МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
"МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)"

ФАКУЛЬТЕТ ИННОВАЦИЙ И ВЫСОКИХ ТЕХНОЛОГИЙ
КАФЕДРА АНАЛИЗА ДАННЫХ

Квалификационная работа на соискание степени магистра
по направлению 01.04.02 «Прикладная математика и информатика»
НА ТЕМУ:

ТЕМАТИЧЕСКИЙ РАЗВЕДОЧНЫЙ ИНФОРМАЦИОННЫЙ ПОИСК

Студент _____ Янина А.О.

Научный руководитель д.ф-м.н _____ Воронцов К.В.

Зам. зав. кафедрой д.ф-м.н, профессор _____ Бунина Е.И.

МОСКВА, 2018

Содержание

Часть 1

Введение

Выбор релевантного материала из большого корпуса статей — распространенная проблема, с которой сталкивался любой ученый или исследователь. Информационный поиск позволяет упростить и частично автоматизировать подобные сценарии. Обычно задачи информационного поиска делят на два больших класса: поиск по четко сформулированному лаконичному запросу (known-item search) и разведочный поиск (exploratory search).

Традиционные информационно-поисковые техники фокусируются в основном на поиске по четкому короткому запросу (1; 2). Разведочный поиск изучен гораздо хуже. Такой поиск применяется, когда информационная потребность пользователя четко не определена. В качестве примеров подобных ситуаций можно привести желание разобраться совершенно новой для пользователя смежной профессиональной области или понять, где находится передний край науки по какому-либо вопросу. В таком случае сценарий поиска оказывается значительно сложнее одного-двух запросов в поисковую систему. Решение таких задач требует больших временных затрат от человека, автоматизация этого процесса нетривиальна. Этому способствует ряд причин: цели разведочного поиска зачастую неточно сформулированы, пользователь может не знать ключевых слов для поиска и иметь недостаточно знаний в предметной области, поисковые мотивы и требования к найденным документам могут меняться в процессе поиска.

Таким образом, разведочный поиск — парадигма поиска, в которой в качестве поискового запроса описывается в произвольной форме поисковая потребность пользователя, а не задается четкий короткий текстовый запрос. Формально в таком случае запросом может быть текст, состоящий из нескольких наиболее

информативных абзацев, выбранных из разных статей, имеющих отношение к теме поиска. Такой способ формулировки текста запроса позволяет сделать разведочный поиск более гибким по сравнению с обычным полнотекстовым. Разведочный поиск включает в себя задачи систематизации знаний, исследования, суммаризации, сравнительного анализа информации из разных источников, синтеза и перефразирования документов (3). Комплексных систем, позволяющих быстро и эффективно решать данные задачи данный момент не существует.

Целью данной работы является разработка нового метода решения задач разведочного исследовательского поиска. Тематическое моделирование рассматривается как одна из ключевых технологий для создания системы разведочного поиска. В рамках поставленной цели необходимо решить следующие задачи:

1. Построить тематическую модель для осуществления разведочного поиска.
2. Показать, что учет дополнительных модальностей улучшает качество разведочного поиска.
3. Предложить методику оценивания качества разведочного поиска.
4. Показать, что использование иерархических тематических моделей позволяет значительно повысить точность поиска.
5. Разработать технологию тематического поиска для решения задач разведочного поиска и показать его преимущество перед полнотекстовым поиском.

Тематическое моделирование — способ построения модели коллекции текстовых документов, которая определяет, к каким темам относится каждый из документов и какие слова или словосочетания образуют каждую тему (4). Вероятностная тематическая модель (probabilistic topic model) коллекции текстовых документов описывает каждую тему дискретным распределением на множестве терминов, каждый документ — дискретным распределением на множестве тем (5; 6; 7). В хорошей модели каждая тема является семантически однородной, что позволяет использовать тематические вектора документов для решения прикладных задач (информационный поиск, рекомендательные системы, визуализация текстовых данных).

В данном исследовании предлагается использовать аддитивную регуляризацию тематических моделей (APTM) (8). APTM позволяет строить модели, удовлетворяющие нескольким ограничениям одновременно путем добавления специальных условий-регуляризаторов в модель.

С точки зрения практического исполнения исследование демонстрирует возможности подхода ARTM на примере библиотеки с открытым кодом BigARTM. В ней реализованы регуляризаторы для построения тематической модели и метрики качества, необходимые для вычислительного эксперимента. С помощью средств BigARTM были построены тематические модели статей коллективного блога Хабрахабр.ру (русский язык) и TechCrunch.com (английский язык) с разным набором модальностей. Была разработана техника подбора параметров тематических моделей, позволяющая существенно сократить вычислительные ресурсы, затрачиваемые на перебор параметров по сетке. Кроме того, были построены различные по структуре тематические модели: одноуровневые и иерархические.

Описанные методы были использованы при построении тематической разведочной поисковой системы (см. 2.4). Затем с помощью разработанной методики оценивания качества разведочного поиска было проведено сравнение и выявлены преимущества тематического поисковика перед стандартным полнотекстовым поиском. Автоматический тематический поиск сравнивался как с ручным поиском с участием асессоров, так и с рядом моделей-образцов (бейзлайнов). (4).

В главе 5.2 подробно описана концепция разведочного поиска и представлен обзор существующих методов. Кроме того, в этой главе описывается разработанный алгоритм тематического поиска и методика оценивания качества разведочного поиска. В главе 2.4. Далее в главе 3 сформулирована постановка задачи тематического моделирования, описаны различные подходы к построению моделей (латентный семантический анализ, аддитивная регуляризация тематический моделей) и описана технология настройки моделей для ее последующего применения в разведочном поиске. В главе 4 представлены основные результаты экспериментов по оцениванию качества разработанного тематического поиска. Сравнивается классический и каскадный подход к поиску. В заключении (5.1) сформулированы основные выводы и результаты работы, а также намечен план дальнейших исследований.

Часть 2

Разведочный поиск информации

2.1 Концепция

Разведочный поиск представляет собой класс задач, связанных с исследованием и систематизацией знаний, накоплением и переработкой информации, суммаризацией текста и т.д. Не имея под рукой удобного инструмента для решения перечисленных задач, пользователь действует следующим образом: формулирует короткий текстовый запрос, получает список релевантных отранжированных документов, анализирует результаты выдачи и если выдача его не устраивает, переформулирует запрос, чтобы выделить более специфичную тему или направить поиск в другом направлении. Этот алгоритм может повторяться несколько раз до тех пор, пока поисковая выдача не позволит решить поставленную задачу.

Исследования показывают (9), что пользователи используют длинные специализированные запросы менее, чем в половине случаев. Даже тогда, когда пользователь точно знает, что нужно найти, он избегает использования длинных или сложных запросов, так как по опыту знает, что по простым и понятным запросам шанс найти релевантную информацию значительно выше. Такой подход приводит к тому, что информационная потребность прорабатывается недетально и и запрос остается неточно сформулированным. В результате, пользователи вынуждены тратить много времени и сил на постепенное изучение вопроса, запрос за запросом сужая круг поиска и приближаясь к необходимому ответу (9).

Зачастую информационную потребность сложно сформулировать в виде короткого списка ключевых слов. Например, если пользователю необходимо узнать последние достижения в конкретной научной области или быстро разобраться

в смежной области знаний, одного-двух запросов в Google или Yandex может быть недостаточно для получения исчерпывающего ответа. Данная проблема сохраняется, если поисковый запрос плохо сформулирован или пользователь не знаком с предметной областью (не знает ключевых слов, понятий, не разбирается в теме поиска).

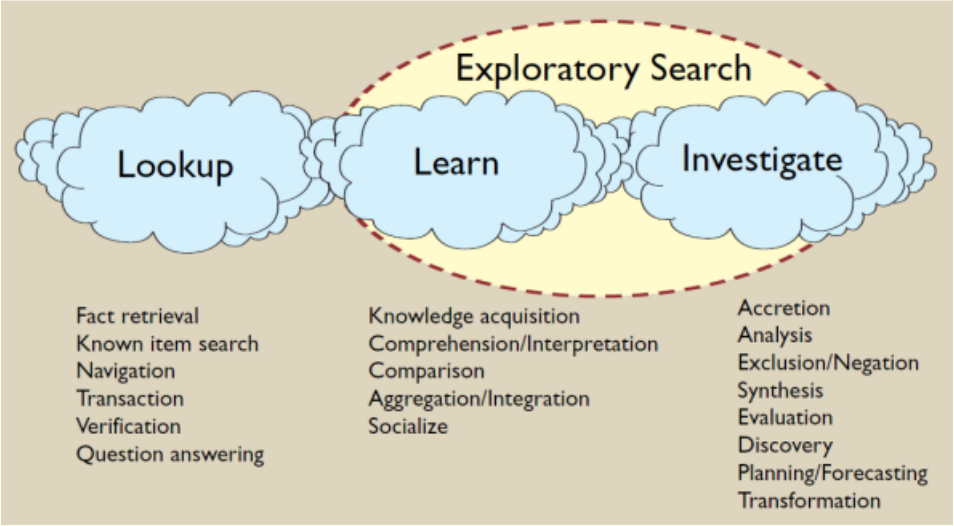

Рис. 2.1: Иллюстрация концепции разведочного поиска

Разведочный поиск — парадигма поиска, в которой в качестве поискового запроса задается не четко сформулированный текстовый запрос, а только обозначается тема, возможно, достаточно широко (10). Таким образом, мы сталкиваемся с необходимостью помочь пользователю решить его поисковую задачу, когда ключевые слова для поиска не определены и неизвестен четкий результат поиска. Когда мы осуществляем стандартный поиск по короткому запросу, обычно требуемый результат поиска четко определен. Например, по запросу "улица Строителей 6к3" мы хотим увидеть карту с обозначенной точкой геолокацией дома. Запрос в разведочном поиске не предполагает одного четкого документа, в котором должен содержаться ответ, а требует получить дорожную карту предметной области, изучение которой должно помочь пользователю разобраться в теме.

Разведочный поиск представляет собой переход от аналитического подхода к нахождению соответствий между запросами и документами и полностью автоматизированного полнотекстового поиска по коротким запросам (запросам в поисковых системах).

Разведочный поиск часто имеет итерационную структуру, когда пользователю приходится несколько раз переформулировать свой запрос, раз за разом уточняя

посиковую потребность. Цель разведочного поиска состоит в том, чтобы понять, исследовать новую для себя тему. Таким образом, акцент делается не на результате поиска, а на знаниях, приобретаемых в процессе поиска.

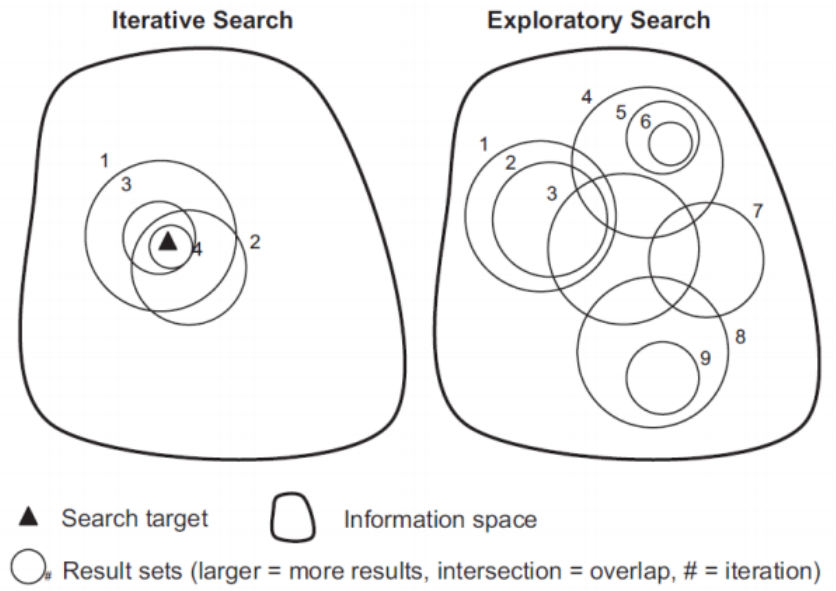

Рис. 2.2: Сравнение поиска по четкому запросу и разведочного поиска

При решении задач разведочного поиска нам могут помочь различные инструменты (10), например, дорожная карта предметной области (визуализация релеватных документов, в которой тематически схожие тексты объединены в кластера), тематические иерархии, суммаризации, текстовые парафразы и энциклопедические статьи.

2.2 Обзор существующих подходов

На текущий момент разведочный поиск изучен не так хорошо, как полнотекстовый поиск по короткому текстовому запросу. Рассмотрим несколько методов решения задач разведочного поиска.

В 2006 году Г.Марчионини в своей статье (1) изложил парадигму разведочного поиска и сформулировал основные характеристики этого типа поиска. Он разработал систему Relation Browser (RB), позволяющую осуществлять разведочный поиск по государственным статистическим веб-сайтам. Эта система умеет искать по большим базам данных и в качестве ответа на запрос выдает упорядоченную коллекцию документов для дальнейшего изучения. RB предоставляет пользователю ограниченный функционал: искать можно только по следующим параметрам: тема, метка времени, формат входного файла. Количество возможных значений каждого параметра ограничено. В (1) показана эффективность этой системы по сравнению с полнотекстовым поиском только для решения узконаправленной задачи поиска по строго заданному реестру сайтов.

