Министерство науки и высшего образования
Российской Федерации
«Московский физико-технический институт
(государственный университет)»
Физтех-школа прикладной математики и информатики
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Дементьева Дарина Степановна

Агрегация и персонализация новостного текстового контента

03.04.01 — Прикладные математика и физика

Выпускная квалификационная работа
(магистерская диссертация)

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

Москва

2019 г.

Содержание

Аннотация

Данная работа посвящена разведочному поиску. На текущий момент существует достаточно много поисковых систем, однако они удовлетворяют далеко не все потребности пользователей. В частности, пользоватеям не всегда комфортно использовать известные большие поисковики в образовательных целях – например, для изучения с нуля конкретной тематики, если пользователь еще ничего не знает о теме. Такими задачами занимается разведочный поиск.В литературе не сложилось устоявшихся представлений о пользовательском интерфейсе, представлении поисковой выдачи и измерения качества разведочного поиска. В этой работе предлагается новый процесс разведочного поиска, система для его реализации, а также методика оценивания системы разведочного поиска.

1 Введение

В наше время каждый человек ежедневно пользуется какой-либо поисковой системой – например, наиболее популярными Яндекс или Google. Эти поисковые системы развиваются уже на протяжении нескольких десятков лет, и за это время они делают все возвожное, чтобы выдавать пользователям наиболее точный ответ на его запрос и за наиболее короткое время. И хотя, казалось бы, возможности современных поисковиков безграничны, все равно есть некоторые кейсы использования поиска, которые остаются непокрытыми.

1.1 Описание проблематики

Достаточно много людей используют сейчас поисковые системы для образования или саморазвития. Многи профессионалы используют поиск для того, чтобы закрыть пробелы в некоторых темах или, наоборот, с нуля изучить какую-нибудь тему для повышения своей квалификации. Однако, поиск нужной информации в таком случае при использовании наиболее популярных поисковых систем может занять большое количество времени и не всегда закончиться результативно.

Далее приведен список проблем, с которыми сейчас наиболее часто сталкиваются при поиске информации: см. табл. 1.

1.2 Разведочный поиск

Одним из решением всех вышеперечисленных проблем может быть применение разведочного информационного поиска. Такой поиск позволяет даже при неточно сформулированном запросе найти необходимую информацию по всей базе знаний. Более того, пользователь в качестве запроса может написать не короткое предложежние или словосочетание, а целый абзац или документ. И, в таком случае, система должна определить тематику поданного ей запроса и, соответственно выдать, статьи по заданным тематикам.

Это позволило бы лишить пользователей многих неудобств. Так как такой поиск всегда учитывает тему запроса, то ненужный шум в ответе будет выдаваться горадо реже. Пользователь может подавать в качестве запроса не одну статью, а целый набор статей, и получать структурированный по всем присутствующим тематикам в запросе ответ. В итоге, пользователь может накидывать в запрос абсолютно разную

№	Проблема	Комментарий
1.	Много полезной информации не доступно, т.к. она платная	К сожалению, это относится ко многим интересным ресурсам, но в общем случае эту проблему решить нельзя.
2.	Среди шума тяжело найти действительно полезную информацию	Нынешние рекомендательные системы или поисковые системы стараются выдавать ответы так, чтобы было больше кликов от пользователей, но совершенно не обращают внимание на действительную релевантность ответа тематике запроса.
3.	При поиске не всегда понятен уровень статьи - для новичков или для глубоко разбирающихся	Сейчас очень много ресурсов онлайн образования, блогов, где авторы стараются популярно объяснить материал, но многие такие учебные материалы, которые быстрее всего ищутся, либо слишком простые, либо слишком сложные. И становится это понятно, когда пользователь потратит несколько минут на беглое ознакомление со статьей.
4.	Не всегда по теме есть подборки. Нет единого ресурса, который действительно агрегировал бы новости по интересующим тематикам	Вся необходимая информация для изучения какой-то одной темы может быть разбросана по разным источникам. В результате этого, пользователю нужно потратить много времени и проявить сообразительность, чтобы все собрать воедино.
5.	На основных ресурсах информация не структурирована	Да, на многих сервисах есть теги, группы, человек может на них "подписаться". Но пользователь может иметь несколько таких подписок. В итоге, он все равно получает ленту с новыми статьями, где далеко не все ему необходимо.

Таблица 1: Список проблем, с которыми сталкиваются пользователи при поиске информации.

информацию, совершенно не заботясь о ее чистоте. А в результате работы системы будет иметь хорошо структурированные знания, которые значительно сэкономят ему время для погружения в новую тему.

1.3 Существующие сервисы

Сейчас существует достаточно много сервисов, которые реализуют разведочный поиск в том или ином виде. Однако, многие из них либо находятся еще в стадии разработки, либо узкоспециализированы.

