавить пользователя в список друзей, анализ пользовательской активности и многое другое. Однако, для обучения моделей необходимы данные об объектах сети. Сложность работы с такой информацией заключается в том, что данных о сущностях социальной сети очень много и использование всей информации не только является сложной инженерной задачей, но и требует огромных трат ресурсов.

Альтернативным способом представления сущностей выступают модели машинного обучения, позволяющие представить объекты в сжатом виде в качестве векторов, которые описывают объект и его свойства.

Алгоритмы векторного представления объектов получили широкое распространение и используются в различных областях, в том числе и в социальных сетях. Например, социальная сеть Pinterest[3] использует модель векторизации объектов социальной сети (Pin'ов)[20], сервис YouTube[4] использует алгоритмы векторного представления для рекомендации видео[2]

В данной работе рассматриваются алгоритмы построения такого представления для сообществ социальной сети. В первой главе описывается постановка задачи и производится обзор существующих способов решения задачи. Во второй главе описывается структура модели, которая позволяет строить векторные представления сообществ по сессионным данным пользователь-

[1] https://www.alexa.com/topsites

[2] https://datareportal.com/reports/digital-2019-global-digital-overview

[3] https://www.pinterest.com

[4] https://youtube.com

ской активности. Рассматриваются три вариации модели: базовая, позволяющая работать с позитивными и негативными редкими откликами пользователя, альтернативная, работающая со слабыми и частыми сигналами и комбинированная, учитывающая оба типа пользовательской активности. В заключительной главе проводятся эксперименты по сравнению предложенных моделей с альтернативными вариантами, а также сравниваются вариации конечной модели между собой путем решения дополнительных задач.

ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

«Вконтакте»[1] — одна из крупнейших социальных сетей СНГ и самая популярная соц. сеть в России. «Вконтакте» на ряду с многими другими социальными сетями предоставляет возможность пользователям создавать объединения по интересам, называемые сообществами. Пользователи имеют возможность подписаться на интересуемое их сообщество для более удобного взаимодействия с информацией сообщества, а также отписаться от него. Сообщества могут добавлять на свою страницу записи (текст, картинки, видео и т. д.), которые пользователи могут помечать как понравившиеся («лайки»), комментировать и прочее.

Существенной сложностью работы с сообществами является число сообществ (миллионы), каждое из которых может состоять из миллионов пользователей.

Таким образом, полная информация о сообществе — это огромный массив данных, предоставляя который полностью некоторой модели машинного обучения, с одной стороны существенно увеличивается размер необходимого объема памяти для хранения этой информации, а с другой — замедляется работа самого алгоритма, что потенциально делает алгоритм бесполезным для применения на практике.

Выходом из этой ситуации является некоторое более емкое представление сообщества без потери интересующей информации в виде вектора.

Такие векторы используются в большом спектре прикладных задач в качестве входных данных моделей машинного обучения. Область применения включает, но не ограничивается, задачами классификации сообществ по некоторым признакам, что позволяет получить более подробное представление о природе сообщества, предсказанием следующего действия пользователя, для рекомендаций сообществ пользователю, кластеризации — сообщества в одном кластере могут быть рекомендованы пользователю, который заинтересован другими сообществами этого кластера, либо для рекламы сообществ кластера пользователям, состоящим в других сообществах этого кластера.

Задача, решаемая в моей работе, — получить модели для выделения универсального векторного представление сообществ небольшой размерности. Полученное векторное представление обязано сохранять семантический

[1] https://vk.com

смысл сообществ, т.е. два сообщества, которые находятся близко в конечном векторном пространстве, должны быть похожи между собой семантически.

1.1. Основные определения

Машинное обучение (*англ. Machine learning*) — класс алгоритмов и статистических моделей. Модели, построенные алгоритмами машинного обучения, способны решать некоторую задачу без явных программных указаний, основываясь на предоставленных модели статистических данных.

Обучение с учителем (*англ. Supervised learning*) — раздел машинного обучения, в котором обучение моделей происходит с помощью размеченного набора данных, т.е. для каждого входного значения сопоставляется правильный «ответ» — выходное значение. Такие модели в процессе обучения анализируют связи между входными и выходными значениями среди всего набора данных и пытаются предсказать ответы по входным значениям.

Обучение без учителя (*англ. Unsupervised learning*) — раздел машинного обучения, в котором обучение моделей происходит без явных выходных сигналов. Такие модели анализируют статистические связи между объектами и пытаются выделить некоторые обобщающие признаки объектов. Например, принадлежность одному кластеру, то есть семантическое сходство.

Унитарный код (*англ. One-hot*) — способ векторного представления объектов. Для такого представления создается список всех объектов, затем объекты нумеруются, и в конце кодируются следующим образом: создается вектор размерности N (N — число уникальных объектов, длина списка), полностью состоящий из нулей, за исключением позиции i, на которой стоит единица. В таком случае i — номер кодируемого объекта в списке.

Косинусное сходство (*англ. Cosin similarity*) — метрика сходства двух векторов, измеряющая косинус угла между ними и задается формулой через скалярное произведение векторов:

$$cosin\_similarity(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

Искусственная нейронная сеть (*англ. Artificial neural network*) — математическая модель, построенная по общим принципам организации биологических нейронных сетей. ИНС состоит из узлов сети, называемых нейронами, соединенных между собой. Каждый узел, получающий сигнал, может пере-

дать его дальше, узлам, с которыми у него есть соединение. В классической реализации ИНН сигналами выступают числа, а способ принятия решения о передаче сигнала — некоторые линейные функции, зависящие от суммы входов нейрона.

Задача классификации — задача сопоставления объектам значений из конечного множества, называемого классами. Разделение объектов на классы является распространненой задачей, которая решается методами машинного обучения.

Метод опорных векторов [1] (*англ. Support vector machine*) — метод машинного обучения для решения ряда задач, в том числе задачи классификации. Является линейным классификатором (существуют, однако, способы обучения нелинейных классификаторов на основе SVM). Принцип работы моделей заключается в перенесении исходных векторов в пространство большей размерности и нахождение оптимальной гиперплоскости (с максимальным расстоянием до представителей) в этом пространстве, разделяющей представителей разных классов.

Дерево решений [19] (*англ. Decision tree*) — дерево, листьями которого являются значения класса объекта, в вершинах записаны условия перехода, а ребрами являются значения, для удовлетворения этого условия. Путь от корня до листа является процессом выбора класса для объекта. В процессе обучения изменяются коэффициенты функций условий переходов.

Бустинг (*англ. Boosting*) — методика машинного обучения, позволяющая объединять несколько слабых классификаторов (например, деревья решений) в один сильный. Существует множество методов бустинга, к примеру градиентный бустинг[8], использующий градиентный спуск для уменьшения функции потери, задаваемой ошибкой классификатора, адаптивный бустинг (более известный как Agaptive Boosting или AdaBoost)[7], создающий веса для каждого классификатора, усиливая, таким образом, влияние лучших классификаторов.

F-мера (*англ. F-measure или* $F_1 score$) — метрика оценки качества модели. Задается формулами

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}$$

$$precision = \frac{true\_positive}{true\_positive + false\_positive}$$

$$recall = \frac{true\_positive}{true\_positive + false\_negative}$$

Где $true\_positive$ — число истинно позитивных результатов, $false\_positive$ — число ложно позитивных результатов, $false\_negative$ — число ложно негативных результатов.

t-SNE (*T-distributed Stochastic Neighbor Embedding*)[15] — алгоритм машинного обучения, позволяющий понизить размерность векторного пространства. При обучении модель пытается приблизить вероятность похожести двух объектов в конечном пространстве к той же вероятности в изначальном пространстве. Таким образом, конечное пространство (чаще всего двухмерное) позволяет визуализировать исходное многомерное пространство с учетом похожести объектов.

Метод k-средних (*англ. k-means*)[12] — алгоритм кластеризации, выделяющий k кластеров и минимизирующий функцию потерь (среднеквадратичное отклонение) от представителей кластера до центра.

1.2. Постановка задачи

Поставленная задача формализуется следующим образом: для заданного множества C сообществ, пользователей N и сессий пользовательской активности A_n, $n \in N$ найти представление $v_c \in \mathbb{R}^d$ размерности d для каждого сообщества $c \in C$ так, что семантически похожие сообщества находятся близко друг к другу в конечном векторном пространстве.

Данная задача решается для 3 случаев представления сессий A_n

В первом случае сессия состоит из множества событий двух типов:

$$A_n = (S_n, U_n) \tag{1}$$

Где события S_n — список последовательных K уникальных подписок на сообщества пользователя n за некоторый промежуток времени: $S_n = (c_1, c_2, \ldots, c_K), \forall i \neq j : c_i \neq c_j$, а U_n — список последовательных M уникальных отписок от сообществ пользователя n: $U_n = (c_1, c_2, \ldots, c_M), \forall i \neq j : c_i \neq c_j$, $S_n \cap U_n = \varnothing$.