В 2012 году С.Голденберг предложил две системы разведочного поиска (11). Технология позволяет группировать документы на основе анализа их метаданных. Связи между документами можно изобразить с помощью node-link диаграммы для упрощения дальнейшего анализа. Инструмент позволяет итеративно оценивать информационные нужны пользователя и сужать область поиска от запроса к запросу.

В 2014 году на конференции CIKM был представлен инструмент для упрощения поиска по Википедии и Yahoo Answers под названием DEESSE (12). Эта система представляет информацию в виде графа, где для каждой ноды указана метаинформация: тема документа, ее категория, уровень качества материала и научная точность документа. Вершины графа соединяются исходя из их тематической схожести. Для обработки пользовательских запросов используются предподсчитанные тематические кластера: сначала ищется наиболее близкий к запросу кластер, затем внутри него отбираются наиболее релевантные документы. Создатели DEESSE отказались от привычной для поисковых систем организации выдачи — списка отранжированных по релевантности документов. Вместо этого выдача отображается в виде тематических кластеров, что упрощает поиск интересующей информации среди найденных документов. DEESSE поддерживает английский и испанский языки и умеет осуществлять кросс-язычный поиск.

На данный момент не существует популярной и удобной системы разведочного поиска. Такая система к тому же должна быть мультиязычной и поддерживать разные источники информации. Существуют только решения для смежных, как правило узконаправленных, задач. Например, задача ранжирования документов в порядке удобном для чтения достаточно хорошо изучена. В (13) предлагается прототип такой системы. Статьи группируются в виде дерева: тексты более общей тематики находятся в корне, листья дерева — статьи узкой направленности. Такое представление информации взамен стандартной выдачи поисковика помогает более широко представить себе исследуемую область и, как следствие, решать задачи разведочного поиска быстрее и эффективнее.

Еще один пример решения задачи смежной с разведочным поиском — это системы закладок. Они помогают группировать тексты, сортировать статьи и упорядочивать найденные документы. В (14) построен теговый поисковый сервис, который объединяет в себе идеи разведочного поиска и системы закладок. В этой работе применяется коллаборативная фильтрация для рекомендации тегов пользователям. Пользовательские профили из разных интернет-сервисов интегрируются в одну систему, которая помогает пользователям решать задачи поиска и систематизации информации путем рекомендации им статей с релевантными тегами.

Описанные в обзоре системы решают задачи исследования и систематизации информации, приближаясь, но не решая полностью задачу разведочного поиска информации. Крупных и популярных систем тематического поиска на данный момент не существует.

2.3 Постановка задачи разведочного поиска

Глобально задачу разведочного поиска можно описать как задачу поиска и систематизации большого количества информации по неизвестной теме, У такого поиска зачастую неявно сформулирован ожидаемый результат, а также есть трудности с заданием четкого запроса. Поисковая потребность пользователя может быть сформулирована неточно или общо. На данном этапе важно понять, чем является запрос в задачах разведочного поиска и формализовать саму задачу поиска.

Запрос — это описание поискового намерения пользователя. Он может задаваться в виде статьи или набора статей по теме поиска, плана поиска или короткого текста с описанием поисковой задачи. В нашем эксперименте запрос — это текст объема

примерно на один лист А4. В качестве типовой модельной ситуации тематического поиска мы рассматриваем поручение, которое менеджер информационного агентства мог бы дать своему подчинённому. Менеджер мог обозначить направление поиска собрав из разных источников несколько абзацев релевантного текста и попросить своего коллегу составить отчет по указанной тематике. Будем считать, что подчинённый потратит порядка часа времени на выполнение этого задания. В процессе поиска он может пользоваться любыми средствами поиска: Яндекс, Google, любые системы ссылок и рекомендаций, категоризаторы и рубрикаторы.

Мы предлагаем упростить и автоматизировать решение подобных задач (15; 16). Мы разработали метод для решения задач тематического разведочного поиска, основанный на мультимодальном тематическом моделировании.

2.4 Алгоритм тематического поиска

Рассмотрим алгоритм разведочного поиска информации по большой текстовой коллекции. Пусть мы сформировали текстовую коллекция запросов для разведочного поиска. Обозначим коллекцию запросов как Q. Запрос в терминах разведочного поиска —это текст, отражающих поисковую потребность пользователя.

Первым этапом алгоритма является построение тематической модели на датасете, состоящим из документов текстовой коллекции и текстов запросов из Q. Тематическая модель может быть мультимодальной, иерархической, темпоральной, иметь любое количество тем или регуляризаторов.

Второй этап работает онлайн с запросами, приходящими в поисковую систему. Если запрос содержится в Q, то найти его тематический профиль можно по формуле 2.1:

$$p(t|q) = \Theta[q_i] \tag{2.1}$$

В 2.1 q_i — номер запроса q в текстовой коллекции (тематический профиль запроса — соответствующая строка из матрицы Θ).

Если запроса в коллекции Q не оказалось, то сначала надо тематизировать его: определить, какие темы присутствуют в текстовом запросе и сформировать тематический вектор запроса.

Затем среди тематических профилей документов ищем k близких к профилю запроса векторов. Близость тематических профилей можно оценивать разными

способами: с помощью евклидова расстояния, косинусной меры, манхеттенского расстояния, КЛ-дивергенции или расстояния Хеллингера. Дальнейшие эксперименты покажут, что косинусная мера позволяет получить наилучшее качество поиска, поэтому рассмотрим здесь именно ее. Косинусная мера для двух векторов a и b вычисляется по формуле:

$$cossim(a,b) = \frac{a \cdot b}{||a||||b||} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2}} \tag{2.2}$$

Полученный список документов нужно разметить на релевантные и нерелевантные, а затем оценить качество тематического поиска по метрикам precision@k, recall@k.

2.5 Алгоритм оценивания качества разведочного поиска

Мы разработали методику для оценивания качества тематического разведочного поиска (подробно эта методика описана в (15; 16)). Она основана на выполнении вручную разведочного поиска с использованием любых поисковых средств и сравнении полученных результатов с выдачей автоматического тематического поисковика. Алгоритм оценки предполагает участие асессоров, которые должны выполнить два задания. Для того, чтобы уменьшить влияние человеческого фактора на результаты работы, предлагается размечать выборку с перекрытием.

Подготовка к тестированию Для начала организатор тестирования должен составить множество запросов, соответствующих тематической направленности коллекции. Запрос формируется из релевантных фрагментов текста документов коллекции или сторонних источников. В результате должно получиться описание поисковой потребности пользователя, достаточно точно отражающее тематику запроса. Формально запрос должен выглядеть как текст примерного объема $1-2$ страницы формата $A4$. При этом каждый запрос должен быть достаточно емким и точным, чтобы асессор мог понять его смысл, и достаточно полным, чтобы поисковая потребность могла интерпретироваться однозначно. Запрос может содержать или не содержать заголовок. В 2.1 представлены заголовки запросов разведочного поиска.

Табл. 2.1: Заголовки запросов для разведочного поиска

Алгоритмы раскраски графов	IBM Watson
Рекомендательная система Netflix	3D-принтеры
Методики быстрого набора текста	CERN-кластер
Космические проекты Илона Маска	AB-тестирование
Технологии Hadoop MapReduce	Облачные сервисы
Self-driving Google car	Контекстная реклама
Криптосистемы с открытым ключом	Марсоход Curiosity
Обзор платформ онлайн-курсов	Видеокарты NVIDIA
Data Science Meetups в Москве	Распознавание образов
Образовательные проекты mail.ru	Сервисы Google scholar
Межпланетная станция New horizons	MIT MediaLab Research
Языковая модель word2vec	Платформа Microsoft Azure

Процесс формирования запроса можно проиллюстрировать реальной задачей, которая может ставить перед подчиненным менеджер информационного или рекламного агентства: нужно поручить сотруднику разобраться в новой для него теме. Менеджер максимально быстро накидывает фрагменты текста из релевантных статей и отчетов и передает полученный сырой текст подчиненному для доработки и составления подробного отчета по теме.

Задание 1. Каждому асессору предлагается запрос (или несколько запросов) и инструкция к выполнению задания. В инструкции прописана модельная ситуация и описан процесс разведочного описка. Асессору предлагается провести разведочный поиск информации самостоятельно: найти в обозначенной коллекции как можно больше релевантных документов используя любые поисковые средства (Google, Yandex, базы данных, рубрикаторы). Кроме того, асессор замеряет время потраченное на поиск.

Перекрытие (число асессоров m на один запрос), с которым происходит разметка —важный параметр методики. Чем больше число m, тем объективнее будут оценки полноты поиска. Важно обратить внимание на сложность задания, а значит и предполагаемую стоимость выполнения задания на любой краудсорсинговой платформе. Баланс между стоимостью разметки и объективностью полученных результатов подбирается в каждом конкретном случае отдельно.

Задание 2. После выполнения задания 1 асессору предлагается посмотреть на результаты тематического поиска по тому же самому запросу и разметить выдачу

тематического поисковика. Используется стандартная шкала оценивания документа на релевантность (Vital, Useful, Relevant+, Relevant-, Irrelevant).

Документ считается релевантным запросу, если хотя бы один асессор нашёл этот документ или если этот документ был найден тематическим поиском и хотя бы n из m асессоров отметили его как релевантный. Параметры методики n и m подбираеются для каждого эксперимента отдельно.

Для каждого запроса определим две меры качества поиска: *точность* Precision@k — доля релевантных документов среди первых k найденных; *полнота* Recall@k — доля k первых найденных релевантных документов среди всех релевантных.

Теперь введем более формально метрики Precision@k и Recall@k. Для этого введем обозначения:

- TP (true positive) — правильно рекомендованные статьи, которые должны понравиться пользователю,
- TN (true negative) — статьи, которые не нравятся пользователю и не были ему рекомендованы системой,
- FP (false positive) — ошибки первого рода, т.е. не интересные пользователю статьи, рекомендованные системой,
- FN (false negative) — ошибки второго рода, т.е. не рекомендованные пользователю статьи, которые оказались интересными.

Запишем формулы для метрик Precision (P) и Recall (R) с учетом введенных обозначений:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.3}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.4}$$

Для больших корпусов документов Precision и Recall перестают быть информативными, так как выдача может содержать несколько тысяч релевантных статей. В этом случае для оценки качества лучше использовать Precision at k (Precision@k, P@k) и Recall at k (Recall@k, R@k) - метрики, применимые к первым наиболее популярным k документам. Так, $P@k$ — доля релевантных документов (тех, которые

оказались интересны пользователю, рекомендации, которые он просмотрел) среди первых k документов из отранжированного списка рекомендаций. $R@k$ — доля релевантных документов из топ-k списка рекомендаций среди всех релевантных документов по данному запросу.

Для измерения качества тематического поиска точность и полнота усредняются по всем запросам. Для измерении качества асессорского поиска точность и полнота усредняются ещё и по асессорам. Агрегированная оценка качества поиска F_1-*мера* определяется как среднее гармоническое точности P и полноты R: $F_1 = \frac{P+R}{2PR}$.

Часть 3

Тематическое моделирование

Системы полнотекстового поиска позволяют находить документы по словам, в разведочном поиске предлагается вместо слов использовать темы (17). Таким образом, совокупность тем текста выступает в роли короткого запроса стандартной системы полнотекстового поиска. Используя такую интерпретацию тем, будем строить инвертированный индекс, только не по словам, а по темам. Поиск присутствующих в запросе тем по проиндексированной коллекции осуществляется значительно быстрее, чем при прямом сравнении тематических векторов документа и запроса. Для того, чтобы составить тематический профиль документа и запроса, будем использовать тематическое моделирование.

3.1 Постановка задачи вероятностного тематического моделирования

Пусть D — коллекция (множество текстовых документов), а W — словарь (множество всех употребляемых в документах из коллекции терминов). В качестве терминов могут выступать не только слова, но и словосочетания, биграммы, буквенные и словесные n-граммы. Каждый документ $d \in D$ представляет из себя последовательность терминов $(w_1, ..., w_{n_d}) \subset W$, где каждому термину соответствует число его вхождений n_{dw}. Таким образом, матрица частот F для текстовой коллекции D будет выглядеть так:

$$F = (f_{wd})_{W \times D} \tag{3.1}$$

$$f_{wd} = \frac{n_{dw}}{n_d} \tag{3.2}$$

Примем предположение, что существует конечное множество тем T, описывающее множество документов D. Коллекция документов рассматривается как случайная и независимая выборка троек $(w_i, d_i, t_i), i = 1..n$ из дискретного распределения $p(w, d, t)$ на конечном вероятностном пространстве $W \times D \times T$. При этом термины и документы — это наблюдаемые переменные, а тема документа является скрытой переменной. В данной модели используется гипотеза «мешка слов», согласно которой порядок, в котором термины встречаются в документе, не важен, а также гипотеза «мешка документов» (порядок документов в коллекции не важен). Для учета информации о взаимном расположении слов в документе будем считать не только частоты встречаемости отдельных слов, но и частоты словосочетаний.