Так, одним из недавно возникших решений в сфере разведочного поиска является сервис Cortical.io [21]. Данный ресурс предоставляет разного рода решеня – извлечение ключевых слов, определение языка текста, сравнение двух текстов о схожести тематик. Главным преимуществом этой компании является извлечение смыслов и тематик из текста. Их подход заключается в проведении параллели между векторными представлениями документов и корковым механизмом запоминания информации. Текст можно представить в виде разреженной матрицы присутсвия слов. Затем, это представление можно преобразовать в "отпечатоквумерную матрицу – который будет передавать смысл текста. Далее, с такими матрицами можно проводить абсолютно разные операции – сравнивать тексты, вычислять схожесть, проводить поиск.

Довольно популярным среди ученых является сервис Mendeley [15]. Это программное обеспечение, которое позволяет проводить поиск по статьям, читать статьи в удобном формате, а также по сформированной папке документов смотреть новые релевантные статьи. Главным ограничением Mendeley является то, что поиск и добавление статей может происходить только по базе научного издательского дома Elseveir.

Еще одной новой работой является проект Arxiv Sanity Preserver [12]. Он позволяет сохранять в личный кабинет пользователя статьи, а также выдает рекомендации для прочтения на основе подборки пользователя. В качестве базы статей используется ресурс arxiv.org. В базе сохраняются научные статьи по многим категориям. Поэтому, в базовой реализации Arxiv Sanity включены лишь разделы, связанные с Computer Science и Machine Learning. А в качестве рекомендательной системы стоит реализация на основе tf-idf.

1.4 Содержание работы

На текущий момент в литераутре не сложилось устоявшихся представлений о пользовательском интерфейсе системы разведочного поиска, представлении поисковой выдачи и измерения качества разведочного поиска. Целью данной работы яв-

ляется разработать новый процесс разведочного поиска и методику оценки системы разведочного поиска. В рамках поставленной цели необходимо решить следующие-задачи:

1. Разработать систему разведочного поиска;
2. Разработать модели векторного представления текстовых данных для такого вида поиска;
3. Оценить качество работы системы разведочного описка.

Далее в работе в разделе 2 будет более подробно описано понятие разведочного поиска, современное состояние проблематики касательно описания систем разведочного поиска, разные подходы реализации такого типа поиска и способы их оценки. В части 3 будет представлен предлагаемый в данной работе процесс разведочного поиска, методика оценивания качества и описание прототипа системы развелочного поиска. Далее в раделе 4 будут описаны модели, которые используются для ранжирования документов. Будут подробно описаны разные способы векторного представления документов и как потом они используются при построении рекомендательной ленты. И в разделе 5 будет описан эксперимент по оцениванию качества работы предложенной системы и будут приведены полученные результаты, основанные на ассесорских оценках.

2 Обзор современного состояния проблемы

В данном разделе будет рассмотрена концепция разведочного информационного поиска и его онсовные понятия. Также здесь приведен обзор основных исследований в этом направленит. Будет описано, каким главным набором функций должна обладать система разведочного поиска и их мотивацию из основных потребностей пользователя. В конце будут рассмотрны разные методики оценивания качества работы систем разведочного поиска и оценено, насколько они отображают реальную удовлетворенность пользователя.

2.1 Понятие разведочного поиска

Люди используют поиск для разных целей, а, значит, по-разному ведут себя с системой для достижения этих целей. Первая классификация разных типов поведения пользователей в поисковых системах была представлена в работе [6]. Там же и упоминается вид поиска, когда пользователь может блуждать по базе знаний в поиске нужной информации, но совсем не имея конкретной цели. Также предполагалось, что специально под такой тип поиска необходимо создавать свою информационно-поисковую систему со своим проработанным интерфейсом.

После значительного развития поисковых систем и внедрения поиска в повседневную жизнь, понятие разведочного поиска вновь поднимается в работе [14]. Действительно, не всегда использование коротких запросов, под которые заточены большинство поисковиков, может удовлетворить все потребности пользователей. Поскольку каждый поисковик предоставляет функции просмотра, навигации, а также выбора информации путем проб и ошибок, то было бы естественно использовать такую систему в образовательных целях. Именно в работе [14] приводится классификация типов поиска на два больших класса – просмотровый поиск (Look-Up Search) и разведочный поиск (Exploratory Search). Нельзя сказать, что эти два класса абсолютно не пересекаются. Раздведочный поиск может стать продолжением просмотрового поиска. Зависимость между этими двумя классами и более точная подклассификация приведена в табл. 2

Однако, не смотря на то, что уже было несколько попыток дать определение разведочному поиску, в работе [18] утверждается, что до сих пор нет точного определения, что же такое разведочный поиск. Из этого делается предположение, что этот тип поиска проще описать через его основные характеристики.

Просмотровый поиск	Разведочный поиск	
	Изучение	Исследование
Установление фактов	Приобретение знаний	Планирование
Ответ на вопрос	Сравнение	Аккумулирование
Известный предмет	Постижение	Анализ
Транзакционная проверка	Агрегирование	Исключение
Изучение	Социализирование	Оценка
		Открытие
		Синтез
		Преобразование

Таблица 2: Список поисковых задач, разбитых по категориям просмотрового поиска и разведочного поиска согласно [14]

С другой стороны, в работе [11] снова пытаются провести границу между просмотровым поиском и разведочным поисков. Для этого при определении типа поиска авторы опираются на две основные характеристики:

- **Цель:** В развелочном поиске цель является неточной и открытой. То есть, не существует единого ответа, который удовлетворяет информационные потребности пользователя, и нет четкого критерия, когда нужно завершить поиск. Следовательно, оценка актуальности результатов не является дискретной. В задачах просмотрового поиска существует точная цель поиска. Цель поиска достигается путем извлечения конечного набора релевантных результатов, а релевантность результатов можно оценить дискретно.