Во втором случае сессия задается как множество K (не обязательно уникальных) действий («лайков») пользователя n записей сообществ:

$$A_n = (c_1, c_2, \ldots, c_K) \tag{2}$$

В третьем случае каждой сессии задается тип T_n пользовательских событий, представленных в ней $T_n = \{subs, likes\}$ — подписки/отписки, либо лайки. В таком случае сессия задается в зависимости от типа:

$$A_n = \begin{cases} (S_n, U_n), & \text{if } T_n = subs \\ (c_1, c_2, \ldots, c_K), & \text{if } T_n = likes \end{cases} \tag{3}$$

И определяется формулой (1) для первого типа, формулой (2) для второго.

1.3. Обзор существующих решений

1.3.1. Алгоритм факторизации матриц в рекомендательных системах.

Персонализация пользовательского контента является одной из основополагающих частей систем, предоставляющих такой контент конечным пользователям. Лидеры интернет-торговли или интернет-услуг особенно заинтересованы в таких алгоритмах, поскольку это позволяет увеличить пользовательский отклик. В общем случае, рекомендательные системы разделяются на две категории по принципу работы.

Первой категорией является фильтрация контента (*англ. content filtering*)[13]. Главная идея состоит в том, чтобы создать профиль для каждого пользователя или продукта с целью охарактеризовать сущность. Например, в качестве профиля фильма могут выступать жанр, актеры, кассовые сборы и т.д.

Альтернативный вариант, коллаборативная фильтрация (*англ. collaborative filtering*), опирается на действия пользователей, сделанных в прошлом. Такие алгоритмы анализируют статистические связи между пользователями, продуктами, их взаимодействиями и дают рекомендации, основываясь на исторических данных. Большим плюсом таких систем является независимость от домена, в котором они применяются.

Одним из самых распространенных на данный момент времени алгоритмов в сфере рекомендательных систем является алгоритм факторизации матриц [11], относящийся к группе алгоритмов коллаборативной фильтрации.

Модель матричной факторизации переносит пользователей и продукты в пространство признаков размерности n. Таким образом, продукт p представим в виде вектора $a_p \in \mathbb{R}^n$, а пользователь u в виде вектора $b_u \in \mathbb{R}^n$. Для продукта каждый из элементов вектора отвечает за то, насколько каждый признак характерен для этого продукта. В то время для пользователя элементы вектора показывают, насколько пользователю интерес признак.

Это позволяет задать оценку пользователя u продукта p в виде формулы

$$r_{ui} = a_i^T b_u$$

Таким образом, зная матрицу пользовательских оценок R, можно разложить ее на произведение матриц A и B, отвечающих за векторное произведение продуктов и пользователей соответственно.

$$R = A^T B$$

В рамках задачи, поставленной формулой(1), матрица оценок составляется следующим образом: для подписки пользователя i на сообщество j выставляется единица на эту позицию $R_{ij} = 1$, аналогично для отписки $R_{ij} = -1$. В случае отсутствия действий пользователя для сообщества оценка будет равна нулю.

Адаптируя алгоритм ALS [11] для представления сессий формулы (2), можно представить оценку пользователя сообществу, как число «лайков», которые он поставил этому сообществу за сессию.

Для работы с комбинированными данными (формула 3) эти два подхода объединяются. Конечная матрица составляется следующим образом: если сессия имеет тип $subs$, данные представляются, так, как описано в первом варианте (для подписок), иначе, как во втором. Поскольку подписка более важное событие, чем лайк, вводится коэффициент для подписок q, на который будут умножаться события из сессий типа $subs$.

1.3.2. Обзор алгоритма LDA

Latent Dirichlet allocation (LDA) [14] — вероятностная модель над корпусом документов. Базовая идея заключается в том, что документы представимы в виде набора тем, где каждая тема характеризуется распределением по словам. Данная модель позволяет выделять несколько тем для документа и рассчитывать вероятности отношения каждой из тем к документу. В качестве продукта работы алгоритма LDA позволяет строить вероятностные распределения тем по документу, слов по темам и темам по словам.

Например, для такой модели будет верно, что слово «кот» имеет большую вероятность того, что оно относится к теме, интерпретируемую наблюдателем как «животные» (имеет большие вероятности для слов, обозначающих животных), чем к другим темам.

Стоит отметить, что несмотря на тот факт, что модель LDA была впервые применена в области NLP, это не ограничивает использование алгоритма в других доменах. Так, в оригинальной статье [14] приводится пример использования для коллаборативного фильтрования. Кроме того, модель может быть использована для поиска групп в графах [10], для обнаружения классов объектов на изображениях, представленных в трехмерном виде (обучение без учителя) [21] и прочее.

В рамках данной работы (для задач, определяемых формулами (1) и (2)) алгоритм LDA применим следующим образом: сессии представляются в виде документов, где словами являются подписки (лайки во втором случае) на сообщества. Применив алгоритм на корпусе сессий, получается список тем, а также вероятность принадлежности сообщества к теме для всех сообществ. Так составляется вектор вероятностей для каждого сообщества, которым можно охарактеризовать любое сообщество из всего словаря. Можно интерпретировать такое представление сообществ как то, что похожие сообщества будут иметь похожие вероятностные распределения тем.

Для задачи, задаваемой формулой (3), аналогично пункту 1.3.1 задается коэффициент q, на который будут умножаться события-подписки в модели (т.е. число слов в документе)

Важным является тот факт, что модель не будет учитывать сигналы отписок от сообщества, и это является недостатком применения данного алгоритма.

1.4. Модели естественного языка

Для множества алгоритмов обработки естественного языка (Natural Language Processing, NLP) некие специализированные цели алгоритма могут быть обобщены на задачу нахождения значений вероятностей для последовательностей слов. Таким образом, развитие алгоритмов обработки естественного языка шло путем выявления статистических свойств слов и нахождения зависимостей между ними.

Традиционно такие подходы представляли каждое слово в виде one-hot вектора, однако недостатком такого представления является наложение ограничений на практическое применение алгоритмов, в связи с большой размерностью объектов и разряженности данных, приводящих к ощутимой потери производительности.

Исследователями были предложены модели на основе нейронных сетей[18], которые позволяют решить эти проблемы путем представления слов в виде векторов гораздо меньшей размерности. Такие модели основываются на гипотезе о том, что близкие друг к другу слова в предложениях статистически более зависимы.

Исторически неэффективность обучения моделей нейронных сетей была главным препятствием на пути к применению таких алгоритмов на практике, поскольку словарь может достигать миллионов слов. Однако предложенные в [5] и [3] модели (в том числе skip-gram модель, на которой основан алгоритм 2.1) оказались хорошо масштабируемыми и могут быть успешно использованы на практике. Эксперименты, приводимые в статьях, показали, что модели способы выявлять синтаксические и семантические связи между словами в больших корпусах слов.

Универсальность такого представления объектов вышла за рамки задач естественного языка и была применена в областях, с ним не связанных. Так были предложены алгоритмы векторного представления товаров [4][22], событий бронирования отелей[9], для использования в рекомендательных системах, рекомендаций музыкальных композиций[6]. Были показаны различные способы использования таких моделей в рекомендательных системах [17]. Также был предложен алгоритм персонализированных рекомендаций пользователю [16]

Выводы по главе 1

В данной главе была описана предметная область, для которой проводились исследования. Даны общие определения понятий, испольщующихся в данной работе. Была обозначена формальная постановка задачи, основанная на трех способах представления сессионных взаимодействий пользователя с сообществами. Были рассмотрены существующие решения выделения векторного представления объектов, и отмечены их недостатки для решения поставленных задач. В пункте 1.4 был рассмотрен подход, применяющийся в области обработки естественного языка и проведен обзор переноса такого подхода в другие области.

ГЛАВА 2. ОПИСАНИЕ МОДЕЛЕЙ ВЫДЕЛЕНИЯ ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ СООБЩЕСТВ

В данной главе будет рассмотрена модель, позволяющая выделять векторы сущностей, информация о которых представлена в виде сессии взаимодействия пользователей с этими сущностями. Также будет рассмотрено обучение модели на примере сессионных данных пользователей «Вконтакте».

Будут рассмотрены вариации модели для данных пользователей по подпискам, затем по данным отметок «мне нравится» (или «лайкам»), и в конце по комбинированным данным.

2.1. Описание модели

Для применения базовой skip-gram [3] модели (см. пункт 1.4) необходимо облегчить постановку задачи пункта 1.2. Для этого сессия A_n представляется как множество подписок пользователя S_n, т.е. $A_n = S_n$. В терминах NLP «документами» будут выступать сессии, а «словами» — действия пользователя.