Введем еще несколько понятий. $p(w|t)$ — вероятность встречания термина $w \in W$ в теме $t \in T$, $p(t|d)$ — вероятность встречания темы $t \in T$ в документе $d \in D$. Зная эти вероятности, получаем матрицу терминов тем Φ (3.3) и матрицу тем документов Θ (3.4):

$$\Phi = p(w|t)_{W \times T} \tag{3.3}$$

$$\Theta = p(t|d)_{T \times D} \tag{3.4}$$

Сформулируем окончательно задачу тематического моделирования: по заданной коллекции D найти множество тем T и оценить параметры модели $\phi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$. Задача сводится к поиску матричного разложения заданной матрицы частот в виде произведения неизвестных матриц терминов тем (3.3) и тем документов (3.4):

$$F \approx \Theta_{W \times T} \times \Theta_{T \times D} \tag{3.5}$$

3.2 Вероятностный латентный семантический анализ

В вероятностном латентном семантическом анализе (PLSA) (5) для оценки матриц Φ и Θ предлагается максимизировать логарифм правдоподобия выборки при ограничениях неотрицательности и нормировки столбцов этих матриц(3.6):

$$L(\Phi,\Theta) \to \max_{\Phi,\Theta}$$

$$L(\Phi,\Theta) = ln \prod_{d\in D}\prod_{w\in d} p(w|d)^{n_{dw}} = \sum_{d\in D}\sum_{w\in d} n_{dw} ln \sum_{t\in T} \phi_{wt}\theta_{td} \tag{3.6}$$

$$\sum_{w\in W} \phi_{wt} = 1, \quad \phi_{wt} \geq 0, \quad \sum_{t\in T} \theta_{td} = 1, \quad \theta_{td} \geq 0.$$

Поставленная задача решается с помощью итерационного EM-алгоритма. На E-шаге алгоритм вычисляет условные вероятности всех тем для каждой пары термин-документ по текущим значениям (обновленным на предыдущем M-шаге):

$$p(t|d,w) = \frac{p(w|t)p(t|d)}{p(w|d)} \tag{3.7}$$

На M-шаге пересчитываются новые приближения параметров ϕ_{wt} и θ_{td} (3.9):

$$n_{dwt} \approx n_{dw} \cdot p(t|d,w) \tag{3.8}$$

$$\begin{aligned} \phi_{wt} &= \frac{n_{wt}}{n_t} & \theta_{td} &= \frac{n_{dt}}{n_d} \\ n_{wt} &= \sum_{d\in D} n_{dwt} & n_t &= \sum_{w\in W} n_{wt} \\ n_{dt} &= \sum_{w\in d} n_{dwt} & n_d &= \sum_{t\in T} n_{dt} \end{aligned} \tag{3.9}$$

Начальные приближения ϕ_t и θ_d можно задавать отнормированными равномерно распределенными случайными векторами или инициализировать случайными векторами.

3.3 Аддитивная регуляризация тематических моделей

В общем виде задача тематического моделирования имеет бесконечно много решений. Если $F = \Phi\Theta$ — решение задачи, то для всех невырожденных матриц S, при которых матрицы $\Theta' = S^{-1}\Theta$ и $\Phi' = \Phi S$ являются стохастическими, $F = (\Phi S)(S^{-1}\Theta)$ также является решением. Данную проблему можно решить с помощью регуляризации (добавлять к логарифму правдоподобия штрафное слагаемое, которое сужает множество решений). Мы будем использовать метод аддитивной регуляризации (4) .

Допустим, что наряду с правдоподобием L требуется максимизировать еще n критериев $R_i(\Phi, \Theta)$, $i = 1, 2, \ldots, n$, называемых регуляризаторами. В байесовских методах обучения тематических моделей (5; 18; 19) регуляризатор $R(\Phi, \Theta)$ интерпретируется как логарифм априорного распределения, а оптимизационная задача соответствует принципу максимума апостериорной вероятности. На практике зачастую нужно иметь возможность совмещать большое число регуляризаторов. Байесовский вывод оказывается слишком громоздким для совмещения в одной модели более, чем двух-трех дополнительных критериев-регуляризаторов. Теория аддитивной регуляризации тематических моделей (APTM) позволяет решить эту проблему (4), снимая ограничение на вероятностную природу регуляризатора. Таким образом, мы максимизируем взвешенную сумму критириев-регуляризаторов $R_i(\Phi, \Theta)$ с логарифмом правдоподобия $L(\Phi, \Theta)$ (3.10):

$$\begin{gathered} L(\Phi, \Theta) + \sum_{i=1}^{r} \tau_i R_i(\Phi, \Theta) \to \max_{\Phi, \Theta} \\ \sum_{w \in W} \phi_{wt} = 1, \quad \phi_{wt} \geq 0, \quad \sum_{t \in T} \theta_{td} = 1, \quad \theta_{td} \geq 0. \end{gathered} \tag{3.10}$$

В указанной выше формуле τ_i — неотрицательный коэффициент регуляризации.

Эта задача решается с помощью EM-алгоритма (см. 3.6). Добавление нового регуляризатора приводит к изменению формул для M-шага на аналогичную добавку. Это позволяет строить модели с любым количеством регуляризаторов.

$$\begin{aligned} \phi_{wt} &= \frac{n_{wt}}{n_t} + \phi_{wt} \frac{\partial R(\Phi, \Theta)}{\partial \phi_{wt}} \\ \theta_{td} &= \frac{n_{dt}}{n_d} + \theta_{td} \frac{\partial R(\Phi, \Theta)}{\partial \theta_{td}} \end{aligned} \tag{3.11}$$

В формулах 3.11:

$$n_{wt} = \sum_{d \in D} n_{dwt} \qquad n_t = \sum_{w \in W} n_{wt}$$
$$n_{dt} = \sum_{w \in d} n_{dwt} \qquad n_d = \sum_{t \in T} n_{dt} \tag{3.12}$$

Здесь стоит упомянуть теорему, в которой приводятся явные формулы для подсчета параметров модели ϕ_{wt}, θ_{td}.

Теорема 3.1. *Пусть $R(\Phi, \Theta)$ непрерывно дифференцируема и точка (Φ, Θ) является локальным экстремумом задачи 3.10. Обозначим $p_{tdw} = p(t|d, w)$. Тогда для любой темы t и любого документа d выполняется система уравнений:*

$$p_{tdw} = norm_{t \in T}(\phi_{wt}\theta_{td}) \tag{3.13}$$

$$\phi_{wt} = norm_{w \in W}\left(n_{wt} + \phi_{wt}\frac{dR}{d\phi_w t}\right) \qquad n_{wt} = \sum_{d \in D} n_{dw} p_{tdw} \tag{3.14}$$

$$\theta_{td} = norm_{t \in T}\left(n_{td} + \theta_{td}\frac{dR}{d\theta_t d}\right) \qquad n_{td} = \sum_{w \in d} n_{dw} p_{tdw}, \tag{3.15}$$

где $normx_i = \dfrac{max(x_i, 0)}{\sum\limits_{j \in I} max(x_j, 0)}$, *для всех* $i \in I$.

Доказательство. Доказательство основано на применении условий Каруша-Куна-Такера и приводится в (4). □

Наиболее часто в моделях используются регуляризаторы декоррелирования и разреживания (сглаживания). Остановимся на них более подробно.

Сглаживающий регуляризатор минимизирует кросс-энтропию между столбцами $\vec{\phi}_t$ и фиксированным распределением $\vec{\beta} = (\beta_w \colon w \in W)$ и кросс-энтропию между столбцами $\vec{\theta}_d$ и распределением $\vec{\alpha} = (\alpha_t \colon t \in T)$:

$$R(\Phi, \Theta) = \beta_0 \sum_{t \in T} \sum_{w \in W} \beta_w \ln \phi_{wt} + \alpha_0 \sum_{d \in D} \sum_{t \in T} \alpha_t \ln \theta_{td}, \tag{3.16}$$

Здесь вектора $\beta_0\vec{\beta}$ и $\alpha_0\vec{\alpha}$ можно интерпретировать как гиперпараметры распределения Дирихле в контексте байесовского вывода для тематической модели. β_0 and α_0 интерпретируются как коэффициенты регуляризации в рамках APTM подхода. Выбор равномерного распределения в качестве $\vec{\beta}$ и $\vec{\alpha}$ соответствует выбору в качестве априорного распределения симметричного распределения Дирихле, которое часто используется в экспериментах с моделью LDA.

Разреживающий регуляризатор имеет такую же форму, что и сглаживающий регуляризатор 3.16, отличается только наличие знака минус перед коэффициентами

β_0 и α_0 (20). Разреживающий регуляризатор максимизирует кросс-энтропию заставляя распределения $\vec{\phi}_t$ и $\vec{\theta}_d$ быть настолько непохожими на $\vec{\beta}$ and $\vec{\alpha}$, насколько это возможно.

Декоррелирующий регуляризатор заставляет темы быть различными минимизируя сумму ковариаций между тематическими векторами $\vec{\phi}_t$:

$$R(\Phi) = -\sum_{t,s \in T} \sum_{w \in W} \phi_{wt}\phi_{ws}.$$

Кроме того, этот регуляризатор стимулирует увеличение разреженности, группирует стоп-слова и слова общей лексики в обособленные группы. Доказано, что комбинация этих трех регуляризаторов повышает интерпретируемость тем (20; 21; 22).

3.4 Мультимодальное тематическое моделирование

Зачастую в тематическом моделировании используется не только сам текст документа, но и некоторая дополнительная информация. Такие данные часто называют метаинформацией. Например, фамилия автора статьи, теги, комментарии, отметки времени, лайки, метки классов или категорий являются метаинформацией и могут использоваться при построении моделей. Вхождение элементов каждой модальности рассматривается точно так же, как вхождение терминов в текст. Тематическая модель, при построении которой использовались метаданные, называется мультимодальной. Каждая модальность $m \in M$ имеет свой словарь W_m. Первая модальность соответствует терминам (словам, биграммам или словосочетаниям), остальные — метаданным.

Введем несколько понятий для задачи мультимодального тематического моделирования. $\forall m$ $p(w,t)_m$ — вероятность встречания термина $w \in W^m$ в теме $t \in T$, $p(t,d)$ — вероятность встречания темы $t \in T$ в документе $d \in D$. Для каждой модальности $m \in M$ есть своя матрица терминов тем Φ_m:

$$\Phi_m = (p(w|t))_{W^m \times T} \quad \forall m \in M \tag{3.17}$$

Объединение матриц Φ_m, записанных в столбец, дает нам общую матрицу терминов тем модели:

$$\Phi = p(w|t)_{W \times T} \tag{3.18}$$

Матрица тем документов имеет такой же вид, как и матрица для унимодальной модели:

$$\Theta = (p(t|d))_{T \times D} \tag{3.19}$$

Запишем постановку задачи мультимодального тематического моделирования (3.20), представив логарифм правдоподобия в виде суммы по модальностям:

$$\begin{gathered} L(\Phi,\Theta) + \sum_{i=1}^{r} \tau_i R_i(\Phi,\Theta) \to \max_{\Phi,\Theta} \\ L(\Phi,\Theta) = \sum_{m \in M} L_m(\Phi_m,\Theta) = \sum_{m \in M} L(\Phi_m,\Theta) = \\ = \sum_{m \in M} ln \prod_{d \in D} \prod_{w \in d} p(w|d)^{n_{dw}} = \sum_{m \in M} \sum_{d \in D} \sum_{w \in d} n_{dw} ln \sum_{t \in T} \phi_{wt}\theta_{td} \\ \forall m \in M \sum_{w \in W^m} \phi_{wt} = 1, \quad \phi_{wt} \geq 0, \quad \sum_{t \in T} \theta_{td} = 1, \quad \theta_{td} \geq 0. \end{gathered} \tag{3.20}$$

Первое слагаемое из суммы $\sum_{m \in M} L_m(\Phi_m, \Theta)$ соответствует модальности терминов. Остальные слагаемые можно интерпретировать как регуляризаторы сооответствующих модальностей. Добавим в эту сумму коэффициенты регуляризации:

$$L(\Phi, \Theta) = \sum_{m \in M} \tau_m L_m(\Phi_m, \Theta) \to \max_{\Phi, \Theta} \tag{3.21}$$

Эти коэффициенты позволяют варьировать вклад разных модальностей в модель.

Теорема 3.1 обобщается на мультимодальный случай.

Теорема 3.2. *Пусть $R(\Phi, \Theta)$ непрерывно дифференцируема и точка (Φ, Θ) является локальным экстремумом задачи 3.20. Обозначим $p_{tdw} = p(t|d, w)$. Тогда для любой темы t и любого документа d выполняется система уравнений:*

$$p_{tdw} = norm_{t \in T}(\phi_{wt}\theta_{td}) \tag{3.22}$$

$$\phi_{wt} = norm_{w \in W_m}\left(n_{wt} + \phi_{wt}\frac{dR}{d\phi_w t}\right) \qquad n_{wt} = \sum_{d \in D} \overline{n}_{dw} p_{tdw} \tag{3.23}$$

$$\theta_{td} = norm_{t \in T}\left(n_{td} + \theta_{td}\frac{dR}{d\theta_t d}\right) \qquad n_{td} = \sum_{w \in d} \overline{n}_{dw} p_{tdw}, \tag{3.24}$$

где $\overline{n}_{dw} = \tau_m \cdot n_{dw} \quad \forall m \in M$.

