МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

КАФЕДРА АНАЛИЗА ДАННЫХ

Квалификационная работа на соискание степени магистра по направлению 03.04.01 «Прикладные математика и физика» на тему:

РЕШЕНИЕ ПРОБЛЕМЫ ХОЛОДНОГО СТАРТА ПРИ ПОСТРОЕНИИ ИНДИВИДУАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ТРАЕКТОРИИ С ПОМОЩЬЮ ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Студент группы М05-893а — Павловская Анастасия Сергеевна

Научный руководитель д.ф-м.н. — Воронцов Константин Вячеславович

Москва, 2020

Содержание

1 Введение

Работа посвящена решению проблемы холодного старта в рекомендательной системе с использованием тематических моделей.

Традиционные рекомендательные системы строятся на основе истории взаимодействия пользователя с контентом, однако в сфере образования такая стратегия не является оптимальной: система, опирающаяся исключительно на исторические данные, навязывает контент по тем темам, которые пользователь уже изучил. Детально эта проблема и пути её решения описаны в разделе 1.

Объекты рекомендательной системы – пользователи и образовательный контент – представлены текстами и дополнительной мета-информацией, подробное их описание находится в первой части раздела 2. Взаимодействие между объектами не ограничиваются связью *«пользователь изучил контент»*, но включает и другие отношения, описанию которых посвящена вторая часть раздела 2. В работе предлагается представить все данные в виде графа, в котором вершины представляют собой объекты системы и описываются текстами на естественном языке, а рёбра – отношения между ними.

Система рекомендаций в такой задаче будет строится в два этапа: на первом – этапе обучения – будут построены векторные представления вершин графа, а на втором – этапе непосредственной рекомендации – будет осуществлён поиск ближайших к эмбеддингу[1] пользователя векторов образовательного контента. Данная работа посвещена первому этапу создания рекомендательной системы: построению таких векторных представлений вершин графа, которые будут учитывать как текстовую информацию-описание вершин, так и связи между ними.

Задача построения векторных представлений вершин графа первоначально появилась в контексте анализа взаимодействий пользователей в социальных сетях. Однако с распространением графов в задачах моделирования различных систем, появилась необходимость агрегировать в эмбеддингах дополнительную информацию о вершинах. Кроме того, в последнее время большое распространение получили *графы знаний*, в которых отображены типы взаимодействий между вершинами. Всё это привело к появлению методов построения векторных представлений вершин, которые учитывают разнородную информацию об объектах и разные виды взаимодействий между ними. Обзору существующих алгоритмов посвящён раздел 3.

Данные можно представлять не только как граф, но и как коллекцию связанных между собой документов. Тогда можно предположить, что вхождение каждого слова в документ, описывающий пользователя или контент, связанно с некоторой латентной переменной, которую в контексте задачи будем называть *темой* или *интересом*. Задача выявления тематики коллекции решается с помощью вероятностного тематического моделирования. Дискретное распределение вероятностей тем в документе будет использоваться в качестве векторного представления документа. Построение тематической модели подробно описано в разделе 4.

Для оценки качества решения задачи предлагается использовать метрики, оценивающие способность модели учитывать как графовую структуру данных, так и доступные текстовые описания. Предложенный для этого набор метрик описан в разделе 5. Раздел 6 данной работы посвящён описанию проведённых вычислительных экспериментов, и сравнению предложенного метода с бейзлайн решением. В выводах сформулированы основные результаты работы, описаны преимущества и недостатки используемого метода.

[1] Термин, обозначающий векторное представление

2 Мотивация

Остановимся подробнее на прикладной области, в которой поставлена данная задача. Работа строится на данных платформы дистанционного обучения, предоставляющей доступ к курсам, офф- и онлайн-мероприятиям, коучам, практическим проектам, другим участникам с интересными вам компетенциями.

С точки зрения рекомендаций у каждого вида контента есть своя специфика: курсы, как правило, предполагают большое количество новой информации, на освоение которой понадобится большое количество времени. Оффлайн-мероприятие – это не только источник знаний, но и инструмент увеличения сети знакомств, поэтому в процессе рекомендаций нужно учитывать не только текстовое описание, но и приглашённых участников. Совсем другой подход нужен для рекомендации коуча: он должен опираться не только на описание коуча, но и на то, какие отзывы оставляют о нём другие участники. Объединяет все эти виды контента общее свойство: все они задаются текстами на естественном языке.

Как и в традиционных рекомендательных системах[1], известна история взаимодействия пользователя с образовательным контентом: мы знаем, насколько успешно он проходил обучение на курсах, с какими людьми посещал мероприятия, какие роли играл в проектах. Эту цепочку последовательных взаимодействий будем называть *образовательной траекторией*. Однако система, обученная исключительно на историю человека, не обладает необходимой гибкостью: она навязывает пользователю продолжать определённую образовательную траекторию, не учитывая его целей.

В данной системе возникает и другая проблема, характерная для рекомендательных систем – проблема «холодного старта»: как дать рекомендацию абсолютно новому пользователю? [2] Как привести человека к нужной образовательной траектории, не имея истории его активности на платформе? Чтобы решить эту проблему, пользователя при регистрации просят заполнить вводную анкету, в которой он указывает интересующие области [3]. Несовершенство такого подхода состоит в том, что пользователь может не знать, какие конкретные темы ему нужно изучить для достижения цели.

Таким образом, традиционные рекомендательные системы сталкиваются с двумя проблемами: «холодный старт» и неспособность корректировать траекторию человека в процессе взаимодействия пользователя с платформой.

Для решения этих проблем предлагается добавить в модель знания об *образовательной цели* человека. На первых этапах, когда пользователь только пришёл на платформу, эта информация поможет системе подобрать актуальную рекомендацию, построить подходящую образовательную траекторию. Уточнение актуальной формулировки цели во время взаимодействия человека с платформой позволит скорректировать построенную траекторию так, чтобы она не потеряла своей актуальности для пользователя.

От того, насколько полно описана цель, зависит качество рекомендации. Как сформулировать исчерпывающее описание образовательной цели? Ответ на этот вопрос даёт одна из самых распространённых методик целеполагания – SMART. Согласно ей, правильно сформулированная цель должна удовлетворять следующим критериям:

1. Specific (конкретность): Что именно необходимо достичь?
2. Measurable (измеримость): В чём именно будет измеряться результат?
3. Achievable (достижимость): Возможно ли достичь цели? И за счёт чего?
4. Relevant (уместность): Соответствует ли цель жизненным задачам и ориентирам?
5. Time-bound (ограниченность во времени): Когда должна быть выполнена цель?

С учётом этой методики была разработана анкета, в результате заполнения которой каждый пользователь генерирует набор текстов на естественном языке: желаемый результат, первые шаги, которые можно предпринять, примерное время, необходимое для достижения цели[2].

Несмотря на то, что вопросы анкеты поставлены конкретно, пользователи, не знакомые с аппаратом целеполагания, испытывают затруднения с ответами на них, поэтому часто отвечают неоднозначно, расплывчато, формулировки их целей требуют уточнений. Автоматизировать процесс детализации цели предлагается с помощью чат-бота, задающего человеку конкретизирующие вопросы. Чтобы построить такого помощника, необходимо в каждый момент разговора с пользователем понимать, насколько поставленная цель соответствует критериям целеполагания, какую информацию следует уточнить, какой вопрос лучше задать пользователю. Для решения этой задачи была собрана экспертная разметка, которая дополняет анкеты пользователей, оценивает надёжность предоставленной ими информации и формирует обучающую выборку для решения задачи ведения диалога с пользователем.

Лексика, которая используется пользователем при постановке цели, сильно отличается от той, с помощью которой описан контент. Эту проблему решает добавление таксономии, содержащей структурную информацию и позволяющей связывать цели и единицы контента. Более полное описание этой структуры будет дано в разделе 3.3. Как и экспертная разметка, разметка по таксономии доступна лишь для части целей и курсов, поэтому одной из второстепенных задач является предсказание оценок экспертов и связей с таксономией для новых пользователей и единиц контента.

Подводя итог, задача состоит в создании рекомендательной системы, способной рекомендовать контент разного вида и адаптивно меняться в зависимости от предпочтений человека. Для решения этой задачи предлагается обогатить модель с помощью анкет, в которых описаны цели пользователей. Построение такой системы – это вопрос целой серии исследований, данная работа будет посвящена решению проблемы «холодного старта» при помощи построения векторных представлений объектов системы.

[2]Более подробное она описана в разделе 3.2

3 Структура данных

3.1 Образовательный контент

Коллекции образовательных ресурсов на платформе будут пополняться, но в рамках данной работы предлагается остановиться на двух видах контента: курсах и мероприятиях.

Курсы Платформа представляет собой агрегатор образовательных курсов: на ней не только собраны лучшие курсы различных учебных заведений и технологических компаний, но и существует возможность получить финансирование на обучение на других площадках. Так как платформа пока не генерирует собственные курсы, то в данных недоступна информация обо всех используемых материалах, и модель должна опираться лишь на текстовые описания.

В таблице 1 представлена полная информация о текстах, доступных для каждого учебного материала. Кроме этого, каждому курсу соответствуют метаданные, описание которых представлено в 2.