Для построения векторов ставится побочная цель: для заданного сообщества c_i предсказать вероятности появления сообществ $c_j, k \in V, V = |C|$ в контексте данного сообщества. Контекстом сообщества называется w его соседей в сессии.

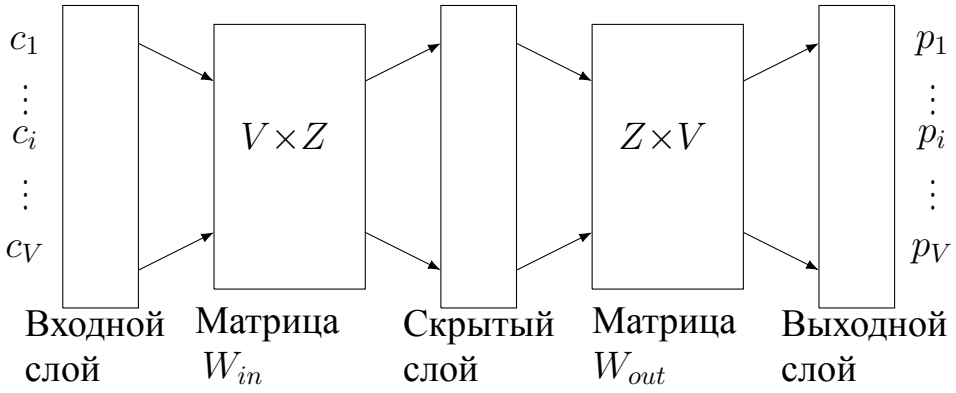

Рисунок 1 – Архитектура модели skip-gram

Для решения задачи. поставленной таким способом, строится нейронная сеть с одним скрытым слоем, который соединяется с входным слоем, на который подается one-hot вектор, представляющий сообщества и выходным слоем, выдающим вероятностное распределение по сообществам, входной и выходной матрицами W_{in}, W_{out}. Параметр Z отвечающий за размерность матриц, является гиперпараметром модели. (см. рис. 1)

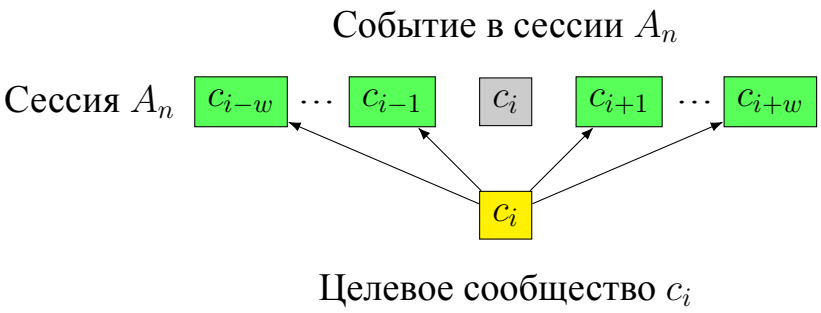

Рисунок 2 – skip-gram модель

Целью модели skip-gram является максимизировать функцию:

$$L = \sum_{A \in D} L_A \tag{4}$$

$$L_A = \sum_{c \in A} P_c \tag{5}$$

$$P_c = \sum_{-w \leqslant j \leqslant w,\, j \neq 0} \log \mathbb{P}(c_{i+j}|c_i) \tag{6}$$

Где D обозначает множество всех сессий, i — индекс целевого сообщества c в сессии A. Вероятность $\mathbb{P}(c_{i+j}|c_i)$ наблюдать сообщество c_{i+j} в сессии рядом с заданным сообществом c_i задается функцией soft-max

$$\mathbb{P}(c_{i+j}|c_i) = \frac{\exp(v_{c_i}^{\top} v'_{c_{i+j}})}{\sum_{c=1}^{V} \exp(v_{c_i}^{\top} v'_c)} \tag{7}$$

Где $v_c \in W_{in}$ и $v_c{}' \in W_{out}$ входное и выходное векторные представления сообщества c, параметр w — длина контекста сообществ, V — общее число сообществ.

Максимизация функции осуществляется стохастическим градиентным спуском, обновляя значения входной и выходной матриц. Конечный результат находится во входной матрице W_{in}, которая представляет из себя векторы сообществ c_k, т. е. в строке k содержится векторное представления сообщества c_k.

Из (4)-(7) можно сделать следующий вывод: обученная модель будет размещать сообщества в векторном пространстве так, что сообщества, встречающиеся в похожих контекстах (имеют похожие соседние сообщества в сессиях) будут иметь схожие векторы.

2.2. Модель обучения по подпискам

Далее было рассмотрено представление сессий, задаваемое функцией (1), и проведен ряд модификаций базовой модели для работы с такими сущностями. Кроме того, формула (5) была представлена в виде

$$L_A = \sum_{c \in S \in A} P_c \tag{8}$$

Поскольку отписки являются отрицательными откликами пользователя и не должны учитываться в качестве целевых сообществ.

2.2.1. Модификации функции подсчета вероятности

Максимизация вероятности, задаваемой функцией (7), имеет существенный недостаток, а именно — сложность вычисления ее градиента, которая обуславливается большим числом операций (пропорциональных размеру словаря, т.е. количеству сообществ C. Число сообществ в социальной сети исчисляется десятками миллионов).

Альтернативой soft-max является подход негативного семплирования (negative sampling) [3], позволяющий существенно снизить вычислительную сложность расчета вероятности, аппроксимируя результирующую величину.

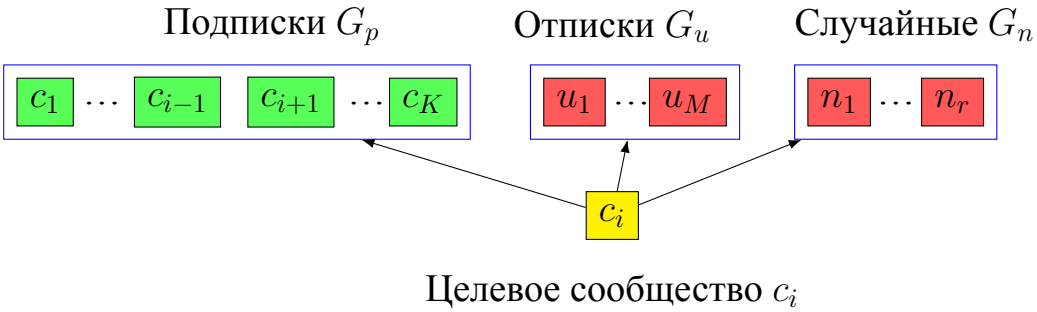

Рисунок 3 – skip-gram модель для подписок

Так, оптимизируемая функция может быть представлена в виде формулы:

$$P_c = \sum_{p \in G_p} \log \frac{1}{1 + \exp(-v'_p v_c)} + \sum_{u \in G_u} \log \frac{1}{1 + \exp(v'_u v_c)} + \sum_{n \in G_n} \log \frac{1}{1 + \exp(v'_n v_c)} \tag{9}$$

Где c — целевое сообщество сессии a, $c \in S_a$, G_p — множество позитивных событий контекста (подписки на сообщества), G_u — множество негативных событий контекста (отписки от сообщества) и равно множеству всех отписок сессии a, $G_u = U_a$, G_n — множество случайных негативных событий, не входящих в G_p и G_u, число элементов множества задается гиперпараметром r модели, $|G_n| = r$. Задача модели — максимизировать данную функцию на каждом шаге обучения, повышая, таким образом, вероятность появления в контексте целевого сообщества для позитивных событий и понижая для негативных. Оптимизация функции осуществляется стохастическим градиентным спуском.

Экспериментально было проверено, что игнорирование отписок либо неиспользование случайных событий ведет к потере точности модели. Эксперименты описаны более подробно в главе 3

2.2.2. Представление данных для модели

Важным фактором для обучения модели являются способы передачи данных модели. Для каждой сессии a согласно формуле 8 составляются пакеты, состоящие из целевого сообщества c, его контекста G_p и негативных сигналов. Целевые сообщества получаются последовательным перебором каждого сообщества из сессии, получение негативных сигналов было описано в предыдущем пункте, а способы получения контекста будут описаны ниже. Таким образом, для каждой сессии получается список пакетов (длина списка равна длине сессии), которые передаются в модель для обучения.

Первый подход, описанный в пункте 2.1, — это представить контекст в виде соседей целевого сообщества.

$$G_p = \{c_{i+j} \in S, -w \leqslant j \leqslant w, j \neq 0\} \tag{10}$$

Где S — подписки сессии, i — индекс целевого сообщества c в S_n, $i \leqslant w \leqslant |S_n| - w$ Данный способ показывает себя довольно плохо на дан-

ных по подпискам, поскольку подписка достаточно редкое явление и является сильным сигналом, при том часть сигналов (отстоящих на $> w$ сессий от целевой) будет игнорироваться.