Доказательство. Доказательство аналогично доказательству 3.1 и приводится в (4). □

Таким образом, переход от одной модальности к нескольким сводится к двум дополнениям:

- Матрица Φ разбивается на блоки. Каждый из них соответствует своей модальности и нормируется отдельно.
- Исходные данные n_{dw} домножаются на коэффициенты регуляризации τ_m для каждой модальности.

3.5 Иерархическое тематическое моделирование

Иерархические тематические модели —мощный инструмент для систематизации больших текстовых коллекций, информационного поиска и разведывательного анализа данных. В иерархических моделях появляется новый параметр —- количество уровней иерархии. Для моделирования связей между уровнями в модель

вводятся параметры $\psi_{st} = p(st)$ — условные вероятности подтем в темах. Подтемы могут иметь по несколько родительских тем. Например, подтема «Social nets», может наследоваться от тем «Networking», «IT» и «Online Communication».

При построении иерархических моделей вводятся дополнительные регуляризаторы, ранее не рассмотренные в работе. Остановимся на них более подробно.

3.5.1 Регуляризатор межуровневых связей

На верхнем уровне иерархии строится обычная плоская тематическая модель. Рассмотрим процесс построения следующих уровней иерархии. Пусть модель ℓ" го уровня с множеством тем T уже построена, и требуется построить модель уровня $\ell+1$ с множеством дочерних тем S (subtopics) и бóльшим числом тем, $|S| > |T|$. Потребуем, чтобы родительские темы t хорошо приближались вероятностными смесями дочерних тем s:

$$\sum_{t\in T} n_{t\,w}\Big(p(wt)\,\Big\|_{s\in S}\,p(ws)\,p(st)\Big) = \sum_{t\in T} n_{t\,w}\Big(\tfrac{n_{wt}}{n_t}\,\Big\|_{s\in S}\,\phi_{ws}\psi_{st}\Big)\;\to\;\min_{\Phi,\Psi},$$

где $\Psi = (\psi_{st})_{S\times T}$ — *матрица связей*, которая становится дополнительной матрицей параметров для тематической модели дочернего уровня.

Регуляризатор связывает тематические модели соседних уровней ℓ и $\ell+1$ так, чтобы родительские темы ϕ_t^ℓ аппроксимировались линейными комбинациями дочерних тем ϕ_s с коэффициентами ψ_{st}:

$$R(\Phi,\Psi) = \tau\sum_{t\in T}\sum_{w\in W} n_{wt}\ln\sum_{s\in S}\phi_{ws}\psi_{st}. \tag{3.25}$$

Задача максимизации $R(\Phi,\Psi)$ с точностью до обозначений совпадает с основной задачей тематического моделирования, если считать родительские темы t *псевдодокументами* с частотами термов $\tau n_{wt} = \tau n_t \phi_{wt}$.

Реализация этого регуляризатора осуществляется следующим образом. В процессе построения родительского уровня добавляем в коллекцию $|T|$ псевдодокументов, задав им в качестве частот термов значения τn_{wt}. Матрица Ψ получится в столбцах матрицы Θ, соответствующих псевдодокументам (23).

В библиотеке этот подход реализован в виде отдельного класса `hARTM`.

3.5.2 Регуляризатор разреживания межуровневых связей

Рассмотрим редположение, что каждая тема дочернего уровня $s \in S$ имеет небольшое число связей с темами родительского уровня $t \in T$. В частности, если все распределения $p(ts)$ вырождены, то есть каждая тема s имеет только одну родительскую тему t, то вся иерархия приобретает вид дерева. Применим кросс-энтропийный регуляризатор для разреживания распределений $p(ts)$. Выражаем $p(ts)$ через ψ_{st}:

$$R(\Psi) = -\tau \sum_{s \in S} \sum_{t \in T} \frac{1}{|T|} \ln p(ts) = -\frac{\tau}{|T|} \sum_{t \in T} \sum_{s \in S} \ln \frac{\psi_{st} n_t}{\sum_z \psi_{sz} n_z}.$$

Формула M-шага для модели дочернего уровня выглядит следующим образом:

$$\psi_{st} =_{s \in S} \Big(n_{st} + \tau \Big(p(ts) - \frac{1}{|T|} \Big) \Big). \tag{3.26}$$

Согласно этой формуле, условные вероятности $p(ts)$, меньшие $\frac{1}{|T|}$, становятся ещё меньше, и при достаточно большом τ обнуляются (23).

Часть 4

Эксперименты по оцениванию качества работы тематического поиска

В этой главе мы покажем, как работает алгоритм для быстрого решения задач тематического разведочного поиска (2.4), основанный на мультимодальном тематическом моделировании. С помощью техники, описанной в 2.5 оценим качество тематического поисковика и изложим подробности построения и обучения моделей.

4.1 Описание и предобработка данных

Эксперименты проводились на двух текстовых коллекциях новостей — TechCrunch.com (английский язык) и Habrahabr.ru (русский язык). Новости из обеих коллекций имеют техническую направленность, наиболее полно представлены обзоры о новых технологиях, IT-сфере стартапах и т.д. Более полно тематическая направленность новостей иллюстрируется заголовками составленных запросов для разведочного поиска (2.1).

Коллекция TechCrunch состоит из 759324 статей. Для каждой статьи известна следующая информация (модальности):

- 11523 слов (униграмм),
- 1200000 биграмм (редко встречающиеся биграммы в процессе предобработки были удалены),

- 605 авторов,
- 184 категорий.

Коллекция Хабрахабра состоит из 175143 статей. Она включает в себя 6 модальностей:

- 10552 слов (униграмм),
- 742000 биграмм,
- 524 авторов,
- 10000 комментаторов (авторов комментариев к статьям),
- 2546 тегов,
- 123 хабов (категорий).

Предобработка обеих коллекций включала в себя удаление пунктуации, приведение всех букв к нижнему регистру, лемматизация для русского языка (с помощью библиотеки pymorphy2) и стемминг для английского языка. Кроме того, мы исключили 5% наиболее часто встречающихся слов в каждой коллекции.

4.2 Сравнение тематического поиска с асессорским

Описанную выше технику оценивания качества разведочного поиска, основанную на работе асессоров, мы применили к данным Хабрахабра и TechCrunch.

Для Хабрахабра мы сконструировали 100 запросов, копируя тематически близкие текстовые фрагменты из любых источников, кроме самого Хабрахабра (например, использовались посты со stackoverflow.com, статьи с ixbt.com и т.д.) Длина запроса варьировалась от 93 до 455, среднее число слов в запросе оказалось равным 262. Примеры заголовков запросов для разведочного поиска представлены в таблице 2.1. Каждый запрос обрабатывался с перекрытием 3. Количество релевантных статей на запрос варьировалось от 5 до 55, среднее число —25.

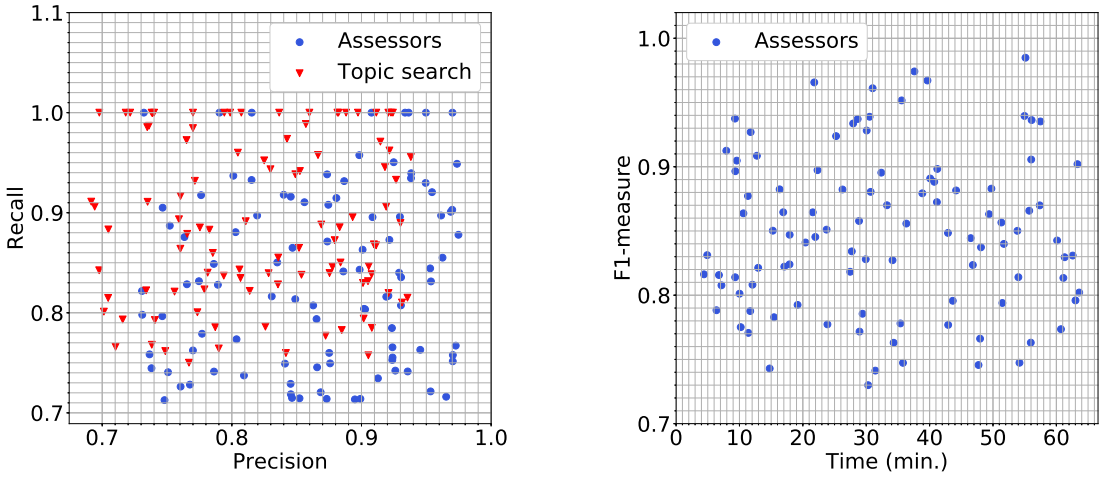

Рис. 4.1: Качество асессорского ручного и тематическоо автоматического поиска (Хабрахабр)

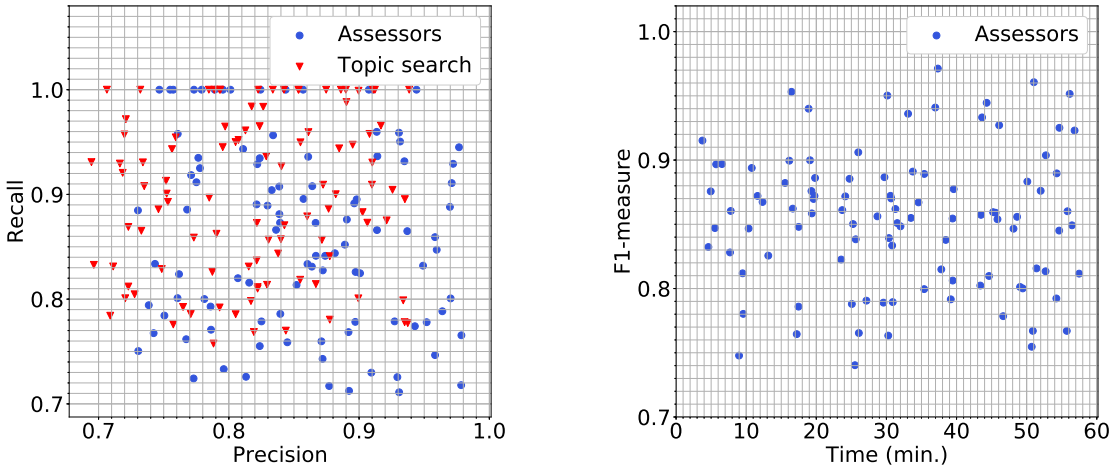

Рис. 4.2: Качество асессорского ручного и тематическоо автоматического поиска (TechCrunch)

Результаты эксперимента для коллекции Хабрахабра представлены на рис. 4.1. Точки на графике соответствуют запросам (красные треугольники обозначают автоматический тематический поиск, синие точки —ручной асессорский поиск). Мы сравнивали точность и полноту ручного и автоматического тематического поиска отдельно по каждому запросу. В среднем точность у асессорского поиска немного выше, в то время, как полнота автоматического поиска выше. Полноту автоматического поиска, равную 1.0, мы получили для 26 запросов из 100. Это означает, что тематический поисковик достаточно часто находит документы, которые асессоры пропустили при поиске вручную. Еще одно важное преимущество

автоматического поиска —это экономия времени. В среднем асессор тратил 30 минут на обработку одного запроса, автоматический поиск работает за $1-2$ секунды. Из правого графика видно, что зависимости между временем, которое асессор потратил на обработку запроса, и качеством поиска нет.

Аналогичный эксперимент был проведен на статьях TechCrunch.com. Мы сформировали 100 запросов. Длина запросов варьировалась от 75 до 392 слов, средняя длина запроса —195 слов. Аналогично эксперименту на данных Хабрахабра, каждый запрос обраатывался 3 асессорами, а затем результаты усреднялись. Среднее число релевантных статей на запрос — 32.

Результаты эксперимента представлены на рис. 4.2. Автоматический тематический поиск позволяет получить более высокую полноту по сравнению с ручным поиском, но немного происгрывает в точности.

В обоих экспериментах была подсчитана стат.значимость различия в точности и полноте для автоматического и ручного асессорского поиска. Использовался критерий знаковых рангов Вилкоксона (критерий Вилкоксона для связных выборок, the Wilcoxon signed-rank test). Для всех тестов p-value был меньше, чем 0.01. Отсюда делаем вывод, что датасета из 100 запросов достаточно, чтобы сравнивать качество поиска ручного и автоматического поиска.

4.3 Сравнение тематического поиска с бейзлайнами

Кроме сравнения разработанного тематического поиска с асессорским, мы провели несколько экспериментов по сравнению тематического поисковика с бейзлайнами. В качестве основного бейзлайна мы выбрали TF-IDF модель. Сначала была применена лемматизация для текстов на русском языке и стемминг для англоязычных текстов. Затем мы получили TF-IDF вектора из документов и запросов с помощью векторайзера из библиотеки sklearn. В качестве результата поиска мы возвращали топ-k документов, TF-IDF вектора которых были наиболее близки к вектору запроса. TF-IDF —простой, но достаточно сильный бейзлайн, так как в нем используется полная информация о частоте встречаемости слов в документе, тогда как в тематическом поиске мы располагаем только низкоразмерной аппроксимацией матрицы частот слов. Чтобы сделать бейзлайн еще сильнее, мы рассматривали частоты встречаемости не только слов, но и категорий. Кроме TF-IDF подхода, был рассмотрен алгоритм BM-25, а также метод, использующий

эмбеддинги. Во втором случае вектор документа формировался на основе предобученных моделей word2vec.

Рис. 4.3 и рис. 4.4 показывают, что тематический поиск дает результат лучше в терминах полноты поиска, и примерно такой же в терминах точности. Это подтверждает то, что построенная тематическая модель дает хорошее семантическое представление документов из коллекции и запросов.