- **Сложность:** Объективная сложность задачи поиска обычно определяется количеством путей, участвующих в процессе поиска. Очевидно, что для разведочного поиска мы не можем определить единый и прямой путь, который приведет к желаемым результатам. Поэтому исследовательские задачи имеют высокую сложность. В задачах просмотрового поиска процесс поиска является более простым и включает всего несколько шагов – задачи такого поиска обычно имеют гораздо меньшую сложность, чем задачи разведочного поиска.

В итоге, основываясь на этих характеристиках, в [11] появляется новая классификация типов поиска: см. табл. 3

	Низкая сложность	Высокая сложность
Точная цель	Основной просмотровый поиск	Пограничный просмотровый поиск
Открытая цель	Пограничный разведочный поиск	Основной разведочный поиск

Таблица 3: Категоризация просмотрового и разведочного поисков.

Разведочный поиск также связан в литературе с другим типом поведения, который можно назвать "добыча"информации. Теория поиска информации пытается понять и объяснить, как люди ищут информацию. Авторы связывают пищевое поведение с поведением, связанным с поиском информации, в том смысле, что модели поведения схожи. Например, на основе концепции «информационного запаха» ищущие информацию обнаруживают и используют сигналы для перехода от одного информационного патча к другому, ища соответствующую информацию для своей цели. Мы можем найти такое поведение также в разведочном поиске.

Делая выводы на основе вышесказанного, можно прийти к заключению, что действительно разграничить четко разведочный поиск от других типов поиска нельзя. Для того, чтобы окончательно для дальнейшего обсуждения разделить просмотровый и разведочный поиски, можно обратиться к примерам:

1. **Просмотровый поиск:**

   - *Установление фактов:* найти конкретный ответ на простой вопрос.
   - *Навигационные задачи:* поиск определенного веб-сайта или документа. В навигационном поиске ищущий информацию может просто «думать», что конкретный веб-сайт / документ существует, и искать его.
   - *Ответ на вопрос:* поиск правильного набора ответов, где существует четкий список соответствующих ответов.

2. **Разведочный поиск:**

   - *Приобретение знаний:* открытые цели поиска, потому что у задач обучения нет четких критериев, когда заканчивать поиск. Искатель информации может продолжать выполнение такой задачи до тех пор, пока не будет достигнут субъективный уровень удовлетворенности.

- *Планирование задач:* сбор обзоров новой области в рамках подготовки к будущей деятельности.
- *Сравнение:* вовлекать сбор информации по двум или более темам для анализа сходств и различий между ними.

Однако, все равно во время своей поисковой активности пользователь легко может переключаться между разными типами поискового поведения. Четко можно лишь сказать, что *главной целью разведочного поиска является* ***обучение***.

2.2 Основные характеристики системы разведочного поиска

В [18] дано подробное описание всех аспектов разведочного поиска. Мы приведем наиболее важные из них:

1. *Развивающийся процесс поиска:* Пользователь будет изменять или указывать цель или задачи поиска или даже стратегии, используемые для их достижения путем переформулирования или уточнения нескольких запросов. Во время поиска пользователь может выполнить шаги вперед или назад.

2. *Развивающаяся информационная потребность:* На протяжении всего сеанса поиска у пользователя возникает развивающаяся информационная потребность. Найденные элементы или результаты могут изменить его информационную потребность и способ, которым он впервые рассмотрел рамки поиска.

3. *Несколько целей поиска или размытая цель:* Пользователь может иметь не одну единственную точную цель, а одну неопределенную цель и несколько более мелких целей, которые могут изменяться или развиваться во время задачи поискового поиска для ее достижения.

4. *Неопределенность колеблется:* Пользователь начинает поиск с сильного чувства неуверенности. Уровень неопределенности неразрывно связан со спецификацией проблемы. Чем дальше пользователь выполняет свои задачи поиска (он будет определять свою цель и, возможно, определять приблизительный план), тем больше он уменьшает свою неопределенность. Но если где-то по пути он меняет свои цели, неопределенность снова будет расти.

5. *Открытая поисковая активность, которая может происходить со временем:* Пользователь может никогда не закончить свой поисковый поиск. Он может

остановиться по некоторым причинам (он считает, что у него достаточно информации, например, для выполнения другой задачи; у него нет времени на поиск и т.д.), И он продолжит поиск несколько часов / дней / недель / месяцы / годы спустя.

Мы можем увидеть, что привычные поисковики попадают не под все характеристики и, в результате, не могут удовлетворить информационные потребности пользователей. Не мало важную роль играет и сам интерфейс, и возможности поисковика, чтобы действительно предоставить пользователям возможность производить разведочный поиск.