Подробнее влияние представления данных для модели описано в главе 3, основываясь на экспериментах.

Стоит отметить, что средняя длина сессии для подписок не велика (в связи с природой таких действий: пользователи не подписываются на сообщества слишком часто), в таком случае, можно применить альтернативные подходы, позволяющие учитывать подписки, которые были бы проигнорированы в первом случае. Кроме того, стоит принимать во внимание факт, что подписки являются долгосрочным интересом пользователя, поэтому даже далеко отстоящие по времени действия пользователя в сессии могут быть связаны между собой.

Так, можно передавать все подписки из сессии в модель, за исключением целевого сообщества. (см. рис. 3)

$$G_p = \{c_j \in s, j \neq i\} \tag{11}$$

Отдельно стоит отметить, что при таком подходе оптимизируемая функция P_c (9) зависит от длины сессии, поэтому ее следует нормализовать. Таким образом, конечная функция, оптимизируемая моделью, будет выглядеть следующим образом:

$$P_c^{'} = \frac{P_c}{|G_p| + |G_u| + |G_n|} \tag{12}$$

2.3. Модель обучения по слабым сигналам

Модель для работы с данными по «лайкам», где сессии описываются формулой (2), модифицируется следующим образом: в качестве базы берется модель, описанная в предыдущих пунктах. Поскольку явные негативные пользовательских сигналы отсутствуют в сессиях, формула (9) будет переписана в виде:

$$P_c = \sum_{p \in G_p} \log \frac{1}{1 + \exp(-v_p^{'} v_c)} + \sum_{n \in G_n} \log \frac{1}{1 + \exp(v_n^{'} v_c)} \tag{13}$$

Где c — целевое сообщество, G_p — список позитивных событий контекста (лайки записей сообществ), G_n — множество случайных негативных событий, не входящих в G_p.

Несмотря на то, что G_p для предоставленного набора данных можно представлять в виде (11), более гибким способом будет использование формулы (10), поскольку лайки являются гораздо более частым событием и отвечают за краткосрочный интерес пользователя, что ведет к несвязности далеких по времени событий. Кроме того, для одинакового временного промежутка количество лайков будет существенно больше числа подписок и представление в виде (11) не позволит эффективно работать модели по обоим типам данных за одинаковый промежуток времени.

Конечную функцию оптимизации P_c можно не нормализовать, поскольку $|G_p|$ и $|G_u|$ фиксированы.

2.4. Модель обучения по комбинированным данным

В конце была рассмотрена еще одна версия модели, которая способна обрабатывать оба типа событий: как подписки, так и лайки. Сессии для такой задачи описываются формулой (3). Простым способом будет варьировать оптимизируемую функцию в зависимости от типа контекста.

Вероятностная функция сессий L_a будет зависеть от типа сессии, и записывается в виде (8) для типа $subs$ и в виде (5) для $likes$

Аналогичным образом задается P_c — используется (12) для $subs$, а для $likes$ (13) — уже нормализованная. Методы получения необходимых данных на шаге обучения остаются такими же. Необходимо учитывать, что подписка является гораздо более важным сигналом, поэтому предлагается усилить влияние таких событий умножением на задаваемый коэффициент, который следует подбирать в зависимости от соотношения лайков к подпискам в исходных данных. В итоге получаются следующие функции оптимизации на шаге обучения.

Для подписок:

$$P_c^{''} = qP_c^{'}$$

Где $P_c^{'}$ задается формулой (12), а q является дополнительным гиперпараметром модели, который отвечает за силу сигналов-подписок.

Для лайков:

$$P_c^{'} = \frac{P_c}{|G_p| + |G_n|}$$

Где P_c задается формулой (13)

Полученные функции подставляются в формулы для L_a

Выводы по главе 2

В этой главе были рассмотрены модели векторного представления сообществ. Были рассмотрены 3 модели: для работы с сессионными данными пользователей по подпискам 2.2, по лайкам 2.3 и смешанным 2.4.

Были обучены модели и получены матрицы векторного представления сообществ для разных типов данных.

Полученные векторы имеют малую размерность (которую можно задать в качестве параметра), что делает применение векторов в качестве входных данных какой-либо другой модели более удобным.

При построении векторов учитывается сессионная и коллаборативная информация. Таким образом, полученный результат будет учитывать как сходство сообществ в похожих контекстах, так и давность данных (предпочтения пользователя могут меняться со временем).

Кроме того, входные данные для модели не требуют дополнительной ручной разметки. Все, что нужно модели для работы, — пользовательские действия, которые могут быть никак не обработаны. Таким образом, использование модели в реальной системе является крайне простым.

ГЛАВА 3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В качестве дополнительной задачи позволяющей понять, что полученные векторы действительно несут полезную информацию (правильным образом представляют положение сообществ в пространстве) было решено несколько подзадач описанных в 3.2 — 3.4, а затем полученные результаты были сравнены с альтернативными алгоритмами, описанными в первой главе.

В главе 3.5 приводятся иллюстрации полученного векторного пространства, отображенного в двухмерный вид.

3.1. Формат входных данных

Компанией были предоставлены анонимные сессионные действия пользователей за небольшой промежуток времени. Всего было предоставлено 3 набора данных.

Первый набор данных состоит из списка сессий, каждая сессия — это список действий некоторого пользователя, элементами списка является идентификатор сообщества (id) и маркер действия (подписка/отписка). Данные были собраны по действиям пользователей за 2-3 месяца.

Также были предоставлены сессионные данные по «лайкам» пользователя в аналогичном формате: список сессий, каждая сессия является списком id сообществ, запись которого понравились пользователю. Как было отмечено ранее, в отличие от данных по подпискам, в сессии по лайкам сообщество может встречаться несколько раз. Данные были собраны по действиям пользователей за 2-3 дня. Таким образом, средняя длина сессий второго набора данных примерно равна средней длины сессий из первого набора данных.

Третий набор данных является комбинацией из первых двух, каждая сессия имеет тип того, какая информация в ней содержится: действия пользователя по подпискам или лайкам.

После фильтрации данных общее число сессий было сокращено до 500 000 (450 000 сессий в первом наборе данных), число уникальных групп до 11 200 для каждого набора данных. В смешанных данных число сессий по подпискам и лайкам было одинаково и случайно распределено по всему массиву данных. Средняя длина сессий всех наборов данных примерно равна 4.5

Данные были разбиты на два множества, тренировочное и тестовое, по первому происходило обучение классификаторов, по второму проводились

вычисления итоговой F-меры. Разбиение на тестовое и тренировочное множество производилось случайным образом. Всего было проведено 5 разбиений с обучением классификатора на этих данных. Во всех таблицах указаны усредненные значения посчитанной F-меры на 5 запусках обучения классификатора.

3.2. Обучение моделей классификации сообществ

Сообщества «Вконтакте» имеют категории, задаваемые администраторами (например, спорт, СМИ и т. д.). Так, можно решить задачу классификации, используя векторы, полученные в ходе работы моделей, в качестве входных данных классификатора. Для обучения классификатора использовались дополнительные размеченные данные, которые содержат идентификатор сообщества и идентификатор его категории. Категории бывают двух типов: общие (всего 50 типов таких категорий) и подробные (более 250). Для проведения эксперимента были выбраны 5 (для экспериментов, проводимых в пункте 3.2.1) и 10 (для экспериментов, проводимых в пункте 3.2.2) самых встречаемых категорий среди сообществ, и отфильтрованы сообщества, не относящиеся к этим категориям.

В качестве классификаторов были использованы алгоритмы, рассмотренные в пункте 1.1 с реализацией библиотекой sklearn (Gradient Boosting Classifier[1], Ada Boost Classifier[2], Linear Support Vector Classifier[3], SGDClassifier[4]), для каждой модели был выбран лучший вариант.

3.2.1. Сравнение модификаций модели, обучаемой по подпискам

В ходе разработки было рассмотрено несколько вариантов модификации оптимизируемой функции, представления данных для модели, исследовано влияние подбираемых гиперпараметров на качество модели. Обучение производилось на небольшой части данных, 45000 сессий и 1200 уникальных сообществ. Был использован фреймворк Pytorch для построения архитектуры модели и для расчета градиента с последующей итерацией обучения.

Сначала был проведен эксперимент по изменению функции для оптимизации. Была попробована стандартная функция soft-max (формула (7)), функ-

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

ция, задаваемая формулой (9) и упрощенные варианты формулы (9):

$$P_c = \sum_{p \in G_p} \log \frac{1}{1 + \exp(-v'_p v_c)} + \sum_{u \in G_u} \log \frac{1}{1 + \exp(v'_u v_c)} \tag{14}$$

$$P_c = \sum_{p \in G_p} \log \frac{1}{1 + \exp(-v'_p v_c)} + \sum_{n \in G_n} \log \frac{1}{1 + \exp(v'_n v_c)} \tag{15}$$

Без учета отписок (14) и случайных (15) событий соответственно.