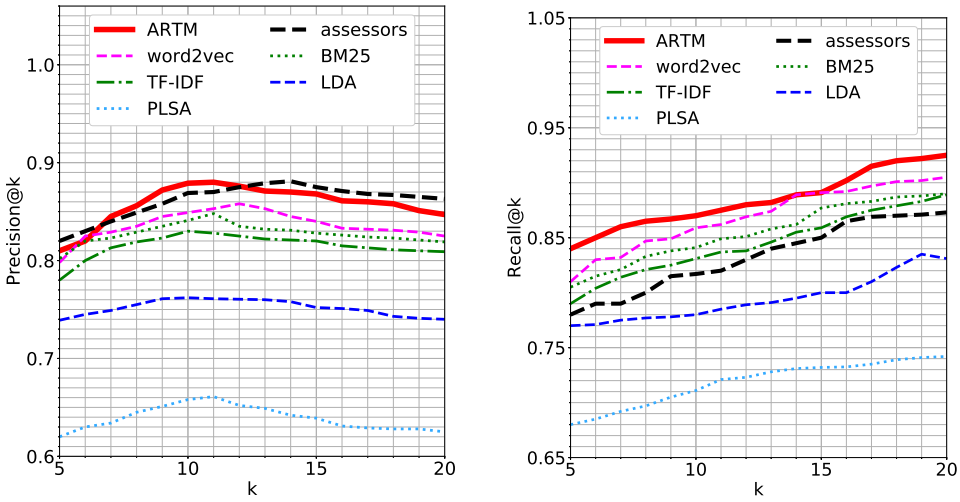

Рис. 4.3: Сравнение асессорского поиска, тематического поиска с регуляризацией (APTM) и бейзлайнов (TF-IDF, PLSA, LDA) для коллекции статей Хабрахабра

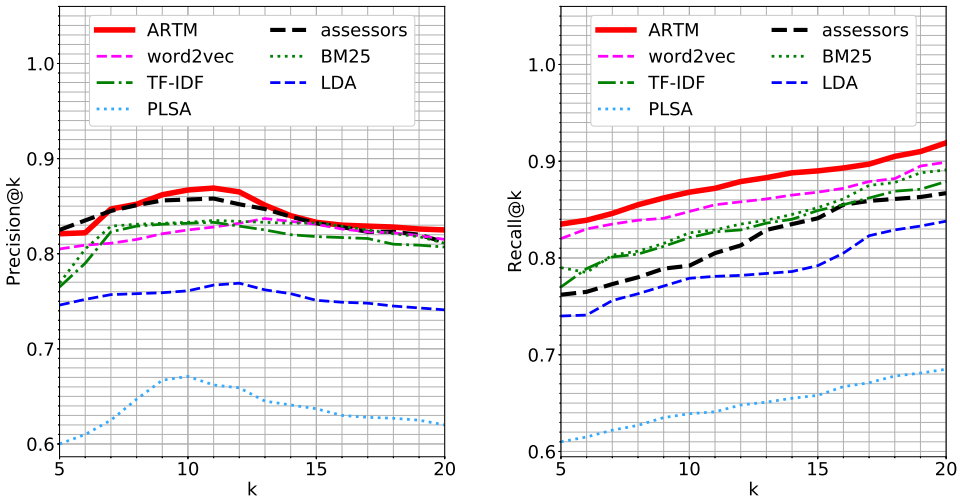

Рис. 4.4: Сравнение асессорского поиска, тематического поиска с регуляризацией (APTM) и бейзлайнов (TF-IDF, PLSA, LDA) для коллекции статей TechCrunch

Другое преимущество тематического поиска перед полнотекстовым (основанным на TF-IDF или BM-25 признаках, например) —это возможность значительно уменьшить размер инвертированного индекса для поиска. Из низкоразмерной разреженной матрицы тематических векторов получается "компактный"

инвертированный индекс, что приводит к более эффективной и быстро работающей реализации поискового механизма.

Кроме того, мы сравнились с двумя бейзлайнами, основанными на тематических моделях PLSA и LDA. Эксперименты показывают, что они обе дают более плохой результат, чем поиск, основанный на ARTM подходе (см. рис. 4.3 и рис. 4.4).

Критерий знаковых рангов Вилкоксона показывает, что разница в качестве поиска между нашим алгоритмом (тематический поиск) и бейзлайнами статистически значима. Всего было прведено 96 тестов: для метрик Precision@k и Recall@k ($k \in \{5, 10, 15, 20\}$), 6 бейзлайнов (асессоры, TF-IDF, BM-25, word2vec, LDA, PLSA), 2 текстовых коллекций (Хабрахабр и Techcrunch). P-values по всем тестам были меньше, чем 0.002 (4.1, 4.2).

Табл. 4.1: P-values для критерия знаковых рангов Вилкоксона по определению стат.значимости разницы в качестве поиска между тематическим поиском и бейзлайнами: асессоры, TF-IDF, BM25, word2vec, PLSA, LDA для коллекции статей Хабрахабра)

	assessors	PLSA	LDA	TF-IDF	BM-25	word2vec
Pr@5	0.00021	0.00011	0.00052	0.00112	0.00118	0.00115
Pr@10	0.00023	0.00034	0.00072	0.00205	0.00118	0.00092
Pr@15	0.00016	0.00012	0.00080	0.00114	0.00131	0.00092
Pr@20	0.00015	0.00037	0.00080	0.00119	0.00151	0.00115
R@5	0.00077	0.00003	0.00084	0.00092	0.00121	0.00109
R@10	0.00035	0.00004	0.00075	0.00078	0.00185	0.00125
R@15	0.00071	0.00007	0.00128	0.00299	0.00131	0.00142
R@20	0.00083	0.00005	0.00109	0.00107	0.00108	0.00153

Табл. 4.2: P-values для критерия знаковых рангов Вилкоксона по определению стат.значимости разницы в качестве поиска между тематическим поиском и бейзлайнами: асессоры, TF-IDF, BM25, word2vec, PLSA, LDA на коллекции статей TechCrunch)

	assessors	PLSA	LDA	TF-IDF	BM-25	word2vec
Pr@5	0.00015	0.00051	0.00071	0.00212	0.00111	0.00092
Pr@10	0.00014	0.00023	0.00075	0.00105	0.00125	0.00152
Pr@15	0.00025	0.00025	0.00052	0.00230	0.00145	0.00098
Pr@20	0.00019	0.00021	0.00091	0.00153	0.00180	0.00116
R@5	0.00053	0.00002	0.00052	0.00087	0.00164	0.00179
R@10	0.00088	0.00011	0.00093	0.00109	0.00192	0.00127
R@15	0.00044	0.00012	0.00175	0.00123	0.00151	0.00172
R@20	0.00057	0.00016	0.00098	0.00098	0.00094	0.00132

4.4 Построение тематической модели для разведочного поиска

Тематические модели строилась с помощью библиотеки BigARTM. Столбцы матрицы Φ инициализировались случайными распределениями, столбцы матрицы Θ — равномерными. В каждую тематическую модель были включены три регуляризатора: декоррелирование распределений терминов в темах (с коэффициентом τ), разреживание распределений тем в документах (с коэффициентом α), сглаживание распределений терминов в темах (с коэффициентом β).

Для оценивания качества работы модели и сравнения моделей друг с другом использовались следующие метрики (20): перплексия, разреженность (доля нулевых элементов в матрице распределений) распределений тем в документах, разреженность распределений токенов в темах для каждой из модальностей: термины, авторы, комментаторы, теги, категории. Указанные метрики являлись промежуточными мерами качества, тогда как основное внимание уделялось оптимизации точности и полноты поиска, построенного на основе тематической модели.

4.4.1 Подбор оптимальных параметров тематической модели

Наборы релевантных документов, найденных асессорами для каждого запроса позволяют оценивать качество работы новых тематических моделей и поисковых алгоритмов без проведения дополнительных оценок с участием асессоров. Ниже мы описываем три эксперимента, в которых оптимизировались следующие параметры тематического поисковика: мера близости, набор модальностей, количество тем. Подбирался по отдельности каждый из трех параметров, при фиксированных двух остальных. Подбор коэффициентов регуляризации подробно описан в следующей главе и здесь не затрагивается.

Таблица 4.3 показывает что косинусная мера оказалась наилучшей мерой близости между тематическими векторами документов и запросов. Тематическая модель, использованная в этом эксперименте, имела оптимальное количество тем и полный набор модальностей.

Табл. 4.3: Качество тематического поиска с использованием разных мер близости: Euclidean, Cosine, Manhattan, Hellinger, Kullback-Leibler

	Хабрахабр					**TechCrunch**				
	Eu	**cos**	Ma	He	KL	Eu	**cos**	Ma	He	KL
Pr@5	0.612	**0.810**	0.682	0.709	0.721	0.635	**0.819**	0.673	0.732	0.715
Pr@10	0.657	**0.879**	0.697	0.735	0.749	0.665	**0.867**	0.683	0.752	0.732
Pr@15	0.627	**0.868**	0.635	0.727	0.711	0.643	**0.833**	0.642	0.742	0.724
Pr@20	0.619	**0.847**	0.627	0.728	0.707	0.638	**0.825**	0.638	0.729	0.708
R@5	0.672	**0.840**	0.692	0.721	0.803	0.658	**0.835**	0.669	0.733	0.775
R@10	0.682	**0.870**	0.707	0.775	0.856	0.671	**0.868**	0.682	0.753	0.787
R@15	0.705	**0.891**	0.725	0.791	0.878	0.715	**0.890**	0.708	0.785	0.809
R@20	0.703	**0.925**	0.732	0.812	0.888	0.712	**0.919**	0.715	0.808	0.812

Таблица 4.4 показывает, что использование всех доступных модальностей повышает и точность, и полноту поиска. Биграммы и теги дают наибольший вклад, в то время как модели с использованием только одной модальности авторов дают наихудший результат. Все модели в этом эксперименте содержат оптимальное количество тем.

Табл. 4.4: Качество тематического поиска с различным набором модальностей
Хабрахабр: Assessors, Words, Bigrams, Comments, Tags, Hubs, Authors
TechCrunch: Assessors, Words, Bigrams, Authors, Categories

	Habrahabr						**TechCrunch**					
	As	W	C	WB	WBTH	**All**	As	W	C	WB	WBC	**All**
Pr@5	0.821	0.612	0.549	0.654	0.737	**0.810**	0.822	0.711	0.557	0.767	0.808	**0.819**
Pr@10	0.869	0.635	0.568	0.701	0.752	**0.879**	0.851	0.721	0.581	0.783	0.818	**0.867**
Pr@15	0.875	0.625	0.532	0.685	0.682	**0.868**	0.835	0.733	0.594	0.793	0.833	**0.833**
Pr@20	0.863	0.616	0.533	0.682	0.687	**0.847**	0.813	0.727	0.566	0.772	0.822	**0.825**
R@5	0.780	0.722	0.636	0.797	0.827	**0.840**	0.762	0.752	0.657	0.775	0.825	**0.835**
R@10	0.817	0.744	0.648	0.812	0.875	**0.870**	0.792	0.776	0.669	0.808	0.855	**0.868**
R@15	0.850	0.778	0.677	0.842	0.893	**0.891**	0.835	0.782	0.684	0.825	0.877	**0.890**
R@20	0.873	0.803	0.685	0.852	0.898	**0.925**	0.867	0.825	0.702	0.837	0.901	**0.919**

Таблица 4.5 показывает, что оптимальное количество тем в модели $|T|$ с полным набором модальностей равно 200 для Хабрахабра, 475 для TechCrunch.

Табл. 4.5: Качество тематического поиска при различных значениях $|T|$

	Habrahabr						**TechCrunch**					
	As	100	150	**200**	250	400	As	350	400	450	**475**	500
Pr@5	0.821	0.662	0.721	**0.810**	0.761	0.693	0.822	0.653	0.725	0.752	**0.819**	0.777
Pr@10	0.869	0.761	0.812	**0.879**	0.825	0.673	0.851	0.663	0.732	0.762	**0.867**	0.811
Pr@15	0.875	0.733	0.795	**0.868**	0.791	0.651	0.835	0.682	0.743	0.787	**0.833**	0.793
Pr@20	0.863	0.724	0.795	**0.847**	0.792	0.642	0.813	0.650	0.743	0.773	**0.825**	0.793
R@5	0.780	0.732	0.807	**0.840**	0.821	0.721	0.762	0.731	0.762	0.793	**0.835**	0.817
R@10	0.817	0.771	0.843	**0.870**	0.851	0.751	0.792	0.763	0.793	0.812	**0.868**	0.855
R@15	0.850	0.824	0.895	**0.891**	0.871	0.773	0.835	0.782	0.807	0.855	**0.890**	0.882
R@20	0.873	0.857	0.905	**0.925**	0.892	0.771	0.867	0.792	0.823	0.862	**0.919**	0.903

Весь набор экспериментов показывает, что оптимальное количество тем остается одинаковым при использовании различных мер близости, а оптимальный набор модальностей остается одинаковым при различных значениях $|T|$.