Название поля	**Описание поля**
Название курса	Короткий текст на русском или английском языке. Медианная длина совпадает со средней длиной – 4 слова.
Описание курса	Текст на русском или английском языке в свободной форме. Средняя длина – 112 слов, медианная длина – 81 слово.
Программа курса	Разной степени детализации текстовое описание тем, освещённых в курсе. Как правило, в описаниях есть определённая структура. Средняя длина – 260 слов, медианная длина – 90 слов. Доступно для 70% выборки.
Требования к слушателям	Короткие тексты на естественном языке, описывающие пререквизиты курса и для кого этот курс предназначен. В качестве требований может служить умение работать с определённой технологией (*«понимание WEB»*), специализация человека (*«курс рассчитан на ручных тестировщиков»*) или наличие опыта в сфере (*«требуется иметь опыт программирования»*). В этом поле возможны упоминания других курсов, необходимых для изучения (*«рекомендуем к прохождению первый модуль курса Python»*). Средняя длина – 20 слов, медианная – 14.
Результат обучения	Текст на естественном языке, в основном состоящий из терминов, описаний приобретаемых навыков и названий технологий, своеобразное резюме описания курса. Средняя длина – 34 слова, медианная – 26 слов.
Автор курса	Небольшая биографическая справка автора курса на естественном языке. В описании, как правило, упоминаются основные проекты, в которых автор участвовал, его карьерные достижения. В этом разделе часто упоминаются и другие курсы этого автора. Средняя длина – 82 слова, медианная – 51 слов.

Таблица 1: Текстовая информация о курсе

Название поля	**Тип данных**	**Описание поля**
Название организации	Строка (15 уникальных значений)	Название организации, предоставляющей курс.
Ссылка на курс	Строка , HTML-ссылка	Ссылка на страницу курса на платформе организации, предоставляющей курс.
Формат курса	Строка («онлайн», «оффлайн», «смешанное»)	Подавляющее большинство курсов проводится в формате онлайн.
Длительность курса в неделях	Дробное число, точность до половины недели	В среднем на изучение курса требуется 8 недель, самый длинный курс занимает 43 недели.
Стоимость курса	Целое число	Стоимость в рублях.
Администратор курса	Целое число	Контакт человека, который занимается администрированием курса в виде id пользователя платформы. Не обязательно должен быть тем же человеком, что и автор курса.
Планируемое количество потраченных часов	Строка	Это поле несёт информацию о сложности курса, а не о его длительности. Описан коротким текстом, содержащим оценку в часах в неделю.
Формальная дата начала курса	Строка	Некоторые курсы (особенно оффлайн) подразумевают фиксированную дату начала.

Таблица 2: Метаданные о курсе

Офф- и онлайн-мероприятия Кроме курсов, платформа является площадкой для организации и проведения мероприятий. На ней можно как подготовить оффлайн встречу, воспользовавшись площадкой в «Точке кипения», так и провести онлайн собрание.

Как и курсы, мероприятия описываются с помощью текстов 3 и дополнительных метаданных 4.

Название поля	**Описание поля**
Название мероприятия	Текстовое описание в свободной форме. Средняя длина – 12 слов, медианная длина – 10 слов.
Полное описание	Текстовое описание в свободной форме. Средняя длина – 110 слов, медианная длина – 80 слов. Самый длинный текст состоит из 2869 слов.
Короткое описание	Краткое резюме полного описания, у части мероприятий совпадает с первыми предложениями полного описания. Средняя и медианная длины совпадают – 17 слов.

Таблица 3: Текстовая информация о мероприятии

Название поля	**Тип данных**	**Описание поля**
Тип мероприятия	Строка (19 уникальных типов)	Это поле отражает специфику мероприятия: например, лекция предполагает одного активного рассказчика, круглый стол – совместное обсуждение, а конкурс – состязание между отдельными участниками или командами.
Время начала и конца мероприятия	DateTime	По этим полям можно вычислить продолжительность мерпориятия. Она колеблется от нескольких часов до нескольких дней, однако большая часть (91% актуальной выборки) – короткие мероприятия, длительностью меньше дня.
Страна мероприятия	Строка (9 уникальных значений)	В большинстве своём мероприятия проводятся на территории РФ, однако есть и такие страны, как Сингапур, США, Южная Корея.
Город мероприятия	Строка (102 уникальных значения)	Половина мероприятий организована на территории Москвы.
Организатор мероприятия	Целое число	Контакт главного организатора в виде id пользователя платформы.

Таблица 4: Дополнительные данные о мероприятии

3.2 Данные о пользователе

Данная информация поступает в систему через анкету или при заполнении пользователем личного профиля и представляет собой две части: описание образовательной цели 5 и социально-демографические характеристики 6.

Кроме всей той информации о цели, которую даёт пользователь, для части данных известна разметка экспертов в целеполагании. Задача этой разметки – оценить, насколько формулировка цели удовлетворяет критериям целеполагания, и создать банк возможных вопросов для её уточнения.

В процессе подготовки инструмента для разметки, методика целеполагания SMART была адаптирована под доступную выборку целей и задачи платформы обучения: критерий «Relevant (уместность)» был трансформирован в «Цель относится к образованию», так как платформа может предложить решение только в рамках образовательного контента. Оценки критериев «Specific (конкретность)» и «Measurable (измеримость)» сильно коррелировали, поэтому было решено оставить только один из них. Чем лучше пользователь концентрируется на одной цели, тем больше вероятность того, что он её достигнет, поэтому в разметку был внесён критерий «Однозначность формулировки». Полученная структура экспертной разметки описана в таблице 7. Каждая цель независимо оценивалась тремя людьми, итоговая метка для бинарных полей была получена с помощью метода «мнение большинства».

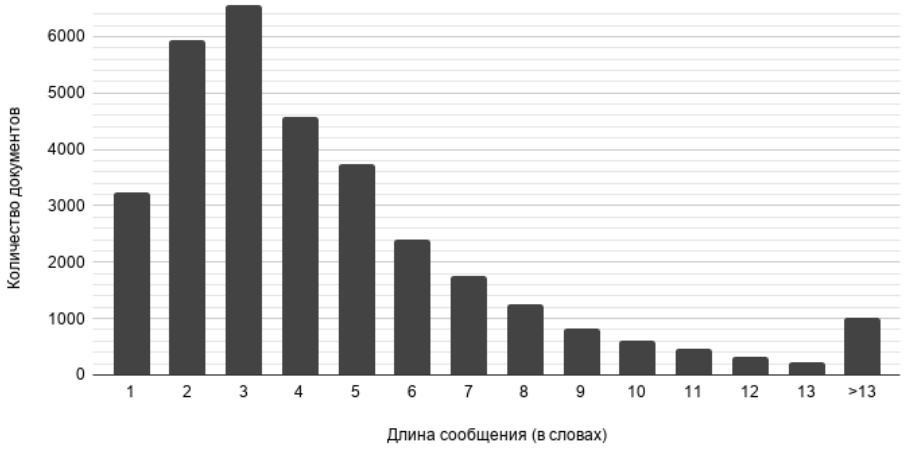
Название поля	**Описание поля**
Текст цели	Текстовое описание цели в свободной форме на русском языке. Максимальная длина – 220 слов, средняя и медианная длины – 4 слова. Распределение длин:  Используемый словарь состоит из 7948 слов, самые употребимые слова: *«новый»*, *«знание»*, *«получить»*, *«навык»*, *«профессиональный»*. В основном используются слова повседневной лексики, почти не встречаются термины. Большинство целей касается профессионального или личностного роста.
Конкретность	Поле содержит ответ на вопрос *«Представляете ли вы себе результат по этой цели?»*. Позволяет понять, может ли пользователь конкретно обозначить результат, которого хочет достичь. Возможные значения этого поля: *«Да, чётко»*, *«Да, нечётко»*, *«Нет»*.
Ограниченность во времени	Поле содержит ответ на вопрос *«Каким может быть первый шаг для достижения цели?»* либо в свободной форме, либо в виде строки *«Не знаю, с чего начать»*.
Первый шаг	Поле содержит ответ на вопрос *«Каким может быть первый шаг для достижения цели?»* либо в свободной форме, либо в виде строки *«Не знаю, с чего начать»*. Помогает уточнить текстовую формулировку цели и косвенно оценить достижимость цели.
Преграды для достижения	Поле содержит ответ на вопрос *«Что может помешать вам достичь цели?»* либо в виде строки *«Не вижу преград»*. Косвенно помогает оценить достижимость.
Тип запроса	Предполагает выбор нескольких ответов из заранее подготовленных вариантов: *«Приобрести новые навыки»*, *«Изменить привычки»*, *«Наработать новые связи»*. Или, если ни один из вариантов не подходит, пользователь вводит информацию на естественном языке. Используется для уточнения цели.
Тематическая область цели	Описывает тематические области, которых, по мнению человека, касается данная цель. Используется для уточнения цели (например, вместо *«профессиональный рост»* мы получим *«профессиональный рост в области строительство »*). Предполагает выбор из заранее подготовленных вариантов, например *«Иностранные языки»*, *«Управление сообществами, организациями»*, или, если ни один из вариантов не подходит, пользователь вводит информацию на естественном языке.

Таблица 5: Данные о цели пользователя

Название поля	**Тип данных**	**Описание поля**
Пол	Строка	Возможные варианты: мужской/женский
Возраст	Целое число	
Город проживания	Строка	Город в РФ
Образование	Строка	Один из 7 предложенных вариантов (например, *«высшее образование»*, *«учёная степень»*) и уточняющее описание в свободной форме.
Основное занятие	Строка	Один из 9 предложенных вариантов или описание в свободной форме.
Сфера профессиональной занятости	Строка	Поле описывает, в какой профессиональной области работает человек. Один из 23 предложенных вариантов или ответ в свободной форме.
Карьера (заполняется в профиле)	Строка	Это поле не является обязательным. Содержит все места работы пользователя.

Таблица 6: Социально-демографические характеристики пользователя

Название поля	**Описание поля**
Специфичность цели	Экспертное мнение о конкретности поставленной цели. Варианты ответа: *«специфична»*/*«не специфична»*
Достижимость цели	Экспертное мнение о том, сможет ли данный пользователь достичь заданную цель. Варианты ответа: *«достижима»*/*«не достижима»*
Ограниченность во времени	Характеризует наличие слов-маркеров, указывающих на категорию времени. Варианты ответа: *«ограничена»*/*«не ограничена»*
Образовательная направленность	Показывает, можно ли порекомендовать образовательную активность для достижения цели. Варианты ответа: *«цель относится к образованию»*/*«цель не относится к образованию»*
Однозначность формулировки	Разделяется ли заявленная цель на несколько самостоятельных целей. Возможные варианты ответа: *«однозначна»*/*«не однозначна»*.
Уточняющие вопросы	Потенциальные вопросы, которые стоит задать человеку для уточнения цели.