Число негативных событий для тех формул, где они используются, было равно 5. Размерность конечного пространства была равна 64.

Таблица 1 – Классификация общих категорий (различные оптимизируемые функции)

Размер окна w	Soft-max	Формула 9	Формула 14	Формула 15
1	0.526 ± 0.016	0.518 ± 0.024	0.504 ± 0.017	0.416 ± 0.035
2	0.542 ± 0.020	0.533 ± 0.025	0.517 ± 0.021	0.453 ± 0.032
3	$\mathbf{0.554 \pm 0.023}$	$\mathbf{0.551 \pm 0.019}$	0.520 ± 0.022	0.475 ± 0.036

Результаты эксперимента, указанные в таблице 1 показали, что при использовании формулы (15) чрезвычайно сильное влияние оказывает начальное состояние векторов (случайное) и модель в процессе обучения не получает данных о том, что сообщества, вероятно, не похожи друг на друга (не встречаются в одинаковых контекстах), а для формулы (14) не получает явного пользовательского отклика о негативных событиях. При этом, использование предлагаемой формулы (9) вместо формулы soft-max не ухудшает качество векторов, особенно с увеличением окна. Как было отмечено ранее, использование soft-max несет следующий недостаток: вычисление формулы зависит от числа уникальных сообществ, в то время как формула (9) от этого числа не зависит, что делает ее удобнее для применения на практике, позволяя считать функцию оптимизации быстрее.

Были рассмотрены два варианта подачи данных, используя формулу (10), т.е. используя окно и формулу (11), передавая все подписки целиком.

Эксперимент показал (см. табл. 2), что увеличение окна повышает качество векторов, а использование всех подписок в качестве контекста дает наилучший результат.

Таблица 2 – Классификация общих категорий (способы представления данных)

Способ передачи данных	Формула 10, $w = 1$	Формула 10, $w = 2$	Формула 10, $w = 3$	Формула 11
F-мера	0.518 ± 0.024	0.533 ± 0.025	0.551 ± 0.019	$\mathbf{0.582 \pm 0.031}$

Также было исследование влияние гиперпараметров на модели.

Таблица 3 – Классификация общих категорий (зависимость от числа случайных событий r)

r	F-мера
1	0.517 ± 0.026
2	0.517 ± 0.031
4	0.530 ± 0.026
6	0.540 ± 0.027
8	0.569 ± 0.026
9	$\mathbf{0.582 \pm 0.031}$
10	0.562 ± 0.015
12	0.542 ± 0.022
16	0.538 ± 0.024
20	0.505 ± 0.012

Результаты, указанные в таблице 3, показали, что эффективность модели растет с числом случайных событий, но начинает угасать после 9. Это связано с тем, что увеличивается вероятность попадания в случайные события положительных откликов. Поэтому число случайных событий для моделей было установлено в 9.

Кроме того, было исследование влияние размерности d конечного векторного пространства на модель. Для этого было запущено обучение модели с разными значениями d, но константным числом итераций.

Таблица 4 – Классификация общих категорий (способы представления данных)

d	F-мера
32	0.536 ± 0.045
48	0.556 ± 0.030
64	$\mathbf{0.576 \pm 0.038}$
80	0.551 ± 0.028
96	0.535 ± 0.024

Эксперимент показал (см. таблицу 4), что несмотря на то, что увеличение размерности пространства повышает качество векторов, это так же влечет увеличение времени обучения модели. Поэтому для дальнейших экспериментов было выбрано значение параметра d равное 64 как баланс между качеством и скоростью.

Последнее исследование влияния гиперпараметров было посвящено значению q, введенного в пункте 2.3. Этот параметр отвечает за приоритет подписок над лайками при обучении модели на комбинированных данных.

Таблица 5 – Классификация общих категорий (зависимость от параметра q)

q	F-мера
1	0.620 ± 0.011
1.5	0.634 ± 0.031
2	$\mathbf{0.663 \pm 0.028}$
2.5	0.622 ± 0.036
3	0.617 ± 0.033
3.5	0.602 ± 0.028
4	0.615 ± 0.023

В результате (см. табл. 5) было проверено, что выбор значения q действительно влияет на качество векторов и оптимальное значение коэффициента для набора данных примерно равно 2. Стоит отметить, что при увеличении параметра больше 2 модель начинает терять эффективность. Это объясняется переобучением на данных по подпискам и недостаточным обучением на данных по лайкам, в то время как малый параметр q не регулирует в достаточной мере приоритет подписок над лайками.

3.2.2. Сравнение с альтернативными вариантами

Были рассмотрены 4 варианта представления сообществ в виде вектора для подачи классификатору. Первым способом является представление сообщества в виде one-hot вектора размерности $|A|$ (количество всех сессий), результаты обучения классификатора на таких векторах указаны в столбце «raw». Остальные три способа — это обучение предложенной модели и альтернативных вариантов. Была использована библиотека spark и реализации ал-

горитмов ALS[5] и LDA[6] Для алгоритмов ALS и LDA обучение строится таким же путем, как было указано в пунктах 1.3.1 1.3.2.

Усредненные результаты работы классификаторов, обученных на разных векторах, указаны в таблицах: в ячейках указаны значения F-меры посчитанной для заданного класса. Последняя строчка содержит среднее значение F-меры по всем классам, учитывая частоту, с которой они встречаются в наборе данных. Категории в таблицах отсортированы по возрастанию частоты встречаемости в данных. Таблицы с результатами приведены в приложении.

В таблицах А.1, А.2 указаны результаты обучения классификатора по векторам, полученных моделью из пункта 2.2. В таблицах А.3, А.4 указаны результаты по векторам, полученных моделью из пункта 2.3. В таблицах А.5, А.6 указаны результаты по векторам, полученных моделью из пункта 2.4.

По таблицам А.1 и А.2 видно, что предложенная модель работает лучше альтернативных способов, обученных на данных по подпискам.

Результаты, указанные в таблицах А.3 и А.4 показывают, что предложенная модель работает лучше альтернативных способов, обученных на данных по слабым сигналам (лайкам).

По таблицам А.5 и А.6 можно заключить, что предложенная модель работает для нескольких типов сигналов и классификатор, обученный на таких векторах, показывает хорошие результаты как для общих, так и подробных категорий.

Эксперимент показал, что различные типы сигналов, на которых обучается модель, влияют на результаты классификации. Стоит отметить, что некоторые категории учитываются лучше моделью, обученной по подпискам, другие — моделью, обученной по лайкам, при этом разрыв между результатами моделей может быть достаточно большим. Однако модель, обученная по обоим типам данных, не сильно уступает лучшему результату и в общем показывает лучшие показатели.

Важным результатом является гибкость классификаторов, обученных на полученных моделями векторах. Так, классификатор распознает хорошо не только самые популярные категории, но и менее распространенные, однако ярко выраженные, такие как «спорт» и «еда».

[5]https://spark.apache.org/docs/2.2.0/api/scala/index.html#org.apache.spark.ml.recommendation.ALS

[6]https://spark.apache.org/docs/latest/ml-clustering.html#latent-dirichlet-allocation-lda

3.3. Предсказание следующего действия

Пользуясь полученными векторами можно попытаться предсказать следующее действие пользователя по его истории. Такой способ часто используется в современных рекомендательных системах. Однако стоит отметить, что задача, поставленная таким способом крайне сложная и требует дополнительной информации. Обычно для решения такой задачи необходимы сложные модели и дополнительные данные о пользователе.

Поэтому ставится следующая задача: требуется найти 50 наилучших кандидатов следующего действия пользователя по его предыдущим действиям. Для такой задачи считается процент попадания настоящего действия в список из предложенных кандидатов. Задача решается следующим способом: вычисляется сумма векторов сообществ из истории, и производится поиск ближайших векторов к полученному, используя метрику косинусоного сходства. Далее полученные вектора сортируются по убыванию значения метрики, и берутся первые 50 кандидатов.

Данные для обучения были составлены следующим образом: были взяты случайные сессии средней длины (т.е. длина сессии должна быть >4). Далее сессии были разбиты на две части, последнее действие пользователя (которое требуется предсказать) и предыдущие, выступающие в качестве исторических данных. Из сессий были исключены негативные события, а также исключены сессии, в которых сообщество последнего действия пользователя содержится в исторических данных. В конце были взяты 1000 случайных сессий для проведения эксперимента.

В таблице 6 указаны результаты предсказания следующего действия, основанного на векторах по разным наборам данных и полученные разными моделями. В ячейках таблицы указано отношение числа попаданий настоящего действия в 50 кандидатов к числу проведенных опытов.