Так, в работе [16] предлагается, чтобы пользователь вводил свой запрос только по предлагаемым ключевым словам. Эти ключевые слова отоброжают основные тематики, которые встречаются в базе. Если пользователь хочет спросить что-то новое, то, к сожалению, ответ на свой запрос он не получит. Главное особенностью предлагаемого в этой работе решения является то, что ответ на запрос, даже если он состоит из одного слова, пытается охватить весь спектр тем, касающихся запроса. Так, например, если вы хотите спросить что-то по анализу данных, то система выдаст не просто определения этого термина, а еще и список технологий и последних новостей, которые могут быть с ним связаны.

В работе [8] делается упор на разнообразие деятельности пользователя – он может углублятся в какую-то тему или, наоборот, откатываться назад. Поэтому, использовалось графовое представление данных для того, чтобы пользователь мог блуждать по разным ссылкам.

Не мало важную роль в упрощении во взаимодействии человека с компьютером при разведочном поиске играет **тематическое моделирование**. В работе [9] приводится ряд случаев, как построение тематик по коллекции документов может помочь пользователю в разведочном поиске:

- **Изучение темы:** отображение общего значения темы в корпусе, осмысленная сортировка слов для описания соответствующей темы и фактические значения слов для предоставления ключевых слов и их относительной важности.
- **Обзор всех тем:** обобщить набор скрытых тем, найденных моделью по корпусу. Это включает в себя следующую информацию: количество тем, их общее значение (или влияние на) корпус и сходства между темами, определенными какой-либо мерой.

- **Нахождение различной многозначной и омонимической семантики терминов:** одним из преимуществ тематических моделей является автоматическое устранение двусмысленности смысловых значений слов в темах. Слово с разными значениями автоматически появляется в разных темах, которые соответствуют его различным семантическим контекстам.

- **Обзор документов по теме:** Определив одну или несколько интересных тем, пользователь может захотеть посмотреть документы, которые охватывают эти темы. Эта задача лежит в основе поискового анализа.

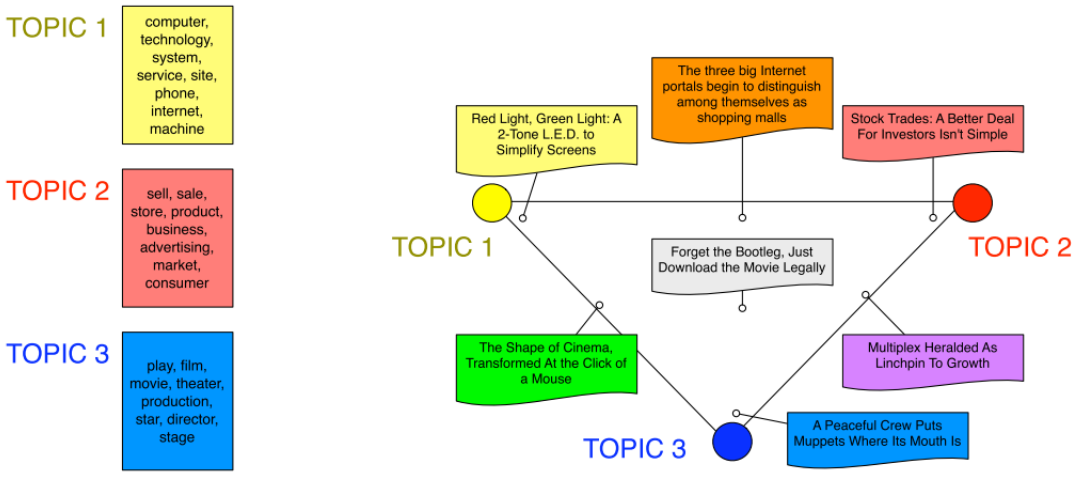
- **Поиск тем, связанных с документом:** После того, как интересный документ был идентифицирован, пользователь может захотеть проверить другие темы, связанные с ним, или, транзитивно, документы, связанные с этими другими темами. Опять же, эта задача направлена на предоставление пользователю инструмента для изучения связанных документов (и, следовательно, корпус) путем выбора интересных тем. В этой задаче содержится следующая информация: интересующий документ, пропорции других тем в этом документе и документы, связанные с документом.

В работе [17] предлагали также вариант интерфейса взаимодействия пользователя с реузльтатами работы тематической модели: см. рис. 1. Здесь предлагается давать пользователю возможность изучать тему (в данном примере, показав список ключевых слов этой темы), а также смотреть темы, которые связаны с конкретно выбранным документом.

Есть отдельные исследования, посвященные визуальным составляющим систем, которые предоставляют возможность осуществления разведочного поиска [4]. Например, изучению подлагались такие вещи, как оптимальный размер выдачи – было выявлено, что для пользователя наиболее полезным оказывается ответ размером в 30 статей. Также изучалось, что необходимо давать на предпросмотр статьи или ссылки и какого размера должен быть текст.

2.3 Методики оценивания систем разведочного поиска

Во многих работах о разведочном поиске вместе с описанием систем идет перечисление методов оценки предлагаемой системы [18, 11]; есть и отдельные работы посвященные обзору разных методик оценки систем разведочного поиска [22, 13].