Таблица 7: Разметка цели пользователя экспертом в целеполагании

3.3 Предметная таксономия

Таксономия содержит структурную информацию, характеризующую как цели пользователей, так и образовательный контент.

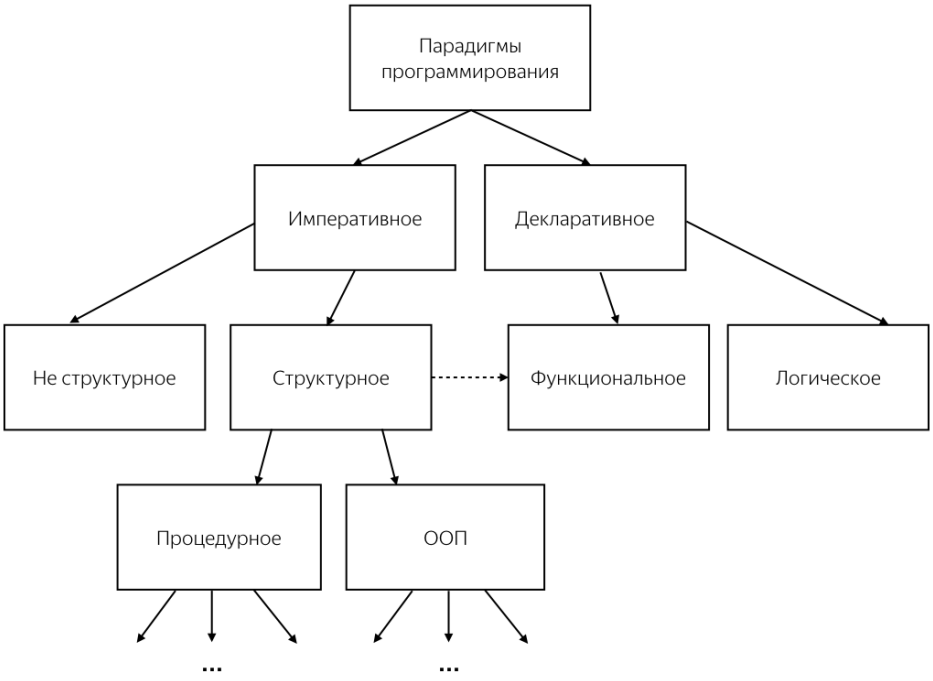


Рис. 1: Пример части таксономии

Она представляет собой древовидный граф, вершины которого – это профессиональные навыки. Таксоны, находящиеся ниже корневого, описывают специфические компетенции, причём совокупность дочерних таксонов исчерпывающе характеризует родительский таксон. Например, навык «Языки программирования» может выступать в роли родительского таксона, тогда дочерними вершинами будут всевозможные языки программирования, доступные для изучения на платформе.

3.4 Связи между данными

Все три типа данных, которые присутствуют в системе, активно взаимодействуют друг с другом.

Прямая связь между пользователями u_1, u_2 может появиться, если u_1 *консультирует* u_2. В таком случае, u_1 выступает в качестве эксперта в некоторой области, поэтому может помочь профессиональным советом или выступить в роли коуча. Кроме этой связи возникают и другие, неявные: например, u_1 и u_2 могут начать работу над одним проектом, пройти одинаковые курсы, посетить одно и то же мероприятие. Такой вид взаимодействий также должен учитываться моделью.

На данный момент виды связей между пользователем u и образовательным контентом c не сильно зависят от типа контента:

1. u *проходит* курс c: как правило, после прохождения курса пользователь приобретает новые навыки, что важно учитывать для следующих рекомендаций. Аналогичная логика соответствует взаимодействию u *посещает* мероприятие c.

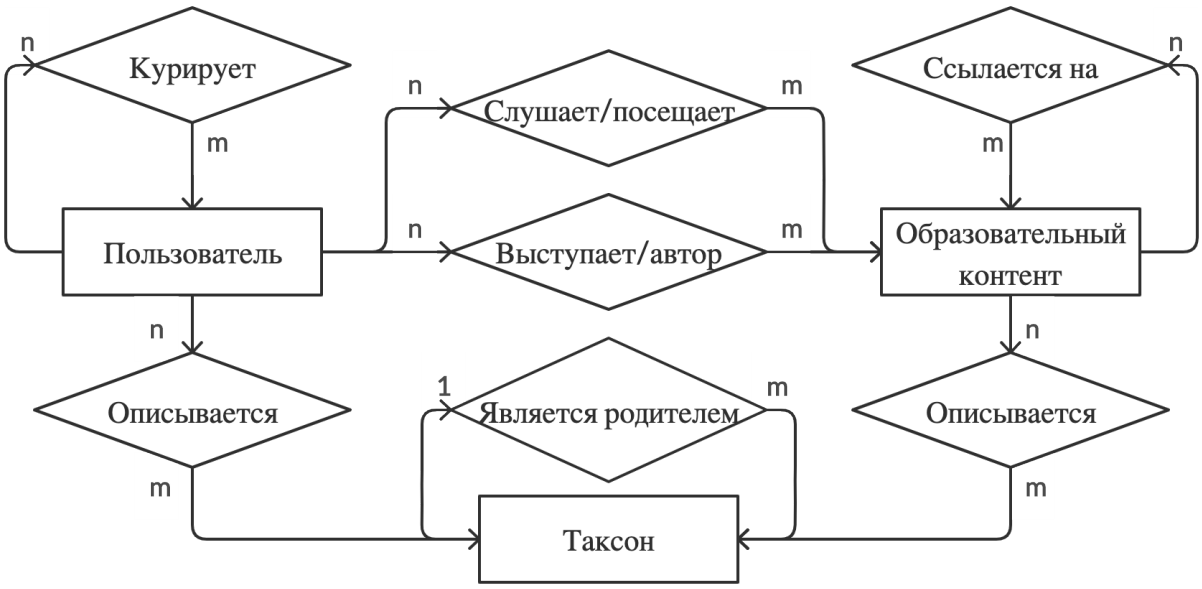


Рис. 2: Схема связей между различными сущностями

2. *и* ***организует*** курс *c* или ***является контактным лицом*** мероприятия *c*: доказывает наличие организационных навыков – одно из важнейших умений для эффективной командной работы;
3. *и* ***является автором*** курса *c* или ***выступает*** на *c*: это означает, что пользователь имеет глубокие знания в предмете, что подтверждает его экспертность в данной теме.

С появлением информации о практических проектах количество типов связей увеличится, например, в них будет учитываться в какой роли выступал каждый пользователь-участник команды.

Кроме этого, существуют связи между элементами образовательного контента: например, курс требует предварительного прохождения другого курса или в процессе обучения предлагается посетить тематическое мероприятие.

Связи между таксонами внутри таксономии описаны в разделе 3.3 и задаются экспертами в области. Для части пользователей платформы также доступна экспертная разметка по таксономии: человек относится к тому таксону, которому соответствует его цель. Разметка для контента (как для курсов, так и для мероприятий) появляется при его создании: организатор проставляет соответствующие теги.

Таким образом, все возможные связи в данных можно отобразить на схеме 2.

На момент первой рекомендации некоторых связей между данными ещё не будет, при решении проблемы «холодного старта» будем опираться только на данные, которые отображены на 3. Пунктирными линиями на ней отображены необязательные связи.

Хорошей моделью данных, обладающих описанной выше структурой, является граф. Построим его следующим образом: вершины графа будут обозначать экземпляры сущностей, а рёбра – связи, возникающие между ними. Пример такого графа, построенного на данных с платформы приведён на 4.

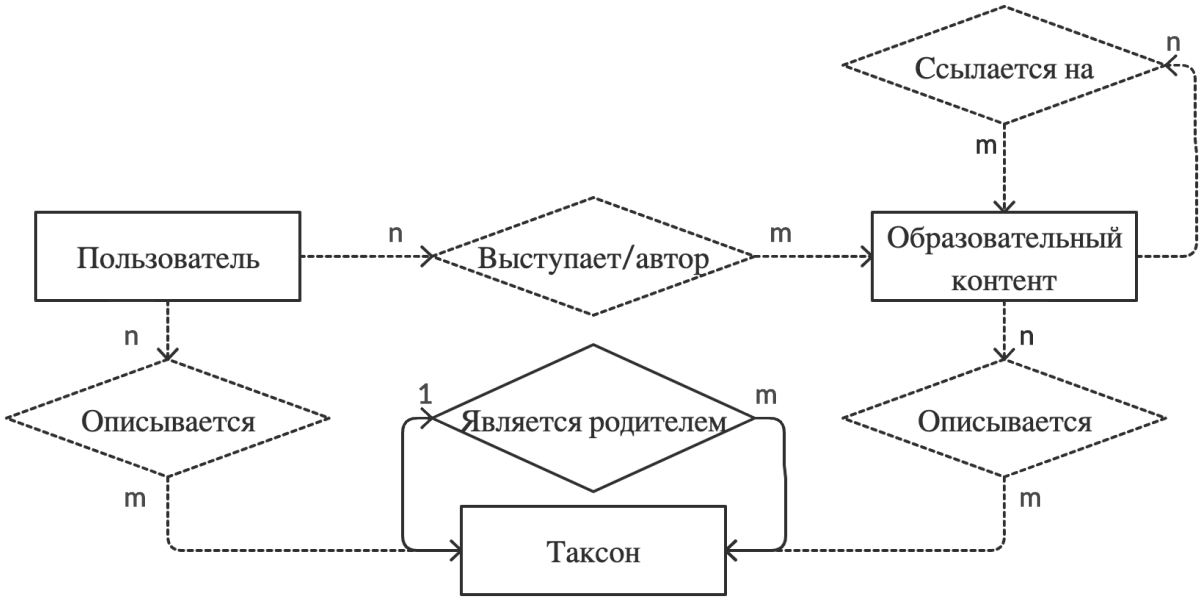


Рис. 3: Данные, доступные на момент первой рекомендации. Пунктирными линиями отмечены необязательные связи