Таблица 6 – Предсказание следующего действия

	Только подписки	Только лайки	Подписки + лайки
ALS	0.092	0.079	0.078
LDA	0.079	0.075	0.112
Предложенная модель	**0.296**	**0.275**	**0.304**

Результаты, представленные в таблице 6 показывают, что предлагаемые модели способны извлекать информацию об отношении сделанных пользова-

телям действий, и конечное пространство содержит информацию о взаимодействии пользователей с сообществами.

Следовательно, полученные моделями векторы могут быть использованы в прочих алгоритмах машинного обучения в области рекомендательных систем.

3.4. Предсказание геолокации

Сообщества социальной сети могут быть посвящены некоторому месту в населенном пункте, в таком случае сообщество содержит информацию о местонахождении. Таким образом, в некоторых случаях сообщество может иметь метку о геолокации. Предполагается, что полученные моделями векторы будут сохранять такую информацию, и близкие сообщества будут относиться к одному городу.

Для решения такой задачи были предоставлены дополнительные данные следующего вида: имеется список информации о паре сообществ. Каждый элемент списка представляет из себя идентификаторы сообществ и бинарный флаг, выставленный в 1, если сообщества имеют одну геолокацию, и 0 иначе. Кроме того, сообщества могут и вовсе не иметь локации. Данные были разбиты на два множества, тренировочное и тестовое. Сообщества, встречающиеся в тренировочном множестве, не встречаются в тестовом. Размеры множеств равны 70 000 и 55 000 соответственно.

Требуется для пары сообществ определить, имеют ли два сообщества одну геолокацию. Для этого были использованы классификаторы, указанные в пункте 3.2. Аналогичным образом выбирался лучший классификатор, результаты работы классификатора, обученного на разных векторах, указаны в таблице (значения F-меры).

Для обучения классификатора данные подавались в следующем виде для каждой пары сообществ был составлен характеризующий вектор. Для этого последовательно были выписаны значения вектора первого сообщества, значения вектора второго сообщества и косинусное сходство этих векторов. Таким образом характеризующий вектор хранил информацию о положении каждого сообщества в паре и расстояние между ними.

По таблице 7 видно, что информация, содержащаяся в подписках пользователей, сильно влияет на определение геолокации, в то время как лайки пользователей слабо влияют на это признак. Кроме того было показано, что

Таблица 7 – Визуализация векторного пространства

	Только подписки	Только лайки	Подписки + лайки
ALS	0.613	0.315	0.459
LDA	0.707	0.123	0.595
Предложенная модель	**0.819**	0.696	0.711

предложенная модель позволяет лучше предсказывать геолокацию сообществ, чем альтернативные способы.

Кроме того, было замечено, что несмотря на возможность обучения классификатора на векторах, полученных не только на данных по подпискам, но и другими способами (лайки или смешанные) (см. таблицу 7), на том же объеме данных классификатор для таких векторов обучается гораздо хуже, особенно на векторах, построенных альтернативными алгоритмами.

Из этого делается следующий вывод: векторы, построенные на подписках, имеют сильный геолокационный признак и позволяют выделять геолокацию сообществ. В то же время, векторы, полученные с учетом лайков, не позволяют классифицировать геолокацию с той же точностью и в общем случае теряют эту информацию, ухудшая, таким образом, результаты и для векторов по смешанным данным.

3.5. Кластеризация полученных векторов

В предыдущих пунктах была обучена модель 2.4 на полном объеме данных. Полученное векторное пространство можно отобразить на двухмерное пространство для визуализации результатов. Для этого был применен алгоритм t-SNE. На изображении также отмечены векторы сообществ, относящихся к общим категориям «Спорт», «Авто и мото», используя размеченные данные из пункта 3.2 (см. рис. 4)

Полученное изображение наглядно показывает, что обученная модель действительно получает семантически похожие векторы. Так, сообщества с одинаковыми категориями находятся рядом друг с другом. Интересно, что различные сообщества, интересующие мужскую аудиторию социальной сети, находятся достаточно близко друг к другу. Кроме того, на рисунке можно заметить несколько небольших скоплений сообщества с категорией «Спорт», относящиеся к разным видам спорта.

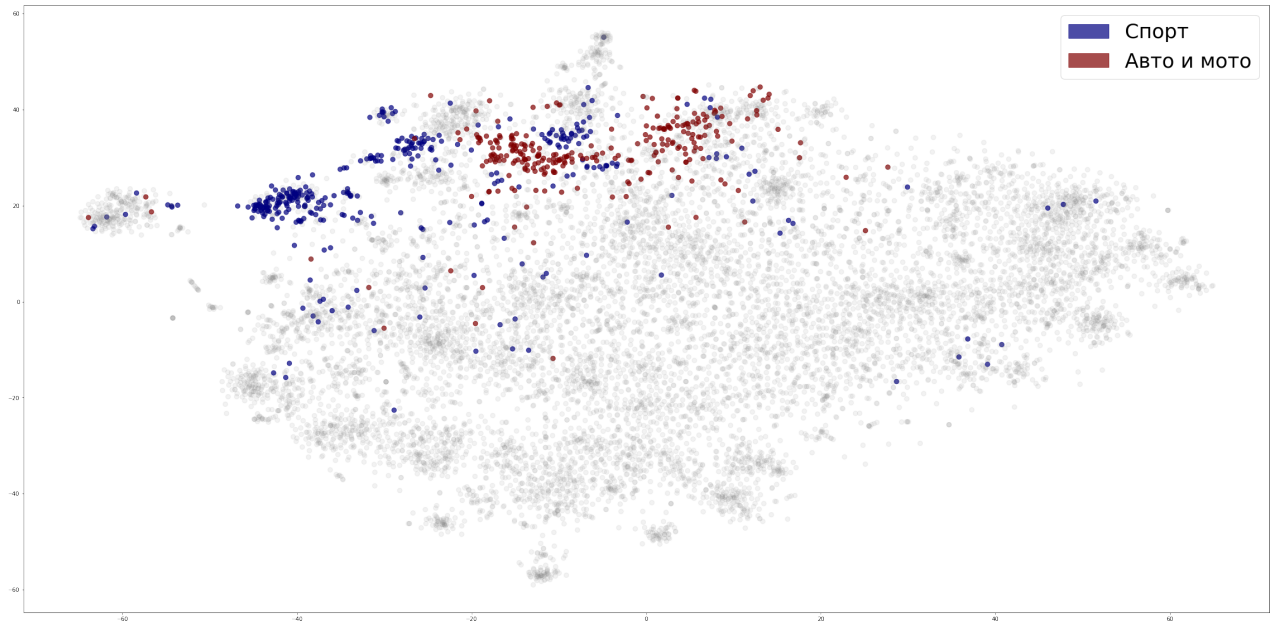

Рисунок 4 – Визуализация векторного пространства (с отметками категорий)

Дополнительно был обучен алгоритм кластеризации k-means на результирующих многомерных векторах. Было проведено разбиение на 32 кластера, мною была проведена ручная разметка нескольких из них. (см. рис. 5)

На рисунке видно, что модель сохраняет важные свойства сообществ. К примеру, в социальной сети существует большое число пользователей из Казахстана, которые создают сообщества, записи в которых ведутся на казахском языке. Стоит отметить, что кластер таких сообществ заметно удален от общей картины.

Возрастные категории пользователей сообществ также учитываются, сообщества тематик, интересующих пользователей школьного возраста расположены рядом с сообществами подростковых тематик, а так же рядом с кластером компьютерных игр. Интересно, что сообщества, которые интересуют юношей, находятся в некотором отдалении от сообществ, привлекающих аудиторию девушек. В то же время оба кластера находятся в близости к кластеру школьных тематик.