(a) Topics (b) Document Assignments to Topics

Рис. 1: Скрытое пространство модели состоит из тем, которые являются распределениями по словам и распределением по этим темам для каждого документа. Симплекс изображает распределение по темам, связанным с семью документами. Строка из заголовка каждого документа показывает положение документа в пространстве темы.

В общем, все встречающиеся в этих работах методы можно свести к следующим основным метрикам оценки качества работы системы разведочного поиска:

1. **Длина запроса:** какая средняя длина запроса, которые задают пользователи, и предполагает ли сам интерфейс к вводу коротких или длинных запросов.

2. **Доля просмотра:** доля продолжительности первого запроса, потраченного на просмотр результатов в первой выдаче.

3. **Максимальная глубина прокрутки:** Максимальное количество результатов, отображаемых при прокрутке, т.е. максимальное количество элементов в выдаче, с которыми пользователь столкнулся при прокрутке во время первой итерации запроса.

4. **Количество кликов:** общее количество ссылок в поисковой выдаче, на которые пользователь нажал.

5. **Время, которое пользователи тратят на изучение нажатых документов:** показывает, на сколько тот или иной документ оказался релевантен запросу пользователя.

6. **Время, за которое человек справляется с заданием:** во время оценки

систем часто даются конкретные задания по изучению темы. Этот пункт отображает общее время, которое пользователи проводят с момента выдачи первого запроса до нажатия кнопки «Выполнено».

7. **Степень обучаемости за время поиска:** как говорилось ранее, целью разведочного поиска является обучение. Если человек не был знаком с темой ранее, то после проведения разведочного поиска он должен повысить уровень своих знаний в интересующей тематике.

Большинство этих метрик измерялись на основе анализа логов поведения пользователя в системе. Пользователям давалось задание, которое они должны были выполнить при помощи системы. Также перед экспериментом отмечалось, какого возраста человек, какое имеет образование и степень ознакомленности с предложенной темой.

В работе [10] проводилась оценка качества работы поиска на основе тематической модели коллекции для разведочного поиска. Для этого организатор тестирования составлял список запросов, которые соответствуют тематикам коллекции. Запрос выглядел как текст примерного объема 1–2 страницы формата A4. Далее ассесору необходимо было ознакомится с заданием, понять тематику задания, провести поиск самостоятельно в любом поисковике, а затем оценить качество выдачи тематической модели. Однако, такая оценка качетсва работы системы разведочного поиска искусственая, так как мы просим оценить релевантность документа запросу придуманого нами, а не самим пользователем.

3 Описание системы разведочного поиска

В данном разделе будет приведено описание предлагаемого процесса разведочного поиска, будет поставлена задача ранжирования и криетрии ее оценки, а также будет описан прототип системы разведочного поиска для осуществления экспериментов. Сделанные выводы основываются на пожеланиях пользователей, описанных в разделе 1, и на основе проведенных исследований по системам разведочного поиска, предоставленных в разделе 2.

3.1 Процесс разведочного поиска

Как было описано в разделе 2.2, одной из основных характеристик разведочного поиска является открытая поисковая цель, которая изначально может быть размытой и постоянно меняться в процессе итеративного взаимодействия пользователя с системой. Кроме того, пользователь может в любой момент прекратить или снова продолжить поиск по мере возможности и необходимости получения новой информации. В связи с этими характеристиками предлагается следующий процесс разведочного поиска см. алгоритм 3.1.

Алгоритм 3.1. Итерационный процесс разведочного поиска.

Вход: Коллекция статей D;

Выход: Пользователь получит необходимый объем знаний на текущий момент;

1: Пользователь просматривает ленту и формирует свою первую подборку Q;
2: **цикл** — бесконечный цикл
3: Система дает ранжированный список рекомендаций;
4: **если** В рекомендациях есть подходящие по тематикам статьи **то**
5: Пользователь добавляет понравившиеся статьи в подборку;
6: Пользователь корректирует подборку, удаляя ненужные статьи или пополняя ее еще из ленты;

На входе имеется коллекция статей $D = \{d_i\}_{i=1}^{N}$. Далее пользователь может работать с этой коллекцией и на основе неё сформировать подборку – набор статей, которые сохраняются для данного пользователя. Как видно из шага 2, пользователь постоянно имеет доступ к своей подборке, которую он может в любой момент просматривать, удалять или добавлять в неё статьи. На шаге 3 пользователь может

получить рекомендации от системы, и именно *подборка* пользователя с понравившимися ему статьями является в данном случае поисковым запросом. Пользователь получает рекомендации и оценивает релевантность рекомендованных статей на шаге 5. Эта информация о том, какие статьи пользователь добавил или не добавил из рекомендаций, на каком месте они находились в списке ответа, используется для оценки качества работы системы. Пользователь своим поведением показывает свою удовлетворенность рекомендациями, опираясь на свое желание изучить разные тематики из своей подборки.

3.2 Постановка задачи ранжирования

На шаге 3 алгоритма 3.1 пользователь запрашивает у системы список рекомендаций на основе своей подборки. Эту задачу можно переформулировать как задачу ранжирования документов коллекции, которая сейчас будет рассмотрена более формально.