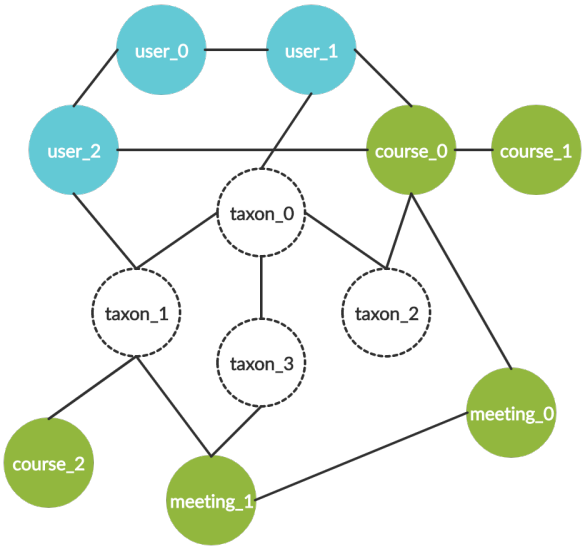


Рис. 4: Пример графа, построенного на данных платформы

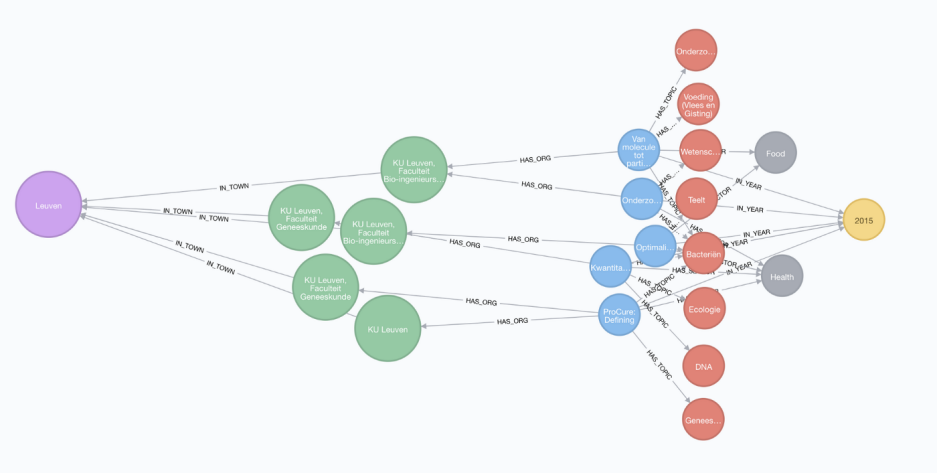
4 Обзор литературы

Одним из традиционных решений проблемы холодного старта является гибридный подход: для нового пользователя или новой единицы контента рекомендация даётся на основе дополнительной информации [4]. Для извлечения признаков из нового контента предлагаются разные подходы: например, используются модели, извлекающие характеристики из текстового описания, из изображений и видео. Основной источник данных для описания пользователя – опросники и информация профиля [5].

Так как агрегирование любой дополнительной информации в модели позволяет дать более качественные рекомендации, то в исследованиях предлагается воспользоваться преимуществами графовой структуры данных и агрегировать в векторных представлениях вершин-сущностей информацию об их взаимодействиях. Такой подход обширно используется для изучения структуры социальных сетей [6], однако в контексте данной работы особый интерес представляют исследования, посвящённые построению векторных представлений на *графе знаний* [7].

В графе знаний отображается не только взаимодействие разных сущностей, но и тип их взаимодействия. Например, на 5 показан граф отношений исследовательских институтов (зелёные вершины), статей, которые они выпускают (голубые вершины), тем, которые эти статьи затрагивают (красные вершины, получены автоматически с помощью тематического моделирования) и года выпуска (жёлтая вершина).

Рис. 5: Пример графа знаний



В [8] было показано, что построение рекомендаций с использованием графа знаний, эквивалентно решению задачи предсказания наличия рёбер между вершинами. При этом в качестве признакового описания вершины предлагается использовать некоторый векторный эмбеддинг, отражающий её описание и уже существующие связи. Классическим способом построения векторных представлений является метод TransE, описанный в [9]. В нём и вершины-сущности и рёбра-отношения погружаются в единое векторное пространство так, что для тройки (head_item, relation, tail_item) соответствующие вектора $\langle i_h, r, i_t \rangle$ удовлетворяют отношению: $i_h + r = i_t$, то есть сумма векторных представлений первой вершины и ребра хорошо приближает эмбеддинг второй.

Такой метод, как и у другие подходы, использующие предположения о геометрических свойствах эмбеддингов, концентрируется на использовании рёбер и не учитывает текстовые описания вершин. Первая попытка агрегировать дополнительную информацию предпринята в [10]. В ней предлагается модель NTN, которая представляет каждую

вершину усреднённым вектором слов из описания. Слабость такой модели в использовании предобученных эмбеддингов и эмпирической их агрегации, поэтому в следующем алгоритме, DKRL [11], предлагается построить энкодер, использующий свёрточную сеть. Как NTN, так и DKRL обучаются, оптимизируя margin-based функцию потерь на парах верных и неверных троек (head_item, relation, tail_item). Сильная сторона DKRL – обучение текстовых векторов в контексте определённой задачи – является одновременно и его слабостью, так как полученные эмбеддинги сильно уступают в качестве тем, при построении которых использовались языковые модели. Устранить этот недостаток позволяет fine-tuning обученных архитектур. Именно такой подход предложен в [12]: в качестве основной модели выбрана архитектура BERT[13], которая обучается на задаче предсказания следующего предложения в статьях из Википедии, а затем дообучается на данных из конкретной задачи. Эта модель показывает самое высокое качество в задаче предсказания ребра в графе.

Несмотря на высокое качество решения задачи дополнения графа, у таких подходов есть существенный недостаток: получаемые в результате векторные представления являются плохо интерпретируемыми, что делает рекомендации необъяснимыми [14].

Таким образом, изучение литературы показало, что добавление в модель дополнительной информации улучшает качество рекомендаций. Кроме того, для данных графовой структуры первоначальная задача эквивалентна задаче предсказания ребра в графе, что делает возможным применение большого количества новых подходов к решению. Малое внимание в исследованиях уделяется интерпретируемости полученных представлений. В данной работе изучим подход к построению векторных представлений с помощью тематических моделей.

5 Тематическая модель

Пусть D – коллекция документов, W – словарь всех слов в документах, T – конечное множество тем. Каждый отдельный документ $d \in D$ представляет собой набор слов $\{w \in W\}$. Каждое вхождение терма w связано с некоторой латентной переменной $t \in T$. Эту переменную в условиях нашей задачи будем называть *темой или интересом*.

Предполагается, что порядок слов в документе можно не учитывать (гипотеза «мешка слов») и что вероятность слова зависит от темы и не зависит от документа. Тогда вероятность встретить слово в документе:

$$p(w \mid d) = \sum_{t \in T} p(w \mid t) p(t \mid d) = \sum_{t \in T} \varphi_{wt} \theta_{td}$$

Или в матричной записи:

$$F = \Phi\Theta$$

где $F = (p(w|d))_{W \times D}$ – матрица частот слов в документах, $\Phi = (\varphi_{wt})_{w \in W, t \in T}$ – матрица вероятностей слов в темах, $\Theta = (\theta_{td})_{t \in T, d \in D}$ – матрица вероятностей тем в документах.

Основная задача тематического моделирования состоит в нахождении распределений Φ, Θ.

В вероятностном семантическом анализе (PLSA) для оценки матриц Φ, Θ максимизируется логарифм правдоподобия выборки:

$$\sum_{d \in D} \sum_{w \in d} n_{wd} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} \to \max_{\Phi, \Theta}$$

при ограничениях неотрицательности и нормировки

$$\sum_{w \in W} \varphi_{wt} = 1; \quad \varphi_{wt} \geqslant 0; \quad \sum_{t \in T} \theta_{td} = 1; \quad \theta_{td} \geqslant 0$$

где (n_{wd} – количество вхождений слова w в документ d).

Данная задача решается с помощью EM-алгоритма. На E-шаге вычисляются условные вероятности всех тем для каждой пары термин, документ:

$$p(t|d, w) = p_{tdw} = \frac{\varphi_{wt} \theta_{td}}{\sum_s \varphi_{ws} \theta_{sd}}$$

На M-шаге происходит оценка параметров распределения:

$$\varphi_{wt} = \frac{n_{wt}}{n_t}, \quad n_t = \sum_{w \in W} n_{wt}, \quad n_{wt} = \sum_{d \in D} n_{dw} p_{tdw};$$

$$\theta_{td} = \frac{n_{td}}{n_d}, \quad n_d = \sum_{t \in T} n_{td}, \quad n_{td} = \sum_{w \in d} n_{dw} p_{tdw}$$

Для добавления в модели информации о связях между данными, теория тематического моделирования предлагает несколько инструментов, описанию которых посвящены следующие части.

5.1 Мультимодальная модель

Существуют способы обогащения модели дополнительными метаданными, которые помогают определить тематику документа. Модель может учитывать автора документа, жанр или упомянутый тег и используемые в тексте изображения. Каждый тип данных образует отдельную модальность со своим словарём, а документ представляет собой контейнер слов разных модальностей.