4.4.2 Подбор оптимальной стратегии регуляризации

Подбор большого количества параметров модели —процесс, требующий большого количества вычислительных ресурсов и времени. Мы частично упростили механизм подбора параметров с за счет оптимизации стратегий резуляризации модели. Все настройки модели (количество модальностей, словари терминов, количество тем) остались прежними, были доработаны только стратегии регуляризации. Вместо перебора значений коэффициентов регуляризации по сетке, была использована более продвинутая техника подбора коэффициентов. Остановимся более подробно на этой технике.

В подходе из предыдущей главы предлагалось добавлять регуляризаторы в модель одновременно, перед началом обучения модели. При этом предполагалось, что мы перебираем по сетке различные значения коэффициентов регуляризации и в итоге выбираем наилучшую модель (по критерию интерпретируемости тем и значениям промежуточных метрик качества: перплексии, разреженности матриц Φ и Θ). Учитывая, что модель содержала три регуляризатора, а для каждого из них нужно было перебрать как минимум 8-10 значений для достижения высокого уровня качества модели, процесс подбора коэффициентов регуляризации занимал очень много времени. Например, на коллекции статей с Хабрахабра модель обучалась в среднем 30 минут. Для ускорения процесса подбора параметров модели был разработан следующий метод.

Основное отличие от обычного перебора состоит в том, что теперь регуляризаторы добавляются в модель последовательно, один за одним. Промежуток времени после добавления i-го регуляризатора до момента добавления $(i+1)$-го регуляризатора будем называть глобальной итерацией. Каждая глобальная итерация содержит некоторое количество итераций EM-алгоритма при обучении модели (в нашем случае 8 итераций). На каждой глобальной итерации работаем только с тем регуляризатором, который был добавлен последним, коэффициенты для остальных регуляризаторов фиксируем. Из этих обученных моделей с разными значениями коэффициента регуляризации выбираем ту, которая позволяет улучшить одну (или несколько) метрик качества тематической модели без существенного понижения значений других метрик. На следующей глобальной итерации выбираем наиболее оптимальную модель с предыдущей итерации, вводим новый регуляризатор и подбираем оптимальное значение коэффициента регуляризации для него. Такой

подход дает значительный выигрыш по времени: вместо обучения $reg_1 \cdot reg_2 \cdot .. \cdot reg_n$ моделей (здесь reg_i — количество параметров для i-го регуляризатора, которые мы будем перебирать), нужно обучить всего лишь $reg_1 + reg_2 + .. + reg_n$. Кроме того, мы получаем возможность контролировать процесс обучения модели и корректировать его за счет введения новых регуляризаторов с различными весами прямо в процессе обучения.

Проиллюстрируем описанную концепцию на примере. При обучении тематических моделей для разведочного поиска было использовано три регуляризатора:

- Декоррелирование распределений терминов в темах
- Сглаживание распределений терминов в темах
- Разреживание распределений тем в документах

Они вводились в модель в том порядке, в котором перечислены в списке. Для каждого регуляризатора перебиралось по четыре значения коэффициентов регуляризации. На рис. 4.5 показаны зависимости перплексии и разреженности от числа итераций при различных значениях коэффициентов регуляризации. В результате была выбрана совокупность коэффициентов регуляризации $\tau = 10^8$, $\alpha = -1.5$, $\beta = 0.5$. Жирной кривой выделена наилучшая траектория регуляризации.

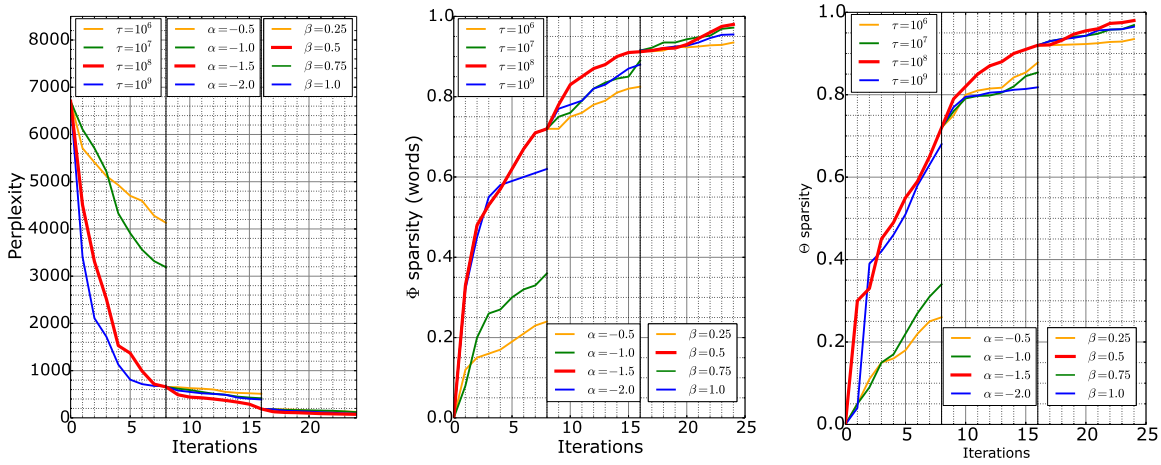

Рис. 4.5: Зависимости перплексии и разреженности матриц Θ и Φ (только по модальности терминов) от числа итераций и коэффициентов регуляризации.

Веса модальностей τ_m также подбирались по сетке методом проб и ошибок, по критериям перплексии, разреженности и качества тематического

поиска (см. ниже). В итоге были подобраны следующие значения τ_m для коллекции Хабрахабра: 1.0 для терминов, 5.0 для биграмм, 0.5 для авторов, 0.75 для комментаторов, 15.0 для тегов, 10.0 для хабов.

Для коллекции TechCrunch значения τ_m составили: 1.5 для терминов, 10.0 для биграмм, 0.7 для авторов, 25.0 для категорий.

В результате итоговое значение перплексии подобранной модели было ниже, чем у модели из предыдущей главы, коэффициенты которой перебирились по сетке. Разработанная техника подбора коэффициентов позволила сократить время на перебор параметров и поиск оптимальной по заявленным критериям модели.

4.4.3 Обоснование необходимости использования регуляризаторов

Вычислительная сложность процесса подбора коэффициентов регуляризации (даже при использовании техники из предыдущей главы) наводит на мысль о целесообразности использования регуляризации. В этой главе мы покажем, что стандартная тематическая модель без регуляризаторов (PLSA) не позволяет получить требуемого качества поиска, хотя и позволяет значительно сократить время обучения модели.

Цель этого эксперимента —показать, что каждый регуляризатор существенно повышает качества поиска. Таблица 4.6 показывает, что хотя регуляризатор декорреляции вносит наибольший вклад в качество поиска, два остальных регуляризатора тоже необходимы. Модель без регуляризации дает наихудший результат.

Табл. 4.6: Качество тематического поиска с разным набором регуляризаторов: Decorrelation, Θ-sparsing, Φ-smoothing

	Habrahabr				**TechCrunch**			
	no reg	D	DΘ	DΘΦ	no reg	D	DΘ	DΘΦ
Pr@5	0.628	0.748	0.771	**0.810**	0.652	0.775	0.779	**0.819**
Pr@10	0.653	0.776	0.812	**0.879**	0.679	0.787	0.819	**0.867**
Pr@15	0.642	0.765	0.792	**0.868**	0.669	0.773	0.798	**0.833**
Pr@20	0.643	0.759	0.783	**0.847**	0.673	0.777	0.792	**0.825**
R@5	0.692	0.784	0.805	**0.840**	0.673	0.812	0.812	**0.835**
R@10	0.714	0.814	0.834	**0.870**	0.685	0.821	0.845	**0.868**
R@15	0.725	0.835	0.867	**0.891**	0.712	0.859	0.869	**0.890**
R@20	0.735	0.862	0.891	**0.925**	0.723	0.882	0.895	**0.919**

4.4.4 Построение иерархической модели

Тематический поиск позволяет не только упростить и ускорить разведочный поиск, но и значительно повысить полноту поиска: автоматический поисковик зачастую находит документы, который асессор пропустил. При этом во всех предыдущих экспериментах точность тематического поиска оставалась сопоставимой с ручным поиском или даже оказывалась чуть ниже. Неоходимость более тщательно фильтровать поисковую выдачу и повышать precision поиска наводит на мысль о том, что используемые до текущего момента тематические модели недостаточно точно описывают тематическую составляющую документа, что приводит к появлению к выдаче "мусорных" или мало релевантных документов.

Для уточнения тематических векторов предлагается вместо классической тематической модели использовать иерархические модели. Иерархические модели обладают большей гибкостью и включают в себя как крупные темы —категории, так и более узкие предметные темы. При этом дочерняя тема может иметь более одного родителя (дочерняя тема «Социальные сети» может наследоваться от двух тем: «Общение» и «IT»).

Поиск на основе иерархической модели отличается от описанного в 2.4 алгоритма. Будем осуществлять поиск в несколько этапов: сначала сравниваем тематические вектора для тем верхнего уровня, затем отбираем релевантные темы верхнего уровня на основе проведенного сравнения и осуществляем следующий этап поиска только

внутри отобранных тем. Аналогично продолжаем сравнивать тематические вектора по уровням, углубляясь только в те темы, которые прошли отбор на предыдущем уровне. Такая каскадная система поиска позволяет отсеивать откровенно нерелевантные документы на первом проходе, когда размер инвертированного индекса очень маленький.

Рассмотрим описанный алгоритм на примере. На рис.4.6 представлен отрывок запроса о социальной сети LinkedIn. Плоская тематическая модель обнаружила в этом отрывке пять наиболее значимых тем: HR, Business, Social Nets, Mobile Dev., Web Dev. Очевидно, что первые три темы действительно релевантны, в то время как последние две попали сюда лишь потому, что в данном отрывке было сказано много слов о мобильном приложении и веб-версии от LinkedIn. Иерархическая модель выделила четыре верхнеуровневые темы: Management, Business, Online Communication, Software Development. Вероятность последней темы оказалась маленькой (про разработку программного обеспечения в отрывке действительно сказано мало), поэтому поиск внутри этой категории не осуществлялся. Внутри этой категории содержатся темы Mobile Dev. и Web Dev, ключевые слова которых действительно часто встречаются в тексте. Таким образом, каскадная система поиска позволила отсечь явно нерелевантные темы еще на первом проходе алгоритма.

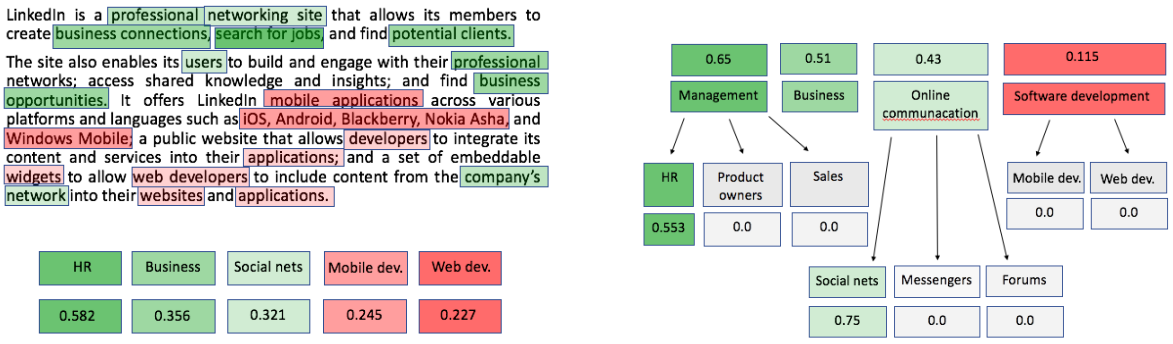

Рис. 4.6: Иллюстрация работы тематического поиска на основе плоской и иерархической модели

Проведем аналогичный 4.3 эксперимент и посчитаем точность и полноту поиска для иерархических моделей на коллекциях Хабрахабра и TechCrunch (рис.4.7, 4.8). Точность поиска при использовании иерархической модели существенно выросла. Полнота поиска тоже немного поднялась.