Пусть имеется коллекция статей $D = \{d_i\}_{i=1}^{N}$ – лента какого-нибудь новостного ресурса. Пользователь из этой ленты собирает себе подборку статей $Q = \{d_j\}_{j=1}^{M}$, которая и является его поисковым запросом. Нашей задачей является для каждого объекта из $d \in D$ построить отображение $r : D \to \mathbb{R}$, которое сопоставит каждому элементу $d \in D$ вес $r(d)$, характеризующей степень релевантности элемента объекту (чем больше вес, тем релевантнее объект). При этом, набор весов $\{r(d)\}_{d \in D}$ задает перестановку $\pi : [1..N] \to [1..N]$ на наборе элементов D исходя из их сортировки по убыванию веса $r(d)$. В итоге, мы получаем отранжированный список статей релевантных по релевантности запросу пользователя Q. Используемые алгоритмы для расчета весов $r(d)$ будут подробно описаны далее в разделе 4.

Далее нам необходимо будет оценить, насколько алгоритм построения оценок $r(d)$ и перестановок π, соответствует истинным значениям релевантности, что в данном случае является критерием удовлетворенности пользователя от полученной информации.

3.3 Методика оценивания

Основываясь на процессе разведочного поиска, описанного в 3.1, была разработана методика оценивания такого поиска. Она основывается на участии ассесоров, которые должны выполнить задания по взаимодействию с системой.

Оценка алгоритмов ранжирования Для начала необходимо оценить в принципе работу алгоритмов ранжирования для подобного поиска, где запросом пользователя может быть подборка из нескольких статей. Для этого ассесору предлагается собрать разного размера подборки из общей ленты: из 1, 5, 10, 15 и 20 статей. Для каждой подборки ассесор получает результат работы каждого алгоритма ранжирования. Ассесор оценивает релевантность статьи запросу путем добавления его себе в подборку. Для каждого запроса определим меры качества поиска:

- **Precision at K (P@K)** — точность на K элементах — базовая метрика качества ранжирования для одного объекта. Допустим, алгоритм ранжирования выдал оценки релевантности для каждого элемента $\{r(d)\}_{d\in D}$. Отобрам среди них первые $K \leqslant N$ элементов с наибольшим $r(d)$ можно посчитать долю релевантных. Именно это и делает precision at K:

$$\text{P@K} = \frac{\sum_{k=1}^{K} r^{true}(\pi^{-1}(k))}{K} = \frac{\text{релевантных элементов}}{K}, \tag{3.1}$$

  где под $\pi^{-1}(k)$ понимается элемент $d \in D$, который в результате перестановки π оказался на k-ой позиции, а r^{true} в данном случае будет принимать значения 0 или 1 – релевантен документ запросу или нет. Данная метрика имеет важный недостаток – она не учитывает порядок элементов в «топе».

- **Average precision at K (AP@K)** нивелирует этот недостаток – она равна сумме P@K по индексам k от 1 до K только для релевантных элементов, деленому на K:

$$\text{AP@K} = \frac{1}{k}\sum_{k=1}^{K} r^{true}(\pi^{-1}(k))\text{P@K} \tag{3.2}$$

- **Mean Average Precision@K (MAP@K)** оценивает качество работы не для отдельного объекта (пользователь, запрос), а усредняет для всех объектов:

$$\text{MAP@K} = \frac{1}{N}\sum_{j=1}^{N} \text{AP@K}_j \tag{3.3}$$

Оценка процесса разведочного поиска Для оценки процесса разведочного поиска ассесорам предлагается выполнить два задания.

Задание 1 Ассесор, когда заходит в систему, может сформировать себе подборку из 1 статьи на любую тему, которая ему интересна – это может быть свой текст или статья из общей ленты. Предполагается, что целью ассесора является получение новых знаний по теме. Процесс работы с системой, как и описано 3.1, должен выглядеть следующим образом:

1. Ассесор формирует подборку. Это стартовая точка его поиска. Для нее он должен оценить работу каждого алгоритма поиска;

2. Далее он выбирает одну и только одну модель, с которой он будет сейчас работать;

3. Ассесор получает в рекомендательную ленту новые статьи по своей подборке. Он оценивает релевантность статьи тем, что добавляет ее из выдачи себе в подборку;

4. Таким образом ассесор может сформировать себе следующую по величине подборку и снова послать запрос системе;

5. Далее шаги 2 и 3 повторяются до тех пор, пока пользователю есть что добавлять из рекомендательной ленты и он чувствует необходимость в пополнении своих знаний.

Задание 2 Сделаем небольшую модификацию Задания 1 и предложим ассесору работать за один раз не с одной моделью, а на каждом шаге выбирать модель, которая, по его мнению, будет давать более релевантные результаты. Весь остальной процесс остается таким же. Это позволит проанализировать, каким моделям на каком шаге и на каких размерах подборок ассесоры отдают большее предпочтение.