Отдельно стоит отметить кластер специфических, но популярных, интересов пользователей, которым является кластер с корейской поп-музыкой (K-pop). Кластер таких сообществ стоит в некотором отдалении от всех остальных, и достаточно близко к сообществам для молодых девушек. Интересно,

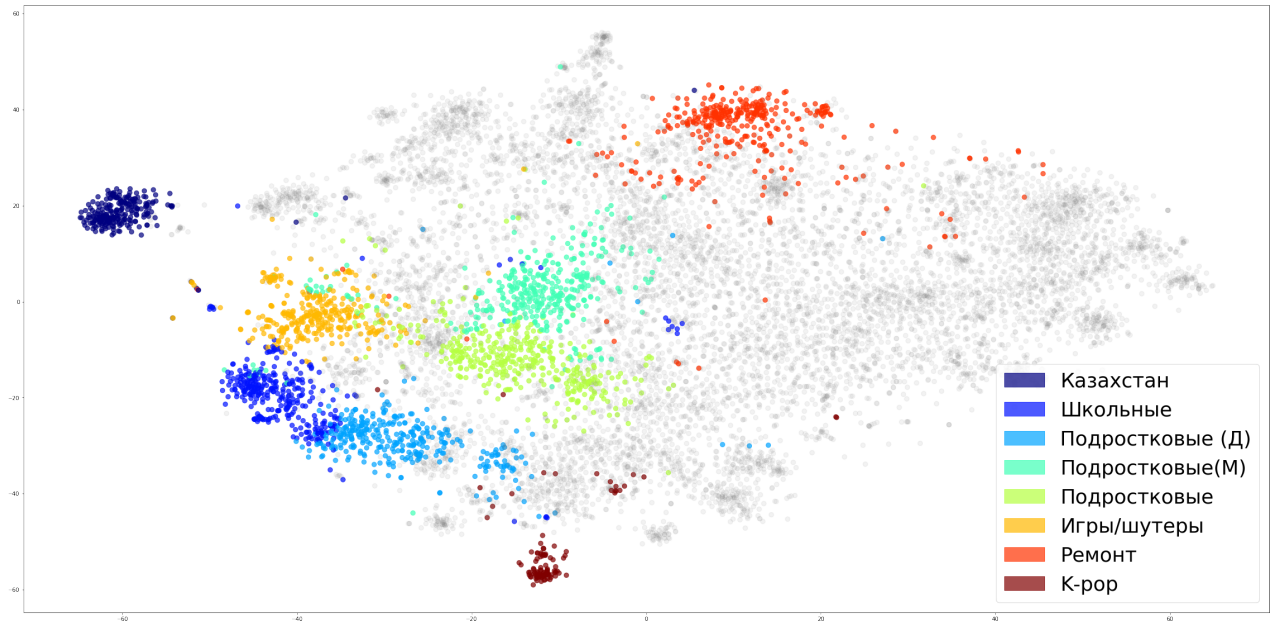

Рисунок 5 – Визуализация векторного пространства (с отметками кластеров)

что согласно интернет-опросам, основная демография слушателей корейской музыки — молодые девушки. [7]

Сообщества, относящиеся к ремонту, лежат на карте близко с сообществам спортивной и авто тематик (см. рис. 4), показывая популярность таких сообществ у мужской части пользователей социальной сети.

Выводы по главе 3

Эксперименты показали, что полученные вектора действительно отвечают главному требованию — они сохраняют семантический смысл сущностей (сообщества социальной сети). Было показано, что предлагаемая модель работает лучше, чем альтернативные варианты. Дополнительным преимуществом модели является меньшая требовательность к ресурсам оборудования, на котором она обучается.

Использование полученных векторов в моделях машинного обучения для решения спектра различных задач показало, что полученные векторы могут быть применены на практике в других алгоритмах. Кроме того, гибкость моделей и возможность работать с различными типами данных позволяет использовать конкретную модель получения векторов, подстраиваясь под условия задачи. Эксперимент показал, что различные вариации модели позволяют лучше решать задачи, в зависимости от их контекста.

[7] https://www.kpopmap.com/age-and-gender-demographics-of-kpop-girl-group-listeners-in-south-korea-2018/

Кроме того, различные модели показывают себя лучше в различных ситуациях (векторы, полученные с помощью модели, обученной по комбинированным данным, показывают себя лучше в классификации категорий, в то время как векторы, полученные по подпискам, подходят лучше для классификации геолокации). Таким образом, в зависимости от конечной задачи и типа интересующих сигналов, могут быть использованы различные предложенные модели.

ЗАКЛЮЧЕНИЕ

В данной работе былм предложены модели выделения векторного представления сообществ на основе сессионных действий пользователя. Было предложено 3 вариации конечной модели, для работы с подписками и отписками пользователей (п. 2.2), лайками пользователей (п. 2.3) и комбинированный подход, учитывающий оба типа событий (п. 2.4)

Экспериментально было показано, что векторы, полученные в результате обучения моделей, могут быть применены для решения других задач, таких как классификация сообществ по признакам (категория или геолокация), предсказание следующего действия пользователя (может быть использовано в рекомендательных системах), кластеризация сообществ (может быть использовано для рекламы сообществ).

3 вариации модели показывают гибкость, и, как показывают эксперименты, разные вариации модели могут быть успешно использованы в различных задачах.

В дальнейшем модель, обучающаяся по слабым событиям, может быть легко расширена на другой тип сигналов с похожими свойствами (например, клики пользователя на сообщества). Также такие сигналы могут быть интегрированы в модель по комбинированным данным, увеличивая спектр пользовательских сигналов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Cortes C.*, *Vapnik V.* Support Vector Network // Machine Learning. — 1995. — P. 273–297.

2 *Covington P.*, *Adams J.*, *Sargin E.* Deep Neural Networks for YouTube Recommendations // Proceedings of the 10th ACM Conference on Recommender Systems. — 2016. — P. 191–198.

3 Distributed representations of words and phrases and their compositionality / T. Mikolov [et al.] // Advances in neural information processing systems. — 2013. — P. 3111–3119.

4 E-commerce in your inbox: Product recommendations at scale / M. Grbovic [et al.] // Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2015. — P. 1809–1818.

5 Efficient estimation of word representations in vector space / T. Mikolov [et al.] // arXiv preprint arXiv:1301.3781. — 2013.

6 Exploiting Music Play Sequence for Music Recommendation. / Z. Cheng [et al.] // IJCAI. Vol. 17. — 2017. — P. 3654–3660.

7 *Freund Y.*, *Schapire R. E.* A decision-theoretic generalization of on-line learning and an application to boosting // Journal of computer and system sciences. — 1997. — P. 119–139.

8 *Friedman J. H.* Stochastic gradient boosting // Computational statistics & data analysis. — 2002. — P. 367–378.

9 *Grbovic M.*, *Cheng H.* Real-time Personalization using Embeddings for Search Ranking at Airbnb // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2018. — P. 311–320.

10 *Henderson K.*, *Eliassi-Rad T.* Applying Latent Dirichlet Allocation to Group Discovery in Large Graphs // Proceedings of the 2009 ACM Symposium on Applied Computing. — 2009. — P. 1456–1461.

11 *Koren Y.*, *Bell R.*, *Volinsky C.* Matrix Factorization Techniques for Recommender Systems // Computer. — 2009. — Vol. 42. — P. 30–37.

12 *Lloyd S.* Least squares quantization in PCM // IEEE transactions on information theory. — 1982. — P. 129–137.

13 *Lops P.*, *Gemmis M. de*, *Semeraro G.* Content-based Recommender Systems: State of the Art and Trends //. — 2011. — P. 73–105.

14 *M. Blei D.*, *Y. Ng A.*, *Jordan M.* Latent Dirichlet Allocation // Journal of Machine Learning Research. — 2003. — Vol. 3. — P. 993–1022.

15 *Maaten L. v. d.*, *Hinton G.* Visualizing data using t-SNE // Journal of machine learning research. — 2008. — P. 2579–2605.

16 *Manotumruksa J.*, *Macdonald C.*, *Ounis I.* Modelling user preferences using word embeddings for context-aware venue recommendation // arXiv preprint arXiv:1606.07828. — 2016.

17 *Ozsoy M. G.* From word embeddings to item recommendation // arXiv preprint arXiv:1601.01356. — 2016.

18 *P. Turian J.*, *Ratinov L.-A.*, *Bengio Y.* Word Representations: A Simple and General Method for Semi-Supervised Learning. // Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. — 2010. — P. 384–394.

19 *Quinlan J. R.* Induction of decision trees // Machine learning. — 1986. — P. 81–106.

20 Related Pins at Pinterest: The Evolution of a Real-World Recommender System / D. C. Liu [et al.] // Proceedings of the 26th International Conference on World Wide Web Companion. — 2017. — P. 583–592.

21 Unsupervised discovery of object classes from range data using latent Dirichlet allocation / F. Endres [et al.] // Robotics: Science and Systems. — 2009.

22 *Vasile F.*, *Smirnova E.*, *Conneau A.* Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation // Proceedings of the 10th ACM Conference on Recommender Systems. — ACM, 2016. — P. 225–232.