Пусть M – множество модальностей, словари которых W_m попарно не пересекаются. Тематическая модель каждой модальности будет иметь такой же вид, как и тематическая модель, введённая ранее:

$$p_{wd} = \sum_{t \in T} \varphi_{wt}\theta_{td}, w \in W_m, m \in M, d \in D$$

Тогда для каждой модальности можно построить матрицу Φ_m. Записанные в столбец, они образуют матрицу Φ; при этом распределение тем в документе остаётся общим для всех модальностей.

При постановке задачи оптимизации берётся взвешенная линейная комбинация прологарифмированных функций правдоподобий каждой модальности:

$$\sum_{m \in M} \tau_m \sum_{d \in D} \sum_{w \in W_m} n_{dw} \ln \sum_{t \in T} \varphi_{wt}\theta_{td} \to \max_{\Phi,\Theta}$$

Данная задача также решается с помощью EM-алгоритма. Добавление модальностей видоизменяет только M-шаг:

$$\varphi_{wt} = \frac{n_{wt}}{n_t}, \quad n_t = \sum_{w \in W} n_{wt}, \quad n_{wt} = \sum_{d \in D} \tau_{m(w)} n_{dw} p_{tdw};$$

$$\theta_{td} = \frac{n_{td}}{n_d}, \quad n_d = \sum_{t \in T} n_{td}, \quad n_{td} = \sum_{m \in M} \sum_{w \in W^m} \tau_m n_{dw} p_{tdw}$$

Инструмент модальностей используется для исчерпывающего описания пользователя или курса. Добавление возраста, образования и основного занятия делает тематические распределения более согласованными с дополнительными характеристиками пользователя. По этой же причине в качестве отдельной модальности предлагается использовать данные об экспертной разметке. Таким образом, каждый пользователь представляет собой документ, состоящий из слов цели и дополнительных модальностей экспертной разметки и характеристик пользователя.

Каждый отдельный вид контента обладает своими, уникальными мета-характеристиками, которые также можно учесть в отдельной модальности. Кроме того, с их помощью можно моделировать связи между единицами контента. Для этого достаточно добавить модальность ссылок $W_m = D$, тогда если курс d ссылается на курсы $\{c_i\}_{i=1}^N$, то идентификатор c_i добавляется в d столько раз, сколько он упомянут в d.

Использование модальностей позволяет также моделировать связи между таксономией и другими сущностями в задаче. Добавление таксона как отдельной модальности документа приводит к тому, что для документов с одинаковым набором таксонов модель строит близкие тематические распределения. Это хорошо согласуется с логикой задачи: пользователи и курсы, привязанные к одному и тому же таксону, имеют схожие тематические распределения.

5.2 Регуляризация

Задача матричного разложения является некорректно поставленной по Адамару, так как множество её решений в общем случае бесконечно. Для решения такого рода задач существует общий подход – регуляризация. Дополнительный критерий – регуляризатор позволяет учесть в модели знания о предметной области и специфику решаемой задачи.

Теория аддитивной регуляризации тематических моделей ARTM снимает ограничение на вероятностную природу регуляризатора и позволяет учитывать влияние сразу нескольких регуляризаторов. Таким образом, максимизируемый функционал представляет собой линейную комбинацию логарифма правдоподобия $L(\Phi, \Theta)$ и регуляризаторов $R_i(\Phi, \Theta)$ с неотрицательными весами $\tau_i, i = 1, ...k$:

$$\sum_{m \in M} \tau_m \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} + R(\Phi, \Theta) \to \max_{\Phi, \Theta}$$

$$R(\Phi, \Theta) = \sum_{i=1}^{k} \tau_i R_i(\Phi, \Theta)$$

при ограничениях неотрицательности и нормировки

$$\sum_{w \in W} \varphi_{wt} = 1; \quad \varphi_{wt} \geqslant 0; \quad \sum_{t \in T} \theta_{td} = 1; \quad \theta_{td} \geqslant 0$$

Добавление регуляризатора не меняет E-шага алгоритма. Формулы M-шага учитывают дополнительную регуляризационную добавку:

$$\varphi_{wt} = \text{norm}_{w \in W} \left(n_{wt} + \varphi_{wt} \frac{\partial R}{\partial \varphi_{wt}} \right) \quad n_{wt} = \sum_{d \in D} \tau_{m(w)} n_{dw} p_{tdw}$$

$$\theta_{td} = \text{norm}_{t \in T} \left(n_{td} + \theta_{td} \frac{\partial R}{\partial \theta_{td}} \right) \quad n_{td} = \sum_{m \in M} \sum_{w \in W^m} \tau_m n_{dw} p_{tdw}$$

где $\text{norm}(x_i) = \frac{\max(0, x_i)}{\sum_i \max(0, x_i)}$.

Благодаря регуляризации модель способна учитывать специфику некоторых связей, которую сложно передать с помощью дополнительных модальностей. Например, отношение иерархии, которое связывает таксоны в таксономии можно формализовать с помощью регуляризатора.

Регуляризация таксономии Таксономия представляет собой иерархическую структуру, причём совокупность дочерних таксонов исчерпывающе описывает родительский таксон. Именно это свойство таксономии позволяет выдвинуть предположение, что вектор интересов родительского таксона должен хорошо приближаться вероятностной смесью [3] векторов дочерних вершин:

$$R(\Theta) = -\sum_{v} \left(p(t|v), \frac{1}{|C_v|} \sum_{c \in C_v} p(t|c) \right) = -\sum_{v} \left(\theta_v, \frac{1}{|C_v|} \sum_{c \in C_v} \theta_c \right) = -\sum_{v} \sum_{c \in C_v} \frac{(\theta_v, \theta_c)}{|C_v|}$$

[3] На первом этапе считаем дочерние вершины равновероятными, однако это предположение нуждается в дальнейшем исследовании.

где v – родительская вершина, c – дочерняя вершина, C_v – множество всех дочерних вершин вершины v, θ_c – вектор интересов для вершины c (в терминах тематического моделирования – тематическое распределение документа c).

Применяя данный регуляризатор, мы будем поощрять модель приближать распределения родительского таксона и смеси распределений дочерних таксонов. Однако на первых этапах построения таксономии дочерние вершины могут не полностью описывать родительскую: например, компетенция «Языки программирования» имеет дочерние вершины «C++», «Python», «Java», однако, очевидно, это не полный набор всевозможных языков программирования.

В таком случае предлагается заложить в модель чуть менее сильное предположение: вектор интересов родительского таксона должен быть близок к вектору интересов каждого дочернего таксона (это не обязывает его хорошо приближаться вероятностной смесью). Данное предположение формализуется с помощью регуляризатора

$$R(\Theta) = -\sum_{v}\sum_{c\in C_v}(p(t|c), p(t|v)) = -\sum_{v,c\in D_{\text{Tax}}}(\theta_c, \theta_v)$$

Регуляризация Φ Данный регуляризатор при обработке документа предполагает знания о тематических векторах других документов, что требует хранения матрицы Θ в памяти. Такой подход затрудняет пакетную обработку большой коллекции, поэтому построим регуляризаторы матрицы Φ, основанные на схожих предположениях. Пусть G – модальность таксонов, тогда $\varphi_{gt} = p(g|t)$. По формуле Байеса выразим тематическое распределение таксона:

$$p(t|g) = \frac{p(g,t)}{p(g)} = \frac{p(g|t)p(t)}{p(g)} = \frac{\varphi_{gt}n_t}{n_g}$$

где n_g – частота таксона, $n_t = \sum_g n_{gt}$ – частота темы t в модальности таксонов G, вычисляемая EM-алгоритмом. Тогда вместо того, чтобы приближать тематические распределения документов, будем приближать тематические распределения таксонов с помощью регуляризатора

$$R(\Phi) = -\sum_{g,w\in W_{Tax}}\sum_{t\in T} n_t^2\frac{\varphi_{gt}\varphi_{wt}}{n_g n_w}$$

При добавлении данного регуляризатора формула E-шага остаётся неизменной, а формула M-шага модифицируется следующим образом:

$$\varphi_{wt} = \text{norm}_{w\in W}\left(n_{wt} + \tau\varphi_{wt}\sum_{g:(g,w)\in W_{Tax}}\sum_{t\in T}\frac{\varphi_{gt}n_t^2}{n_g n_w}\right), \quad n_{wt} = \sum_{d\in D}\tau_{m(w)}n_{dw}p_{tdw}$$

Регуляризатор разреживания Φ Введём *предположение разреженности* для модальности таксономии: каждая тема характеризуется небольшим количеством таксонов. Регуляризатор, формализующий данное предположение, максимизирует кросс-энтропию между столбцами матрицы Φ и фиксированным распределением:

$$R(\Phi, \Theta) = \sum_{t\in T}\sum_{w\in W}\beta_{wt}\ln\varphi_{wt}$$

Положительное β_{wt} соответствует сглаживанию, а отрицательное – разреживанию.

6 Метрики качества

Для оценки качества построенных представлений будем использовать две группы метрик - внутренние метрики, которые характеризуют качество построенной тематической модели, и внешние метрики, которые оценивают полезность с точки зрения практических задач.

Качество рекомендаций Внешняя метрика, которая оценивает качество решения проблемы «холодного старта». Каждая рекомендация подразумевает упорядоченный набор из нескольких единиц образовательного контента, поэтому и для оценки качества используются метрики, подразумевающие упорядоченную выдачу. Опишем первую метрику такого типа – MAP@N.