Оценка системы разведочного поиска В конце итерационного поиска ассесору предлогается оценить, удовлетворен ли он полученными знаниями, по шкале от 1 ("Ничего полезного не найдено") до 5 ("Да, все было интересное и по теме"). Также ассесору предлагается сравнить полученную информацию в системе с известными поисковыми системами и отдать предпочтение либо предлагаемой в данной работе системе, либо известному поисковику.

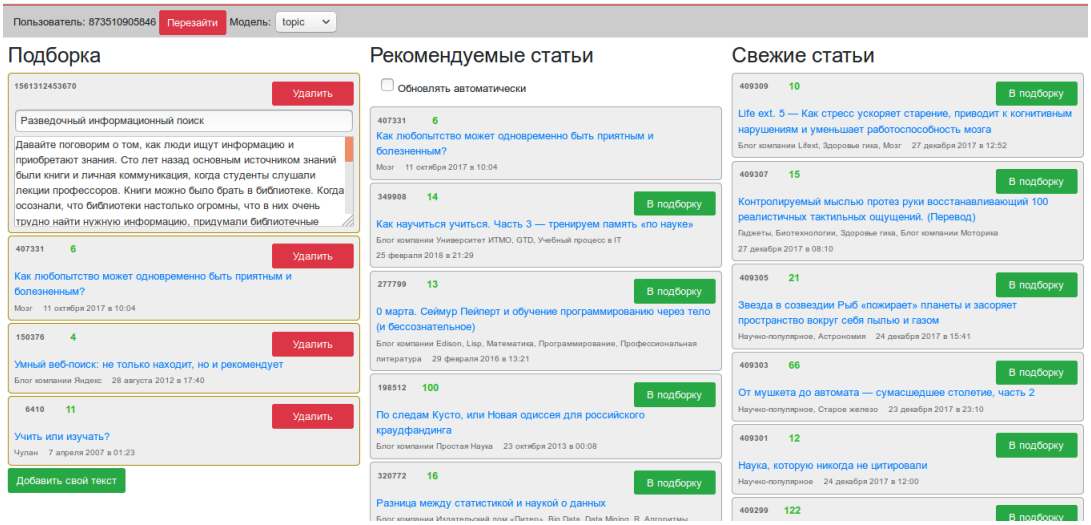


Рис. 2: Прототип системы разведочного поиска. С левой стороны пользователь сформировал свою подборку – добавил свою статью и несколько из ленты. В качестве текущей модель он выбрал тематическую модель и получил рекомендации по своей подборке.

3.4 Описание прототипа системы

Для того, чтобы обеспечить все шаги процесса 3.1, а также иметь возможность собрать ассесорские оценки для оценки качества работы системы, предлагается система разведочного поиска, интерфейс которой представлен на рис. 2.

В ней предоставляются следующие функции:

- С правой стороны страницы в хронологичском порядке представлены все статьи коллекции; с левой отображается подборка пользователя; в центральной – рекомендации системы.
- Пользователь может добавлять новые статьи в свои подборку или удалять из неё ненужные статьи.
- Пользователь может добавлять свой текст статей в подборку, если, например, его тема не предствлена в общей ленте;
- В верхней части интерфейса пользователь может выбрать разные алгоритмы ранжирования;
- После формирования подборки и выбора алгоритма ранжирования пользователь может получить рекомендации системы;
- Если пользователь хочет начать заново и создать полностью новую подборку, то он может сделать это, нажав кнопку в верхней части "Перезайти".

Каждому пользователю присваивается ID, и все его действия по формированию подборок, добавлению статей из рекомендаций логируются. В дальнейшем эти логи испольуются для анализа поведения пользователей и оценки качества работы системы.

4 Алгоритмы ранжирования

В разделе 3.2 была поставлена задача ранжирования. В данном разделе предоставляется описание алгоритмов ранжирования на основе различных векторных представлений текстовых данных.

Введем следующие обозначения:

- D – коллекция всех документов;
- W_D – множество всех слов коллекции;
- Q – подборка пользователя;
- W_Q – множество всех слов подборки.
- $v(d)$ – векторное представление документа d;
- $r(d)$ – релевантность документа d запросу Q.

4.1 Векторные представления «мешка слов»

4.1.1 TF-IDF

TF-IDF расшифровывается как «частота слова - обратная частота документа». Это метод количественного определения слова в документах, в котором обычно вычисляется вес каждого слова, который указывает на важность слова в документе и корпусе. Этот метод широко используется в поиске информации и в текстовом майнинге.

На первом шаге все документы коллекции необходимо представить в виде векторного представления. Для этого необходимо посчитать следующие величины:

$$\mathrm{TF(w|d)} = \frac{n_{wd}}{n_d} \tag{4.1}$$

$$\mathrm{IDF}(w|D) = \log \frac{|D|}{N_w} \tag{4.2}$$

$$\mathrm{TF\text{-}IDF}(w|d) = TF(w|d) \times IDF(w|D) \tag{4.3}$$

где n_{wd} – число вхождений слова w в документ d; n_d – число всех слов в документе d; N_w – число документов, содержащих слово w; TF (*term frequency*) – частота слова в пределах отдельного документа; IDF (*inverse document frequency*) –

инверсия частоты, с которой некоторое слово встречается во всей коллекции. Иногда, во избежание вычислиетельных ошибок, используется модификация вычисления $\text{IDF}(w|D) = \log \frac{|D|+1}{N_w+1}$.