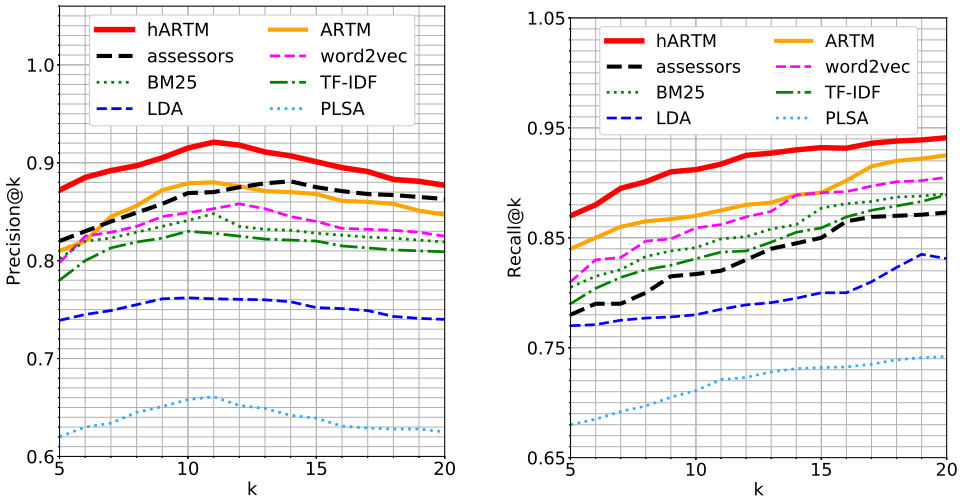

Рис. 4.7: Сравнение асессорского поиска, тематического поиска с регуляризацией (APTM и hARTM) и бейзлайнов (TF-IDF, PLSA, LDA) для коллекции статей Хабрахабра

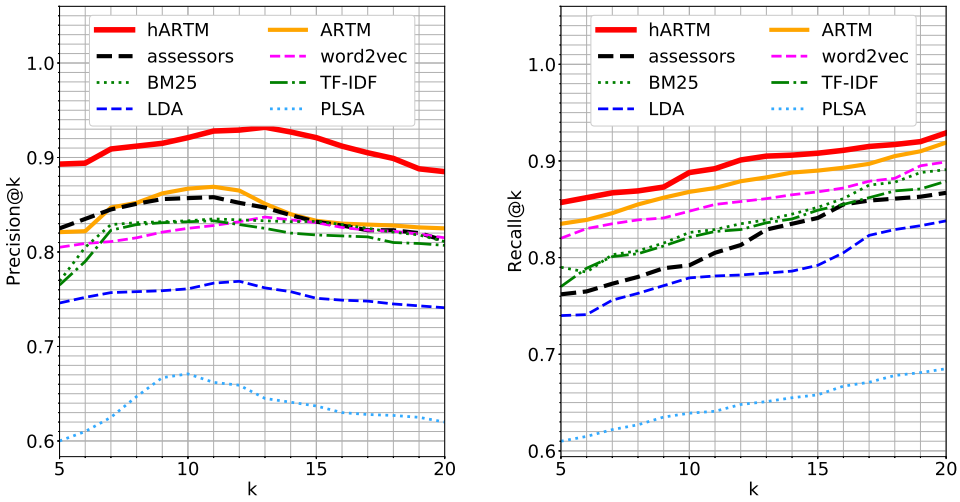

Рис. 4.8: Сравнение асессорского поиска, тематического поиска с регуляризацией (APTM и hARTM) и бейзлайнов (TF-IDF, PLSA, LDA) для коллекции статей TechCrunch

Критерий знаковых рангов Вилкоксона показывает, что разница в качестве поиска между hARTM и остальными моделями (плоские APTM модели, асессорский поиск и бейзлайны) статистически значима. Всего было прведено 80 тестов: для метрик Precision@k и Recall@k ($k \in \{5, 10, 15, 20\}$), 5 бейзлайнов (асессоры, TF-IDF, LDA, PLSA, ARTM), 2 текстовых коллекций (Хабрахабр и Techcrunch). P-values по всем тестам были меньше, чем 0.003 (4.7, 4.2).

Табл. 4.7: P-values для критерия знаковых рангов Вилкоксона по определению стат.значимости разницы в качестве поиска между тематическим иерархическим поиском и бейзлайнами: плоский APTM, асессоры, TF-IDF, BM25, word2vec, PLSA, LDA для коллекции статей Хабрахабр)

	ARTM	асессоры	PLSA	LDA	TF-IDF	BM25	word2vec
Pr@5	0.0014	0.0012	0.00051	0.00075	0.00202	0.00108	0.00151
Pr@10	0.0007	0.0053	0.00039	0.00092	0.00105	0.00106	0.00191
Pr@15	0.0008	0.0036	0.00042	0.00083	0.00214	0.00132	0.00110
Pr@20	0.0011	0.035	0.00047	0.00091	0.00219	0.00115	0.00121
R@5	0.0016	0.0077	0.00013	0.00054	0.00194	0.00161	0.00157
R@10	0.0018	0.0305	0.00024	0.00065	0.00188	0.00123	0.00182
R@15	0.0017	0.0071	0.00027	0.00123	0.00219	0.00165	0.00187
R@20	0.0019	0.0083	0.00025	0.0089	0.00217	0.00134	0.00139

Табл. 4.8: P-values для критерия знаковых рангов Вилкоксона по определению стат.значимости разницы в качестве поиска между тематическим иерархическим поиском и бейзлайнами: плоский APTM, асессоры, TF-IDF, BM25, word2vec, PLSA, LDA для коллекции статей TechCrunch)

	ARTM	асессоры	PLSA	LDA	TF-IDF	BM25	word2vec
Pr@5	0.0019	0.00161	0.00057	0.00061	0.00215	0.00115	0.00119
Pr@10	0.0009	0.0034	0.00025	0.00065	0.00108	0.00117	0.00165
Pr@15	0.0011	0.0025	0.00055	0.00082	0.00235	0.00191	0.00178
Pr@20	0.0015	0.0015	0.00081	0.00101	0.00113	0.00132	0.00127
R@5	0.0019	0.0053	0.00015	0.00042	0.00157	0.00164	0.00134
R@10	0.00015	0.0088	0.00031	0.00103	0.00115	0.00156	0.00182
R@15	0.0014	0.0404	0.00033	0.00174	0.00103	0.00191	0.00183
R@20	0.0015	0.0057	0.00018	0.00078	0.00215	0.00132	0.00154

4.5 Разведочный поиск на стилистически неоднородных текстовых коллекциях

Все вышеизложенные эксперименты проводились на однородных текстовых коллекциях: все статьи корпуса имели похожую тематическую направленность,

стилистику, схожий набор терминов и часто употребимых слов. Тематический поиск хорошо показал себя при работе с такими коллекциями. Для того, чтобы показать применимость описанной методики тематического поиска на боевых разнородных данных, проведем еще один эксперимент.

4.5.1 Описание данных для эксперимента

Смешаем две коллекции: использовавшиеся ранее новостные технические статьи с TechCrunch и аннотации научных статей с arxiv. Первый датасет представлен статьями о стратапах, IT, последних технических новинках и модных трендах: большая часть статей написана достаточно неформально, с использованием сокращений и иногда сленга. Научные статьи написаны формальным языком, стилистика текстов совершенно другая. Мы брали аннотации англоязычных статей за период с 2007 по 2017 год. В коллекции было 3 модальности: 9852 слова, 556783 биграммы и теги (keywords).

4.5.2 Иерархический каскадный поиск на стилистически неоднородных данных

На объединенных данных TechCrunch и arxiv мы обучили две тематические модели: иерархическую трехуровневую и плоскую одноуровневую. На основе каждой из моделей провели разведочный поиск по составленным ранее 100 запросам (см.4.1) и оценили качество поиска (4.9).

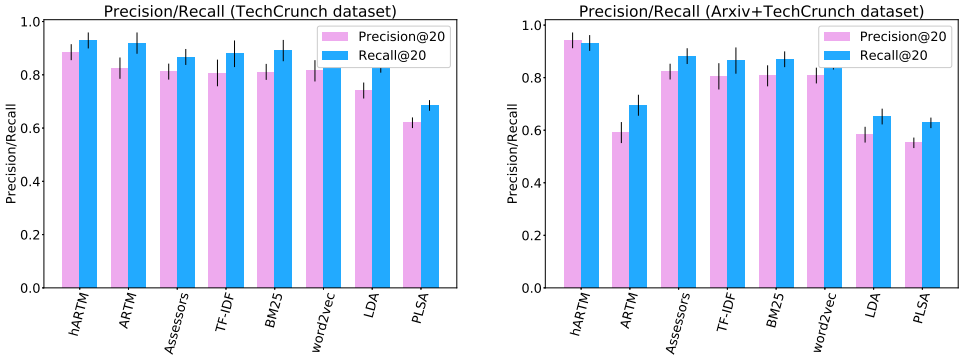

Рис. 4.9: Сравнение качества поиска на основе иерархической и плоской тематической моделей на однородной (TechCrunch) и неоднородной (TechCrunch+arxiv) коллекциях данных

На однородной текстовой коллекции и плоская и иерархическая модели работают

хорошо (точность у иерархической модели немного выше). При этом на неоднородной коллекции плоская тематическая модель дает очень плохой результат (Precision@20 = 0.591, Recall@20 = 0.695), в то время как иерархическая модель позволяет получить высокий уровень качества поиска (Precision@20 = 0.942, Recall@20 = 0.932).

Таким образом, данный эксперимент показывает хорошую обобщаемость предложенного метода каскадного тематического поиска на стилистически неоднородных текстовых коллекциях.

4.5.3 Теоретическое обоснование и реализация бейзлайнов

Кроме сравнения иерархического каскадного поиска с другими вариантами тематического поиска мы проверили несколько экспериментов по сравнению каскадного поиска с бейзлайнами. В качестве бейзлайна была выбрана нейросетевая модель: вместо тематической вектора мы брали выходы со скрытого слоя вариационного автоэнкодера. Опишем построение бейзлайна подробнее.

В качестве альтернативы тематическим векторам предлагается использовать вектор, в сжатом виде представляющий основную смысловую нагрузку документа. Для построения бейзлайна мы решали задачу style transfer для текстов (24). Основная сложность при решении этой задачи —отделить распределение, отвечающее за смысловую нагрузку текста, от распределения, отражающее стилистику и особенности языка статьи. Далее смысловые вектора запроса и документа используются для осуществления разведочного поиска, аналогично тематическим векторам в алгоритме тематического поиска.

Введем несколько обозначений:

1. y —скрытая переменная, отражающая стилистику текста, генерируется из распределения $p(y)$;

2. z —скрытая переменная, отражающая смысловую составляющую текста текста, генерируется из распределения $p(z)$;

3. x —документ, генерируется из условного распределения $p(x|y,z)$.

Сначала рассмотрим ситуацию, когда у нас есть два текста x_1 и x_2 с одинаковой смысловой составляющей, но разными стилями y_1 и y_2. Тогда пусть есть два датасета $X_1 = (x_1^{(1)}, .., x_1^{(n))}$ и $X_2 = (x_2^{(1)}, .., x_2^{(m))}$, насемплированные из распределений $p(x_1|y_1)$

и $p(x_2|y_2)$ соответственно. В этом случае мы хотим определить функции "переноса стиля" $p(x_1|x_2, y_1, y_2)$ и $p(x_2|x_1, y_1, y_2)$. В этом случае нам нужно найти совместное распределение:

$$p(x_1, x_2|y_1, y_2) = \int_z p(z)p(x_1|y_1, z)p(x_2|y_2, z)dz$$

Поскольку x_1 и x_2 условно независимы, получаем:

$$p(x_1|x_2, y_1, y_2) = \int_z p(x_1, z|x_2, y_1, y_2)dz$$

$$= \int_z p(z|x_2, y_2) \cdot p(x_1|y_1, z)dz = \mathbb{E}_{z\sim p(z|x_2,y_2)}[p(x_1|y_1, z)]$$

Из последней формулы вытекает целесообразность применения автоэнкодера. Вообще задача переноса стиля подразумевает два этапа: первый по выделению смыслового вектора $z \sim p(z|x_2, y_2)$, второй —генерация текста с необходимой стилевой составляющей из распределения $p(x_1|y_1, z)$. Нас будет интересовать только первый этап.

Пусть у нас есть энкодер $E : X \times Y \to Z$ и генератор $G : Y \times Z \to X$. E и G образуют автоэнкодер, когда применяются к одному и тому же стилю. Ошибку реконструкции этого автоэнкодера можно записать следующим образом:

$$L_{rec}(\theta_E, \theta_G) = \mathbb{E}_{x_1\sim X_1}[-\log p_G(x_1|y_1, E(x_1, y_1))] + \mathbb{E}_{x_2\sim X_2}[-\log p_G(x_2|y_2, E(x_2, y_2))]$$

Необходимость ввести граничение на x_1 и x_2 (они должны быть сгенерированы из одного распределения $p(z)$, отражающего смысловую направленность текста), наводит на мысль о применении вариационного автоэнкодера. В нем используется следующий регуляризатор:

$$L_{KL}(\theta_E) = \mathbb{E}_{x_1\sim X_1}[D_{KL}(p_E(z|x_1, y_1)||p(z))] + \mathbb{E}_{x_2\sim X_2}[D_{KL}(p_E(z|x_2, y_2)||p(z))]$$

В итоге, для решения задачи нужно максимизировать сумму L_{KL} и L_{rec}. В качестве бейзлайнов предлагается взять VAE, скрытый слой которого позволит нам получить вектор, с закодированной информацией о смысловой составляющей текста. Кроме того, предлагается взять два более сильных бейзлайна —варианты автоэнкодера с некоторыми ограничениями (aligned и cross-aligned автоэнкодеры), о которых подробно рассказано в (24).

4.5.4 Сравнение качества работы тематического поиска с бейзлайнами на стилистически неоднородных данных

Построенные бейзлайны на основе вариационного автоэнкодера оказались достаточно сильными и по качеству практически сравнялись с ручным асессорским поиском по всем трем коллекциям. На однородных коллекциях тематический поиск (даже основанный на одноуровневой тематической модели) уверенно обходит бейзлайны 4.9. На стилистически неоднородной коллекции плоский APTM немного проигрывает моделям, на основе вариационного автоэнкодера, в то время как иерархический APTM все-таки немного выигрывает у бейзлайнов в терминах точности и полноты поиска. Стататистическая значимость результатов была проверена с помощью критерия знаковых рангов Вилкоксона. Для всех тестов p-value был меньше, чем 0.005.