Если пользователь взаимодействовал с единицей контента, то будем называть эту единицу *релевантной*. Точностью (precision) $P(k)$ будем называть долю релевантных рекомендаций в наборе из первых k. Тогда средняя точность (average precision):

$$\text{AP@N} = \frac{1}{N}\sum_{k=1}^{N}\big(P(k) \text{ если } k^{\text{ый}} \text{ элемент выдачи релевантен }\big) = \frac{1}{N}\sum_{k=1}^{N} P(k)\cdot \text{rel}(k)$$

где $\text{rel}(k)$ – индикатор релевантности $k^{\text{ого}}$ элемента рекомендательной выдачи. Усреднив по всем пользователям U, получим MAP@N (mean average precision):

$$\text{MAP@N} = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{N}\sum_{k=1}^{N} P_u(k)\cdot \text{rel}_u(k) \tag{1}$$

Другой метрикой, позволяющий оценить качество рекомендаций является nDCG. Определим discounted cumulative gain at N (DCG@N) следующим образом:

$$\text{DCG@N} = \sum_{k=1}^{N}\frac{2^{\text{rel}(k)}-1}{\log(k+1)}$$

DCG@N принимает максимальное значение, когда первые N элементов отсортированы согласно релевантности. Нормируя DCG@N по максимальному значению, получим nDCG@N. Аналогично MAP, данную метрику можно усреднить по пользователям.

В данной работе эти функционалы качества оцениваться не будут из-за недостатка исторических данных о взаимодействии пользователя и системы.

Качество предсказания экспертной оценки Другой внешней метрикой служит качество решения задачи предсказания экспертных оценок. С одной стороны, эта задача возникает при разработке чат-бота, уточняющего постановку целей пользователя: чтобы принять решение о следующем вопросе для пользователя, в каждый момент времени чат-бот должен понимать, насколько текущая формулировка соответствует критериям целеполагания.

С другой стороны, качество решения задачи предсказания экспертных оценок коррелирует с качеством построенных эмбеддингов: чем полнее они описывают контентное представление вершины, тем лучше восстанавливают зависимость между целью и экспертной оценкой.

Таким образом, на эмбеддингах строится четыре классификатора, которые предсказывают характеристики цели: Specific (специфичность), Achievable (достижимость),

Time-bound (ограниченность во времени), Educational (образовательная направленность), Unambiguity (однозначность формулировки). Для построения классификатора был использован *метод опорных векторов*, для оценки качества классификации – *F1-мера*.

Качество предсказания наличия ребра Последняя метрика из серии внешних. Построенные векторные представления инкапсулируют информацию не только об описании конкретной вершины, но и о связях между ними. Чем лучше они это делают, тем выше качество решения задачи предсказания рёбер между вершинами. Поэтому для оценки качества эмбеддингов предлагается собрать выборку, состоящую из пар вершин (таксон, пользователь), (таксон, элемент образовательного контента) и решить задачу бинарной классификации – предсказать наличие ребра между вершинами пары.

Отрицательные примеры генерируются по следующему алгоритму: из набора целей с разметкой по таксономии выбирается случайная цель g и случайный таксон t. Обозначим за t' – таксон, соответствующий цели g, тогда для пары (g, t) проверяются следующие условия: таксон t не лежит в поддереве с корнем в t' и, обратно, t' не лежит в поддереве с корнем t. Таким образом можно исключить ситуацию, когда сгенерированный таксон хорошо, но недостаточно конкретно описывает цель (то есть ребро могло бы существовать, но не является оптимальным). Для удобства анализа генерировалась сбалансированная выборка.

В качестве классификатора предлагается использовать линейную модель, построенную *методом опорных векторов*, качество работы оценивать с помощью *F1-меры*.

Перплексия Наиболее распространённый внутренний критерий оценки качества. Определяется через log-правдоподобие отдельно для каждой модальности.

$$P(D, W) = \exp\left(-\frac{1}{n}\sum_{d\in D}\sum_{w\in W} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td}\right)$$

где $n = \sum_{d\in D}\sum_{w\in W} n_{dw}$. Чем меньше перплексия, тем лучше модель описывает появление слов w в документах d.

С помощью этой метрики некорректно сравнивать тематические модели, построенные на разных словарях.

Интерпретируемость тем Измерение интерпретируемости тематической модели является плохо формализуемой задачей, поэтому большинство методов предполагает привлечение асессоров.

Для оценки этой характеристики на этапе построения модели в данной работе будут использоваться *лексические ядра тем* – множество слов, которые с большой вероятностью употребляются в теме t и редко употребляются в других темах:

$$W_t = \{w \in W | p(t|w) > 0.25\}\,, \text{ где } p(t|w) = \varphi_{wt}\frac{n_t}{n_w}$$

Для оценки качества финальной модели также можно использовать модифицированный *метод интрузий*: для каждой темы составить список из 7 самых вероятных таксонов, в который подмешать случайный таксон. Тогда тема считается интерпретируемой, если подавляющее число экспертов правильно находит лишний таксон.

7 Эксперименты

7.1 Описание и предобработка данных

Коллекция документов состоит из 32943 целей (из них для 6000 доступна экспертная разметка, 1000 имеет привязку к таксономии) и 8170 единиц контента (из них 7980 – мероприятия, остальное – курсы, к таксономии привязано 500 уникальных единиц).

Предобработка данных включала в себя удаление пунктуации, фильтрацию стоп-слов, приведение слов к нижнему регистру, лемматизацию (приведение к начальной форме). Кроме того, из текстов были выделены *биграммы* – частые словосочетания из двух слов (редкие биграммы удалялись в процессе предобработки).

Среди данных о целях 1% составляет шум – бессмысленный набор символов, который невозможно интерпретировать как цель. Для фильтрации такого вида текстов была построена модель, решающая задачу бинарной классификации. Признаковое описание текстов формировалось из статистических характеристик: длины текста в словах, длины текста в символах, процента букв, количества повторений букв, средней длина слова в тексте, процента слов цели, найденных в словаре русского языка. В качестве модели использовалась логистическая регрессия. Качество работы алгоритма оценивалось по метрике AUC-ROC и на кросс-валидации (при разбиении на 10 частей) достигло значения 0.9886 ± 0.0104.

7.2 Подбор оптимальных параметров тематической модели

При построении тематической модели был произведён подбор оптимальных параметров: количества тем, модальностей и весов, с которыми они учитываются в модели. Перебор параметров производился в следующем порядке: сначала был найден оптимальный набор модальностей и их веса, а затем подобрано количество тем.

Здесь и далее в тексте приняты следующие обозначения для модальностей: 8 и аббревиатуры для для экспертных оценок: 9.

Words	модальность текстовых описаний
Bigramms	модальность биграмм выделенных из текстовых описаний
Taxonomy	модальность соответствующего таксона
SocialInfo	модальность социально-демографических характеристик пользователей
MetaInfo	совокупное название нескольких модальностей: для мероприятия – «Тип мероприятия», для курса – «Формат курса», «Длительность курса в неделях», «Планируемое количество потраченных часов»
Expert Score	модальность экспертной оценки цели

Таблица 8: Аббревиатуры модальностей

Specific	Конкретность
Achievable	Достижимость
Time-bound	Ограниченность во времени
Educational	Цель относится к образованию
Unambiguity	Однозначность

Таблица 9: Аббревиатуры экспертных оценок

Результаты эксперимента, представленные в таблице 10, показывают, что оптимальным с точки зрения восстановления рёбер графа является набор модальностей, состоящих из слов, биграмм и таксонов.[4] Так как добавление мета-признаков незначительно улучшает качество предсказания экспертных оценок, то для дальнейших экспериментов будем использовать модель WBT.

	Edge Prediction			Expert Scores Prediction				
Модальности	Prec	Rec	F1	S	A	T	E	U
W	0.7217	0.7108	0.7053	0.7568	0.7113	0.8069	0.8426	0.9412
WB	0.7719	0.8252	0.7933	0.8001	0.7851	0.8217	0.8798	0.9509
WBT	**0.8656**	**0.8734**	**0.8661**	**0.8142**	0.7790	0.8239	**0.8823**	0.9781
WBSM	0.7832	0.8193	0.8004	0.8115	**0.7882**	**0.8265**	0.8817	**0.9793**
WBTSM	0.8456	0.8623	0.8526	0.8097	0.7746	0.8190	0.8751	0.9761
WBTE	0.8582	0.8463	0.8517	-	-	-	-	-
WBTSME	0.8509	0.8278	0.8381	-	-	-	-	-

Таблица 10: Метрики качества для моделей с различными наборами модальностей: Words, Bigramms, Taxonomy, SocialInfo, MetaInfo, ExpertScore

Таблица 11 показывает, что для модели WBT оптимальное количество тем – 250.

	Edge Prediction			Experts Scores Prediction				
$\|T\|$	Prec	Rec	F1	S	A	T	E	U
100	0.7107	0.7444	0.7261	0.7746	0.7002	0.7812	0.8445	0.9548
200	0.8046	0.8487	0.8247	0.8103	0.7589	0.8177	0.8630	0.9615
250	**0.8656**	**0.8734**	**0.8661**	**0.8142**	**0.7790**	**0.8239**	**0.8823**	**0.9781**
300	0.8398	0.8525	0.8448	0.8130	0.7640	0.8063	0.8757	0.9704
400	0.7270	0.7615	0.7417	0.7921	0.7249	0.7933	0.8584	0.9635

Таблица 11: Метрики качества для модели WBT с разным количеством тем

7.3 Подбор оптимальной стратегии регуляризации

Регуляризация тематических моделей позволяет отобразить дополнительные ограничения поставленной задачи, поэтому при построении модели большого внимания заслуживает оптимизация порядка подключения регуляризаторов и степени их влияния (к-тов регуляризации).

[4] Для каждого набора модальностей подбиралось лучшее отношение весов, с которыми учитывается каждая из модальностей. В данном наборе оптимальным соотношением оказалось $\tau_W = 1, \tau_B = 10, \tau_T = 10$

На первых этапах построения модели важно внести в неё знания о структуре связей между таксонами, поэтому сначала подключается регуляризатор матрицы Φ (обозначим его R_1), описанный в 5.2. Далее в модель предлагается внести предположение о том, что каждая тема описывается небольшим количеством таксонов. Для этого необходимо подключить регуляризатор, разреживающий модальность таксономии (обозначим его R_2). Такая стратегия помогает сначала получить качественные векторные представления таксонов, а затем наложить дополнительные ограничения на связи между таксонами и другими сущностями.