ПРИЛОЖЕНИЕ А. РЕЗУЛЬТАТЫ КЛАССИФИКАЦИИ КАТЕГОРИЙ СООБЩЕСТВ

Таблица А.1 – Классификация общих категорий (данные по подпискам)

Категория	Предложенная модель	Raw data	ALS	LDA
Спорт	0.667 ± 0.043	0.209 ± 0.054	0.481 ± 0.041	0.553 ± 0.090
Городское сообщество	0.231 ± 0.036	0.085 ± 0.018	0.111 ± 0.022	0.336 ± 0.023
Музыка	0.296 ± 0.041	0.131 ± 0.030	0.082 ± 0.023	0.022 ± 0.014
Сообщество по интересам	0.078 ± 0.026	0.153 ± 0.017	0.054 ± 0.011	0.119 ± 0.033
Образование	0.319 ± 0.033	0.136 ± 0.031	0.263 ± 0.028	0.232 ± 0.026
Семья	0.322 ± 0.027	0.147 ± 0.020	0.043 ± 0.015	0.184 ± 0.041
Известная личность	0.315 ± 0.025	0.139 ± 0.025	0.182 ± 0.037	0.185 ± 0.032
Здоровье и красота	0.522 ± 0.019	0.232 ± 0.028	0.309 ± 0.046	0.374 ± 0.028
Культурное общество	0.400 ± 0.018	0.282 ± 0.010	0.339 ± 0.014	0.380 ± 0.017
Развлечения	0.616 ± 0.017	0.492 ± 0.015	0.590 ± 0.010	0.578 ± 0.013
Средн. взвеш.	$\mathbf{0.452 \pm 0.012}$	0.301 ± 0.009	0.356 ± 0.010	0.391 ± 0.014

Таблица А.2 – Классификация подробных категорий (данные по подпискам)

Категория	Предложенная модель	Raw data	ALS	LDA
Отношения полов	0.284 ± 0.043	0.116 ± 0.015	0.106 ± 0.056	0.222 ± 0.041
Молодежная организация	0.410 ± 0.068	0.155 ± 0.037	0.336 ± 0.030	0.323 ± 0.054
Рецепты и еда	0.782 ± 0.030	0.306 ± 0.056	0.628 ± 0.027	0.685 ± 0.052
Городское сообщество	0.540 ± 0.022	0.110 ± 0.015	0.249 ± 0.016	0.500 ± 0.026
Кино	0.577 ± 0.050	0.195 ± 0.024	0.463 ± 0.026	0.349 ± 0.049
Фотография	0.352 ± 0.016	0.137 ± 0.022	0.327 ± 0.040	0.301 ± 0.017
Образование	0.566 ± 0.018	0.194 ± 0.025	0.434 ± 0.041	0.388 ± 0.012
Литература	0.463 ± 0.039	0.157 ± 0.030	0.416 ± 0.028	0.333 ± 0.026
Творчество	0.292 ± 0.022	0.190 ± 0.039	0.170 ± 0.029	0.298 ± 0.033
Юмор	0.716 ± 0.017	0.458 ± 0.012	0.659 ± 0.028	0.644 ± 0.017
Средн. взвеш.	$\mathbf{0.548 \pm 0.010}$	0.266 ± 0.010	0.453 ± 0.014	0.456 ± 0.007

Таблица А.3 – Классификация общих категорий (данные по лайкам)

Категория	Предложенная модель	Raw data	ALS	LDA
Музыка	0.152 ± 0.039	0.062 ± 0.015	0.005 ± 0.011	0.020 ± 0.010
Известная личность	0.093 ± 0.050	0.042 ± 0.014	0.047 ± 0.048	0.088 ± 0.025
Образование	0.055 ± 0.019	0.032 ± 0.029	0.010 ± 0.013	0.044 ± 0.020
Городское сообщество	0.225 ± 0.035	0.099 ± 0.025	0.089 ± 0.042	0.066 ± 0.033
Спорт	0.612 ± 0.026	0.153 ± 0.029	0.226 ± 0.034	0.488 ± 0.028
Сообщество по интересам	0.221 ± 0.021	0.074 ± 0.022	0.044 ± 0.017	0.163 ± 0.033
Здоровье и красота	0.411 ± 0.022	0.055 ± 0.007	0.098 ± 0.017	0.172 ± 0.033
Семья	0.373 ± 0.044	0.045 ± 0.020	0.092 ± 0.036	0.186 ± 0.010
Культурное общество	0.439 ± 0.006	0.085 ± 0.016	0.282 ± 0.025	0.348 ± 0.029
Развлечения	0.648 ± 0.011	0.530 ± 0.009	0.565 ± 0.009	0.575 ± 0.009
Средн. взвеш.	$\mathbf{0.459 \pm 0.007}$	0.242 ± 0.010	0.307 ± 0.008	0.358 ± 0.013

Таблица А.4 – Классификация подробных категорий (данные по лайкам)

Категория	Предложенная модель	Raw data	ALS	LDA
Отношения полов	0.382 ± 0.047	0.114 ± 0.037	0.115 ± 0.059	0.150 ± 0.009
Рецепты и еда	0.796 ± 0.016	0.163 ± 0.027	0.401 ± 0.056	0.438 ± 0.069
Молодежная организация	0.540 ± 0.064	0.088 ± 0.021	0.120 ± 0.023	0.286 ± 0.045
Кино	0.618 ± 0.014	0.133 ± 0.030	0.072 ± 0.029	0.226 ± 0.032
Городское сообщество	0.479 ± 0.045	0.109 ± 0.018	0.337 ± 0.034	0.202 ± 0.043
Образование	0.303 ± 0.034	0.078 ± 0.021	0.054 ± 0.024	0.115 ± 0.031
Фотография	0.444 ± 0.029	0.067 ± 0.018	0.054 ± 0.027	0.217 ± 0.038
Творчество	0.393 ± 0.038	0.113 ± 0.041	0.146 ± 0.014	0.227 ± 0.040
Литература	0.467 ± 0.022	0.161 ± 0.039	0.195 ± 0.029	0.324 ± 0.024
Юмор	0.749 ± 0.012	0.482 ± 0.020	0.596 ± 0.014	0.595 ± 0.016
Средн. взвеш.	$\mathbf{0.575 \pm 0.013}$	0.247 ± 0.014	0.322 ± 0.007	0.370 ± 0.016

Таблица А.5 – Классификация общих категорий (комбинированные данные)

Категория	Предложенная модель	Raw data	ALS	LDA
Музыка	0.340 ± 0.046	0.066 ± 0.018	0.075 ± 0.046	0.075 ± 0.028
Городское сообщество	0.269 ± 0.078	0.098 ± 0.040	0.091 ± 0.030	0.455 ± 0.053
Образование	0.252 ± 0.054	0.121 ± 0.029	0.173 ± 0.041	0.183 ± 0.034
Спорт	0.726 ± 0.033	0.195 ± 0.042	0.375 ± 0.048	0.601 ± 0.034
Известная личность	0.293 ± 0.043	0.150 ± 0.025	0.154 ± 0.026	0.284 ± 0.017
Сообщество по интересам	0.123 ± 0.053	0.075 ± 0.031	0.156 ± 0.014	0.242 ± 0.039
Семья	0.418 ± 0.030	0.168 ± 0.016	0.194 ± 0.031	0.291 ± 0.026
Здоровье и красота	0.571 ± 0.031	0.225 ± 0.027	0.295 ± 0.031	0.393 ± 0.039
Культурное общество	0.468 ± 0.019	0.282 ± 0.025	0.346 ± 0.013	0.397 ± 0.020
Развлечения	0.673 ± 0.009	0.499 ± 0.004	0.586 ± 0.010	0.621 ± 0.010
Средн. взвеш.	$\mathbf{0.511 \pm 0.012}$	0.303 ± 0.004	0.373 ± 0.012	0.449 ± 0.009

Таблица А.6 – Классификация подробных категорий (комбинированные данные)

Категория	Предложенная модель	Raw data	ALS	LDA
Отношения полов	0.298 ± 0.025	0.084 ± 0.033	0.124 ± 0.045	0.247 ± 0.035
Молодежная организация	0.477 ± 0.019	0.128 ± 0.029	0.301 ± 0.034	0.412 ± 0.047
Городское сообщество	0.525 ± 0.014	0.074 ± 0.019	0.316 ± 0.052	0.583 ± 0.026
Рецепты и еда	0.834 ± 0.041	0.213 ± 0.040	0.646 ± 0.023	0.687 ± 0.042
Кино	0.650 ± 0.042	0.180 ± 0.027	0.362 ± 0.020	0.376 ± 0.092
Фотография	0.438 ± 0.029	0.141 ± 0.034	0.302 ± 0.024	0.329 ± 0.036
Образование	0.535 ± 0.059	0.169 ± 0.021	0.332 ± 0.030	0.326 ± 0.041
Творчество	0.407 ± 0.058	0.147 ± 0.036	0.220 ± 0.028	0.337 ± 0.050
Литература	0.464 ± 0.036	0.172 ± 0.023	0.476 ± 0.023	0.467 ± 0.031
Юмор	0.759 ± 0.012	0.510 ± 0.016	0.664 ± 0.016	0.680 ± 0.009
Средн. взвеш.	$\mathbf{0.599 \pm 0.007}$	0.281 ± 0.009	0.460 ± 0.011	0.511 ± 0.013