Табл. 4.9: Сравнение качества работы тематического поиска и бейзлайнов: асессоры, TF-IDF, PLSA, LDA, Variational auto-encoder (VAE), Aligned auto-encoder (aligned AE), Cross-aligned auto-encoder (cross-aligned AE) на примере трех коллекций: Хабрахабр, TechCrunch, arxiv+TechCrunch)

	Habrahabr		**TechCrunch**		**arxiv+TechCrunch**	
	Precision	Recall	Precision	Recall	Precision	Recall
Асессоры	0.863	0.873	0.812	0.867	0.815	0.850
TF-IDF	0.809	0.889	0.807	0.879	0.756	0.763
BM-25	0.811	0.892	0.809	0.881	0.761	0.769
word2vec	0.813	0.893	0.814	0.883	0.771	0.777
PLSA	0.625	0.742	0.620	0.685	0.558	0.572
LDA	0.740	0.831	0.741	0.838	0.591	0.618
VAE	0.771	0.752	0.787	0.753	0.802	0.781
Aligned AE	0.812	0.785	0.827	0.794	0.831	0.791
Cross-aligned AE	0.826	0.805	0.853	0.837	0.847	0.813
ARTM	0.847	0.925	0.825	0.919	0.591	0.695
hARTM	**0.877**	**0.941**	**0.885**	**0.929**	**0.942**	**0.932**

Бейзлайны на основе вариационного автоэнкодера оказались сильными, что подсказывает достаточно очевидную идею осуществлять поиск одновременно по тематическим и смысловым векторам. В среднем такой подход позволяет повысить

качество разведочного поиска на 2% в терминах полноты и 3.1% в терминах точности. При использовании иерархического APTM и cross-aligned автоэнкодера удалось достичь точности поиска в 93.5%, а полноты в 95.1%. Такие высокие значения качества поиска в рамках проведенных экспериментов достигались только за счет объединения тематических векторов и представлений, полученных с помощью вариационного автоэнкодера.

Табл. 4.10: Сравнение качества работы тематического поиска и комбинированных моделей: (h)ARTM + VAE, aligned auto-encoder, cross-aligned auto-encoder на примере трех коллекций: Хабрахабр, TechCrunch, arxiv+TechCrunch)

	Habrahabr		**TechCrunch**		**arxiv+TechCrunch**	
	Precision	Recall	Precision	Recall	Precision	Recall
VAE	0.77	0.75	0.78	0.75	0.80	0.78
Aligned AE	0.81	0.78	0.82	0.79	0.83	0.79
Cross-aligned AE	0.82	0.80	0.85	0.83	0.84	0.81
ARTM	0.847	0.925	0.825	0.919	0.591	0.695
hARTM	0.877	0.941	0.885	0.929	0.942	0.932
ARTM + VAE	0.848	0.929	0.829	0.921	0.850	0.830
ARTM + aligned AE	0.850	0.935	0.835	0.928	0.852	0.837
ARTM + cross-aligned AE	**0.855**	**0.939**	**0.849**	**0.932**	**0.862**	**0.845**
hARTM + VAE	0.921	0.941	0.927	0.939	0.926	0.929
hARTM + aligned AE	0.929	0.945	0.930	0.948	0.932	0.940
hARTM + cross-aligned AE	**0.931**	**0.948**	**0.932**	**0.950**	**0.935**	**0.951**

Часть 5

Заключение

5.1 Итоги работы

Разведочный информационный поиск направлен на решение задач интенсификации процесса обучения, систематизации новой информации, исследования и суммаризации больших объемов знаний. Тематическое моделирование —одна из ключевых технологий разведочного поиска. В данной работе исследуется тематический поиск по длинным текстовым запросам на примере коллекций статей коллективного блога Хабрахабр и новостных статей TechCrunch.

В данной работе разработан новый метод решения задач разведочного поиска, основанный на тематическом моделировании. В рамках проведенного исследования была достигнута поставленная цель и решены сформулированные в начале исследования задачи. Подведем итоги по проделанной работе:

1. Разработан метод решения задач разведочного поиска: тематический поисковик, позволяющий по текстовому описанию поискового намерения пользователя находить статьи необходимой тематики.

2. Предложена методика оценивания качества разведочного поиска с использованием асессорских разметок.

3. Сгенерированы запросы для разведочного поиска и проведен эксперимент по оцениванию качества тематического поиска с участием более, чем 200 асессоров.

4. В результате эксперимента было выявлено преимущество созданного тематического поисковика перед полнотекстовым поиском при решении задач разведочного поиска. Выявлено, что автоматический тематический

поиск позволяет получить recall превышающий 90%, что в среднем на 5% больше, чем у ручного асессорского поиска. Кроме того, тематический поисковик значительно выигрывает по времени у ручного поиска: в среднем, асессор тратит 30 минут на обработку одного запроса, в то время как поисковик работает $1 - 2$ секунды.

5. Разработана каскадная уточняющая система поиска на основе иерархических тематических моделей. Использование данного подхода позволило увеличить точность поиска более, чем на 7%.

6. Доказана эффективность работы каскадной системы поиска на стилистически разнородных больших текстовых коллекциях. Таким образом, можно говорить о хорошей обобщаемости предложенного метода на различных данных.

Каскадный подход к поиску позволяет эмулировать итеративный процесс поиска, когда пользователю приходится несколько раз переформулировать свой запрос, постепенно уточняя свой вопрос. Иерархический тематический поиск позволяет избавиться от необходимости переформулировки запроса, что приводит к экономии времени ищущего. Вложенная тематическая структура является некоторой аппроксимацией возможных разветвлений дорожной карты предметной области, по которой осуществляется поиск. Таким образом, разработанная технология позволяет решать задачи разведочного с высоким уровнем качества в автоматическом режиме практически без участия человека.

5.2 Дальнейшие исследования

Рассмотрим возможные направления дальнейшего исследования в рамках данной работы:

1. Разработать и внедрить алгоритм ранжирования статей в порядке, удобном для чтения. Формировать поисковую выдачу в соответствии с ним.

2. Добавить возможность задавать степень "широты" и "глубины" поиска: пользователь должен уметь искать либо максимально много документов на заданную и смежную темы, либо находить только документы четко регламентированные поисковым запросом.

3. Расширить базу бейзлайнов для сравнения: добавить подходы на основе эмбеддингов (word2vec, fasttext) и нейросетевые подходы.

4. Разработать веб-интерфейс разведочного поиска, провести UX/UI тестирование для определения максимально удобного пользователю интерфейса. После тестирования продумать стратегию внедрения системы и продолжать работу над тематическим поисковиком как над самостоятельным коммерческим продуктом.

Список источников и литературы

[1] *G.Marchionini.* Exploratory search: from finding to understanding / G.Marchionini // *Communications iof the ACM.* — 2006. — April. — Vol. 49, no. 4. — Pp. 41–46.

[2] *White, Ryen W.* Exploratory Search: Beyond the Query-Response Paradigm / Ryen W. White, Resa A. Roth. Synthesis Lectures on Information Concepts, Retrieval, and Services. — Morgan and Claypool Publishers, 2009.

[3] *Wang, Wei.* Probabilistic Topic Models for Learning Terminological Ontologies / Wei Wang, Payam Mamaani Barnaghi, Andrzej Bargiela // *IEEE Trans. on Knowl. and Data Eng.* — 2010. — Vol. 22, no. 7. — Pp. 1028–1040.

[4] *K.Vorontsov, A.Potapenko.* Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization / A.Potapenko K.Vorontsov // *Communications in Computer and Information Science (CCIS).* — 2014. — Vol. 436. — Pp. 29–46. — AIST2014, Analysis of Images, Social networks and Texts. Springer International Publishing Switzerland. `http://www.machinelearning.ru/wiki/images/1/1f/Voron14aist.pdf`.

[5] *T.Hoffman.* Probabilistic Latent Semantic Analysis / T.Hoffman // *Uncertainity in Artificial Intelligence.* — 1999. `http://cs.brown.edu/~th/papers/Hofmann-UAI99.pdf`.

[6] *Blei, David M.* Latent Dirichlet allocation / David M. Blei, Andrew Y. Ng, Michael I. Jordan // *Journal of Machine Learning Research.* — 2003. — Vol. 3. — Pp. 993–1022.

[7] *Blei, David M.* Probabilistic topic models / David M. Blei // *Communications of the ACM.* — 2012. — Vol. 55, no. 4. — Pp. 77–84.

[8] *K.Vorontsov O.Frei, M.Apishev A.Yanina P.Romov M.Dudarenko.* Non-Bayesian Additive Regularization for Multimodal Topic Modeling of Large Collections / M.Apishev A.Yanina P.Romov M.Dudarenko K.Vorontsov, O.Frei. — 2014.

[9] Directing exploratory search: Reinforcement learning from user interactions with keywords / Dorota Glowacka, Tuukka Ruotsalo, Ksenia Konuyshkova et al. // Proceedings of the 2013 international conference on Intelligent user interfaces / ACM. — 2013. — Pp. 117–128.

[10] Social bookmarking and exploratory search / David R Millen, Meng Yang, Steven Whittaker, Jonathan Feinberg // ECSCW 2007. — Springer, 2007. — Pp. 21–40.

[11] *S.Goldenberg.* Exploratory Search in WorkTop / S.Goldenberg // *Brown University.* — 2012. — February. — P. 19.

[12] DEESSE: entity-driven exploratory and serendipitous search system / Olivier Van Laere, Ilaria Bordino, Yelena Mejova, Mounia Lalmas // Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management / ACM. — 2014. — Pp. 2072–2074.

[13] *G. Koutrika L. Liu, S. Simske.* Generating Reading Orders over Document Collections / S. Simske G. Koutrika, L. Liu // *HP Labs, Palo Alto.* — 2015. — March.

[14] *Gontek, Mirko.* User modeling for exploratory search on the Social Web / Mirko Gontek // *Dissertation zur Erlangung des Doktorgrades der Philosophischen Fakultet der Universitet zu Keln im Fach Informationsverarbeitung.* — 2011. — June. — P. 266.

[15] *Янина, Анастасия Олеговна.* Мультимодальные тематические модели для разведочного поиска в коллективном блоге / Анастасия Олеговна Янина, Константин Вячеславович Воронцов // *Машинное обучение и анализ данных.* — 2016. — Vol. 2, no. 2. — Pp. 173–186.

[16] *Ianina, Anastasia.* Multi-objective topic modeling for exploratory search in tech news / Anastasia Ianina, Lev Golitsyn, Konstantin Vorontsov // Conference on Artificial Intelligence and Natural Language / Springer. — 2017. — Pp. 181–193.

[17] *J.Chang J.Boyd-Graber, S.Gerrish Ch.Wang D.M.Blei.* Reading Tea Leaves: How Humans Interpret Topic Models / S.Gerrish Ch.Wang D.M.Blei J.Chang, J.Boyd-Graber // *Advances in neural information processing systems.* — 2009. — Pp. 288–296. http://www.umiacs.umd.edu/~jbg/docs/nips2009-rtl.pdf.

[18] *H.Wu Y.Wang, X.Cheng.* Incremental probabilistic latent semantic analysis for automatic question recommendation / X.Cheng H.Wu, Y.Wang // *Proceedings of the 2008 ACM conference on Recommender systems.* — 2008. — Pp. 99–106.

[19] *D.M. Blei A.Y. Ng, M.I. Jordan.* Latent Dirichlet Allocation / M.I. Jordan D.M. Blei, A.Y. Ng // *Journal of Machine Learning Research.* — 2003. — January. — Pp. 993–1022. http://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf.

[20] *Vorontsov, K. V.* Additive Regularization of Topic Models / K. V. Vorontsov, A. A. Potapenko // *Machine Learning, Special Issue on Data Analysis and Intelligent Optimization with Applications.* — 2015. — Vol. 101, no. 1. — Pp. 303–323.

[21] *Vorontsov, K. V.* Additive Regularization of Topic Models for Topic Selection and Sparse Factorization / K. V. Vorontsov, A. A. Potapenko, A. V. Plavin // The Third International Symposium On Learning And Data Sciences (SLDS 2015). April 20-22, 2015. Royal Holloway, University of London, UK. / Ed. by A. Gammerman et al. — Springer International Publishing Switzerland 2015, 2015. — Pp. 193–202.

[22] Additive Regularization for Topic Modeling in Sociological Studies of User-Generated Text Content / Murat Apishev, Sergei Koltcov, Olessia Koltsova et al. // MICAI 2016, 15th Mexican International Conference on Artificial Intelligence. — Vol. 10061. — Proceedings, Part I. Lecture Notes in Artificial Intelligence, 2016. — P. 166–181.

[23] *N. A.Chirkova, K.V.Vorontsov.* Additive regularization for hierarchical multimodal topic modeling / K.V.Vorontsov N. A.Chirkova // *Journal Machine Learning and Data Analysis.* — 2016. — Pp. 187–200. http://www.machinelearning.ru/wiki/images/d/d7/Voron14dan-rus.pdf.

[24] *T.Shen T.Lei, R.Barzilay T.Jaakkola.* Style transfer from non-parallel text by cross-alignment / R.Barzilay T.Jaakkola T.Shen, T.Lei // *In Advances in Neural Information Processing Systems.* — 2017. — Pp. 6833–6844. http://www.machinelearning.ru/wiki/images/d/d7/Voron14dan-rus.pdf.

Список рисунков

Список таблиц