Так как перплексия модели перестаёт значимо меняться после 40 итераций, то подключать второй регуляризатор будем через 20 итераций после начала процесса обучения. Подбор к-тов организован следующим образом: сначала строится 4 модели с разными к-тами для R_1, затем из них выбираются 2 лучших с точки зрения перплексии и разреженности модальности таксонов матрицы Φ. Далее к каждой из этих моделей подключается R_2 с 3 разными к-тами. Из полученных 6 моделей выбирается лучшая.

Согласно результатам эксперимента 6, оптимальный набор к-тов регуляризации: $\tau_1 = 10^4, \tau_2 = -1.5$.

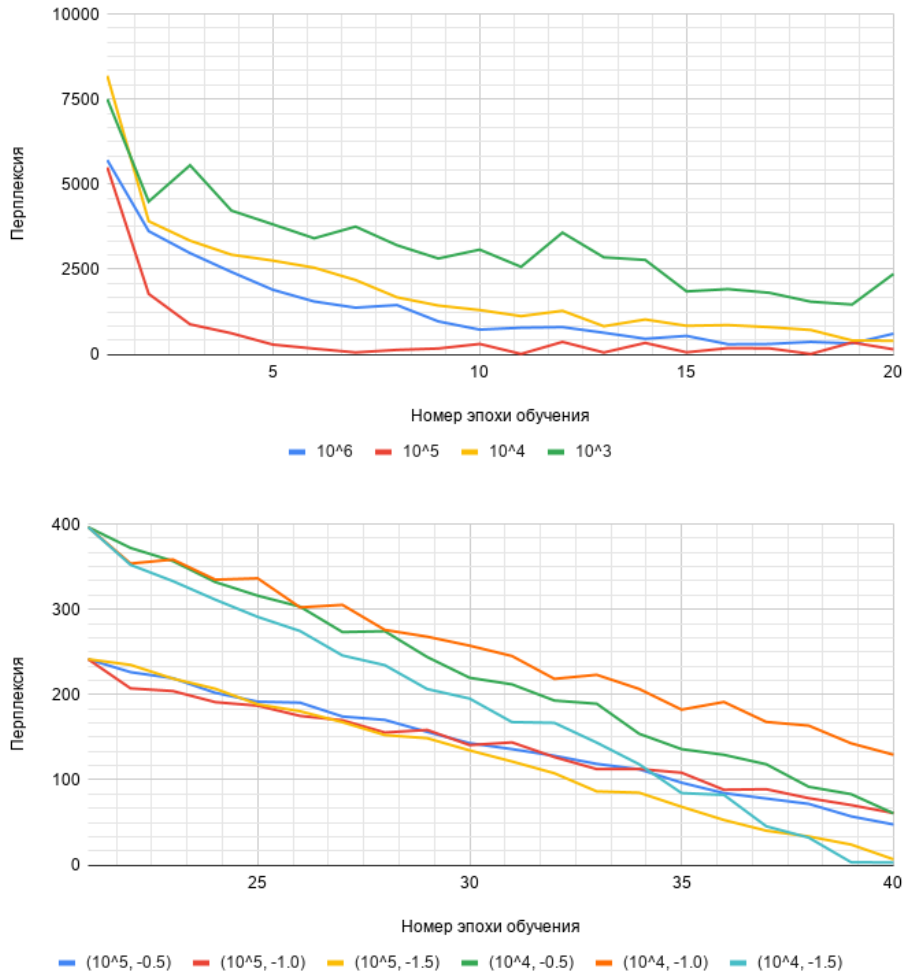


Рис. 6: Результаты эксперимента по подбору к-тов регуляризации

Такой подход к регуляризации модели дал прирост в качестве, который показан в 12.

	Edge Prediction			Expert Scores Prediction				
	Prec	Rec	F1	S	A	T	E	U
$\tau_1 = 0$, $\tau_2 = 0$	0.8656	0.8734	0.8661	0.8142	0.7790	0.8239	0.8823	0.9781
$\tau_1 = 10^4$, $\tau_2 = -1.5$	**0.8773**	**0.9132**	**0.8922**	**0.8279**	**0.7903**	**0.8317**	**0.8961**	**0.9883**

Таблица 12: Метрики качества для модели WBT до регуляризации и после

7.4 Сравнение моделей

Данных достаточно, чтобы выбрать в качестве бейзлайна модель DKRL[11] [5].

С точки зрения методологии, обучение DKRL – это минимизация энергии E, которая состоит из двух компонент: E_S – функция энергии структурного представления графа, E_D – функция энергии текстового представления вершин в графе [6]. Такой выбор функционала энергии мотивирован тем, что и вершины, и отношения между ними должны быть помещены в единое векторное пространство. В процессе обучения оптимизируется marging-based loss – функционал на основе отступа:

$$L = \sum_{(h,r,t)\in T} \sum_{(h',r',t')\in T'} \max(\gamma + d(h+r,t) - d\left(h'+r',t'\right), 0)$$

где $\gamma > 0$ – гиперпараметр отступа, $d(h+r,t)$ – функция различия между head_item + relation и tail_item, T' – множество отрицательных примеров, построение которого описано в 6. Метод оптимизации – стохастический градиентный спуск (SGD). Оптимизация данного функционала позволяет учитывать графовую структуру данных.

Для добавления информации о текстовых описаниях вершин в оригинальной статье предлагается использовать одну из двух архитектур: первая использует в качестве входных эмбеддингов вершин вектора, построенные с помощью tf-idf представления текстов-описаний, а вторая подразумевает использование энкодера, архитектура которого изображена на 7. Каждый столбец входной матрицы в таком случае представляет из себя конкатенацию векторных представлений слов[7] в окне размера k. Для сравнения были построены обе модификации модели.

В рамках эксперимента оптимизация гиперпараметров не проводилась: были использованы параметры, описанные в статье (скорость обучения $\lambda = 0.001$, отступ $\gamma = 1$, ширина окна $k = 2$, размерность векторных представлений $n = 100$).

В ходе эксперимента предлагалось сравнить качество работы трёх моделей:

1. DKRL с tf-idf представлениями текстов
2. DKRL с tf-idf представлениями текстов и не текстовыми признаками, извлечёнными из мета-информации
3. DKRL с энкодером на основе fastText представлений входных текстов[8]

Результирующее качество работы каждого из этих алгоритмов и их сравнение с тематическими моделями отражено в 7.4.

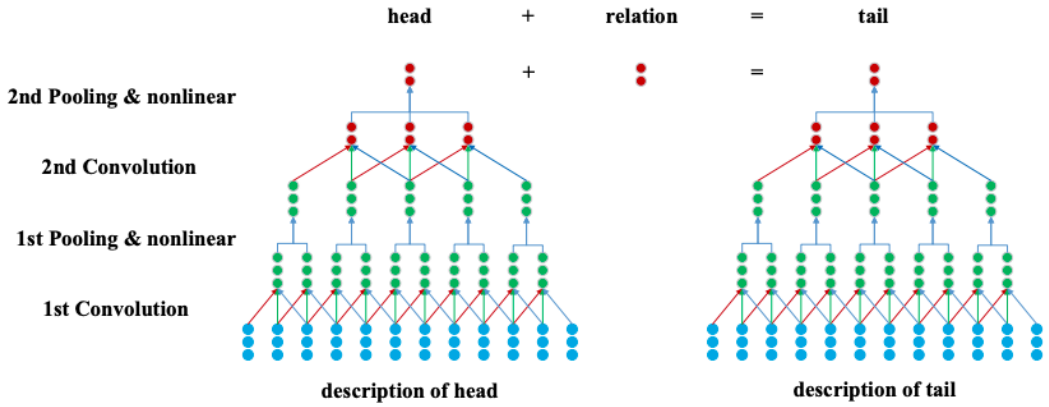
[5] Источник: https://github.com/xrb92/DKRL

[6] Более детально процесс построения функции энергии описан в статье [11].

[7] В оригинальной работе были использованы word2vec представления, в данном эксперименте выбраны fastText эмбеддинги

[8] Источник: http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html?highlight=embeddings

Рис. 7: Архитектура энкодера в DKRL



	Edge Prediction			Expert Scores Prediction				
	Prec	Rec	F1	S	A	T	E	U
WBT	0.8656	0.8734	0.8661	0.8142	0.7790	0.8239	0.8823	0.9781
Regularized WBT	0.8773	0.9132	0.8922	**0.8279**	**0.7903**	**0.8317**	**0.8961**	**0.9883**
DKRL (fastText)	**0.8910**	**0.9433**	**0.9148**	0.8008	0.7754	0.8216	0.8623	0.9402
DKRL (tf-idf)	0.8814	0.8910	0.8846	0.7893	0.7869	0.7932	0.8735	0.9515
DKRL (tf-idf + OHE)	0.8608	0.9135	0.8850	0.7864	0.7891	0.8111	0.8793	0.9596

В задаче предсказания рёбер в графе DKRL даёт более высокое качество, что говорит о том, что эмбеддинги, полученные с его помощью, лучше отражают графовую структуру данных. Однако значимо хуже этот подход показывает себя в задаче предсказания экспертных оценок, что говорит о плохой способности отображать текст и неуниверсальности данных векторных представлений.

7.5 Качественная оценка рекомендаций

Воспользуемся тем, что как пользователи, так и контент находятся в одном векторном пространстве и между ними можно измерить расстояние. Это позволяет строить рекомендацию на основе фиксированной метрики близости между векторными представлениями пользователей и курсов. В качестве мер близости можно использовать

1. скалярное произведение векторов: $d_1(u, c) = \sum_i u_i c_i$ (чем больше, тем лучше);
2. евклидово расстояние: $d_2(u, c) = \sqrt{\sum_i (u_i - c_i)^2}$ (чем меньше, тем лучше);
3. KL-дивергенция: $d_3(u, c) = \sum_i u_i \log(\frac{u_i}{c_i})$ (чем больше, тем лучше).

Для качественной оценки зафиксируем в качестве меры близости скалярное произведение. В таблицах ниже приведены примеры рекомендаций для фиксированной цели и разных моделей.

Первая исследованная цель «Понять, что такое большие данные» сформулирована очень общо. Рекомендации, полученные с помощью DKRL, более разнообразны: в них

присутствуют как курсы, так и мероприятия. Тематическая модель уловила интересную связь между большими данными и инструментами для их визуализации и анализа.

Цель: «Понять, что такое большие данные»	
Тематическая модель	1. Курс «Искусственный интеллект и большие данные цифровой экономики» 2. Курс «Введение в машинное обучение» 3. Курс «Excel» 4. Курс «Power BI» 5. Курс «Мастер Google-таблиц»
DKRL	1. Курс «Введение в машинное обучение» 2. Встреча «Программа развития компетенций «Цифровые навыки» » 3. Олимпиада НТИ по естественно-научному профилю 4. Курс «Применение технологии анализа больших данных при эксплуатации и обслуживании многоквартирного дома» 5. Хакатон «Технологии НТИ»

Вторая цель – «Написать свой сайт» – намного более конкретно сформулирована. Здесь интересно, что DKRL в своих рекомендациях делает упор на фронт-энд разработку, тогда как тематические модели предлагают полный необходимый комплекс.

Цель: «Написать свой сайт»	
Тематическая модель	1. Серия мероприятий «Разработка шаг за шагом современной информационной системы, используя web и мобильные технологии Django, Python, Angular, Xamarin Forms, Docker, PostgreSQL» 2. Курс «Разработка серверной части приложений. Базовый курс (DEV1)» 3. Семинар по Веб-технологиям - Основы HTML 4. Курс «Веб-дизайн и разработка» 5. Финальный хакатон конкурса «Budget Apps»
DKRL	1. Технологический семинар по вопросам разработки фронт-энд части веб-проекта 2. HTML-тренажер 3. Курс «Веб-разработка. Быстрый старт» 4. Семинар «Дизайн интерфейсов» 5. Курс «Веб-дизайн и разработка»

Третья цель – «Научиться создавать программы на Python» – тоже достаточно конкретна. Интересно, что DKRL нашёл связь между языком программирования Python и анализом данных – сферой, в которой этот язык используется чаще всего.

Цель: «Научиться создавать программы на Python»	
Тематическая модель	1. Курс «Создание Web-сервисов на Python» 2. Курс «ООП и паттерны проектирования в Python» 3. Курс «Автоматизация тестирования с помощью Selenium и Python» 4. Курс «Программирование на языке Python. Основы анализа и визуализации данных на языке Python. Библиотеки NumPy, Pandas, Matplotlib» 5. Воркшоп «Математика и Python для анализа данных (Математика и Python)»
DKRL	1. Воркшоп «Математика и Python для анализа данных (Математика и Python)» 2. Курс «Программирование на языке Python. Основы анализа и визуализации данных на языке Python. Библиотеки NumPy, Pandas, Matplotlib» 3. Курс лекций «Введение в машинное обучение» 4. Курс «Анализ данных на Python» 5. Курс «Базовое программирование и анализ данных с помощью Python»

Последняя цель – «Определиться с профессией». Сделать рекомендацию к такой цели очень сложно, для этого нужно много дополнительных данных о человеке. Судя по результатам, модели сильно реагируют на слово «профессия», что не всегда является правильным поведением.

Цель: «Определиться с профессией»	
Тематическая модель	1. Комплекс профориентационных мероприятий для старшеклассников «Атлас новых профессий» 2. Курс «Формирование профессионального сообщества» 3. Лекция на тему «Продолжение работы над темой профориентация будущего» 4. Форсайт-сессия для старшеклассников «Образ жизни: будущее» 5. Семинар по актуальным профессиям и знаниям
DKRL	1. Форсайт-сессия для старшеклассников «Образ жизни: будущее» 2. Открытая экспертная панель «Технологический предприниматель – основная профессия цифровой экономики 4.0» 3. Семинар по актуальным профессиям и знаниям 4. Круглый стол «Развитие некоммерческого сектора в России: волонтеры или профессионалы за зарплату» 5. Семинар «Образование будущего»

Такой оценки недостаточно, чтобы сравнивать между собой DKRL и тематическую модель или делать выводы о качестве решения задачи рекомендации контента. Однако качественная оценка показывает, что модели пригодны для задачи построения рекомендаций и выявляет некоторые особенности каждой из них.

7.6 Интерпретируемость тематической модели

Одно из основных преимуществ тематической модели – это её интерпретируемость. Каждая компонента вектора – это вероятность принадлежности документа некоторой теме, которую в свою очередь можно описать с помощью самых вероятных слов. Для построения таблицы 13 фиксировалась некоторая тема и выбирались списки слов, отсортированных по $p(w|t)$ и на документы, для которых вероятность фиксированной темы максимальна.

Тема	**Частые слова и биграммы**	**Цели**	**Образовательный контент**
Управление, менеджмент	бизнес, предприниматель, заработок, руководитель, управленческий, личные финансы, новые связи, лидерские качества	«Освоение проектного управления», «Развитие лидерских качеств», «Научиться управлять отделом»	«Лидерство – управление собой», «Новый цифровой уклад и устойчивое развитие бизнеса», «Корпоративное управление в обществе»
Анализ данных	аналитик, математика, питон, курсы, сбор данных, большие данные	«Изучить новые технологии обработки данных», «Разработать новые финансово емкие образовательные программы по анализу данных»	«Python для анализа данных», «Открытая лекция «Машинное обучение» », «Введение в машинное обучение»
Разработка	программирование, разработка, приложение, серверный, frontend, веб-дизайн	«Написать сайт или мобильное приложение», «Разобраться в основных фреймворках веб-программирования», «Разработать клиент-серверное приложение»	«Создание Web-сервисов на Python», «iOS-разработка с помощью Swift», «Быстрый старт в разработке Android-приложений»

Тема	**Частые слова и биграммы**	**Цели**	**Образовательный контент**
Начало или развитие карьеры	карьера, стажировка, профессия, заработок, успешная карьера	«Определиться с профессией», «Получить повышение по карьерной лестнице», «Найти, как применить знания, полученные в ВУЗе»	«Карьерный Квест по развитию компетенций для студентов», «Стратегическая сессия «От профессиональной культуры к профессиональной карьере» для студентов колледжей», «Мастер-класс для студентов от Лаборатории Карьеры»

Таблица 13: Темы, слова и биграммы, которыми они описываются, цели и контент, относящиеся к данной тематике

8 Выводы

В работе поставлена задача построения векторных представлений объектов рекомендательной системы: пользователей и образовательного контента. С целью повышения качества эмбеддингов, предлагается учитывать все дополнительные данные: как текстовые описания объектов, так и информацию о связях между ними.

Поставленная задача решалась методом тематического моделирования. В работе описаны два инструмента, позволяющих внести в модель дополнительную информацию: мультимодальная модель и регуляризация. Предложен и реализован регуляризатор, позволяющий внести в модель знания об иерерхических связях в данных. Кроме того, с помощью вычислительных экспериментов показано, что каждый из описанных инструментов повышает качество векторных представлений.

Согласно экспериментам, сильной стороной тематической модели является её способность агрегировать текстовую информацию. Качественно продемонстрировано и другое преимущество – интерпретируемость получаемых векторных представлений.

В работе приведён алгоритм построения рекомендаций с использованием векторных представлений объектов и качественный анализ показывает возможность использовать тематические эмбеддинги при решении проблемы «холодого старта», однако данная тема нуждается в дальнейшем исследовании.

Список литературы

[1] Dhoha Almazro и др. *A Survey Paper on Recommender Systems*. 2010. arXiv: 1006.5278 [cs.IR].

[2] *A survey on solving cold start problem in recommender systems*. 2017.

[3] *Improving the Onboarding User Experience in MOOCs*. июль 2014.

[4] Robin Burke. «Hybrid Recommender Systems: Survey and Experiments». в: *User Modeling and User-Adapted Interaction* 12 (нояб. 2002). DOI: 10.1023/A:1021240730564.

[5] Matthias Braunhofer, Mehdi Elahi и Francesco Ricci. *User Personality and the New User Problem in a Context-Aware Point of Interest Recommender System*. дек. 2015. DOI: 10.1007/978-3-319-14343-9_39.

[6] Hongyun Cai, Vincent W. Zheng и Kevin Chen-Chuan Chang. *A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications*. 2017. arXiv: 1709.07604 [cs.AI].

[7] Qingyu Guo и др. *A Survey on Knowledge Graph-Based Recommender Systems*. 2020. arXiv: 2003.00911 [cs.IR].

[8] Enrico Palumbo и др. «An empirical comparison of knowledge graph embeddings for item recommendation». в: 2018. URL: http://www.eurecom.fr/publication/5576.

[9] Antoine Bordes и др. *Translating Embeddings for Modeling Multi-relational Data*. дек. 2013.

[10] R. Socher и др. «Reasoning with neural tensor networks for knowledge base completion». в: *Lake Tahoe* 2013 (янв. 2013).

[11] Ruobing Xie и др. «Representation Learning of Knowledge Graphs with Entity Descriptions». в: *AAAI*. 2016.

[12] Liang Yao, Chengsheng Mao и Yuan Luo. *KG-BERT: BERT for Knowledge Graph Completion*. 2019. arXiv: 1909.03193 [cs.CL].

[13] Jacob Devlin и др. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. arXiv: 1810.04805 [cs.CL].

[14] Xiang Wang и др. *Explainable Reasoning over Knowledge Graphs for Recommendation*. 2018. arXiv: 1811.04540 [cs.IR].