Теперь у нас есть векторные представления всех документов коллекции, где для конкретного документа d каждый элемент вектора $v(d)$ представляет собой TF-IDF$(w|d)$, $\forall w \in W_D$.

Релевантность документа $r(d)$ запросу Q вычисляется на основе следующих соображений – запрос пользователя Q представляется в виде «мешка слов», к $r(d)$ добавляются TF-IDF$(w|d)$ для слов w, которые присутсвуют и в запросе, и в документе:

$$r(d) = \sum_{w \in W_Q \bigcap W_D} \text{TF-IDF}(w|d) \tag{4.4}$$

4.1.2 BM25

BM25 – поисковая функция, которая также основывается на предположении «мешке слов» и множестве документов, которые она оценивает на основе встречаемости слов запроса в каждом документе, без учёта взаимоотношений между ними. Данную функцию можно рассматривать как модификацию выше описанного TF-IDF. Оценка релевантности документа d в данном случае будет выглядет так:

$$r(d) = \sum_{w \in W_Q} \text{IDF}(w|D) \times \frac{n_{wd}(k_1+1)}{n_{wd} + k_1(1 - b + b\frac{|d|}{avdl})} \tag{4.5}$$

где $avgdl$ — средняя длина документа в коллекции, k_1 и b — свободные коэффициенты. BM25 построена на предположении теоретической модели, согласно которой каждый текстовый документ представляется как смесь двух Пуассоновских распределений. Одно из них отвечает за распределение обычных слов, другое – за распределение тех слов, на которых лежит основная смысловая нагрузка в разрезе рассматриваемой тематики. Таким образом, BM25 придаёт больший вес «значимым» термам и меньший – «незначимым».

Данные подохды, основанные на предположении о «мешке слов», показывают хорошие результаты во многих задачах анализа текстов. Однако, они имеют ряд существенных недостатков:

- Размер корпуса всех слов W может быть очень большим, что приведет к большой размерности векторов $v(d)$;

- Они совсем не учитывают взаимное расположение слов и позиции слов в документе;
- Совершенно не учитывается само смысловое значение слов и тематики, которые могут быть заложены в документах.

4.2 Нейросетевые векторные представления

В последнее время значительное распространение получили векторные представления слов, которые обучаются при помощи нейросетевых моделей. Традиционные подходы, такие как one-hot кодирование и модели «мешка слов» не собирают информацию о значении или контексте слова. Это означает, что потенциальные отношения, такие как контекстная близость, не отражаются в наборах слов. Напротив, нейросетевые векторные представления представляют слова в виде многомерных вещественных чисел, где семантически сходные слова сопоставляются с ближайшими точками в геометрическом пространстве. Проще говоря, вектор слов – это ряд вещественных чисел (в отличие от фиктивных чисел), где каждая точка отражает размерность слова и где семантически похожие слова имеют похожие векторы.

Одним из таких подходов обучения векторных представлений является подход, описаный в [7] и получивший название fasttext. Данная модель строит векторные представления не для отдельных слов, а для символьных n-грамм.

4.2.1 Обучение векторных представлений

Имея корпус слов размера W, где слово идентифицируется по его индексу $w \in \{1, ..., W\}$, цель состоит в том, чтобы получить векторное представление для каждого слова w. Если мы обратимся к skipgram модели, которая была еще описана в работе [5], главной целью при обучении векторных представлений слов явлется максимизация следующей функции правдоподобия:

$$\sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c | w_t) \tag{4.6}$$

где C_t – набор индексов слов, которые находятся в контексте слова w_t.

То есть, одной из целью является предсказать контектсное слово w_c в окружении слова w_t. Тогда задача прогнозирования контекстных слов может быть сформулирована как набор независимых задач бинарной классификации. Цель состоит в том,

чтобы самостоятельно предсказать наличие (или отсутствие) контекстных слов. Для слова на позиции t мы будем рассматривать все контекстные слова как положительные примеры и выберем случайные негативные примеры из словаря. Для выбранной позиции контекста c, используя бинарную логистическую функцию потерь, мы получаем следующую отрицательную функцию правдоподобия:

$$\log\left(1 + e^{-s(w_t, w_c)}\right) + \sum_{n \in N_{t,c}} \log\left(1 + e^{s(w_t, n)}\right) \tag{4.7}$$

где функция s сопоставляет пары (слово, контекст) с оценками в $\mathbb{R}$, $N_{t,c}$ – набор негативных примеров, взятых из словаря.

Естественная параметризация для скоринговой функции s между словом w_t и контекстным словом w_c заключается в использовании векторных представлений слов. Определим для каждого слова w из словаря два вектора u_w и v_w в $\mathbb{R}^d$. Эти два вектора иногда обозначаются как входные и выходные векторы. В частности, у нас есть векторы u_{w_t} и v_{w_c}, соответствующие словам w_t и w_c. Тогда оценка может быть вычислена как скалярное произведение между векторами слова и контекста как $s(w_t, w_c) = u_{w_t}^\top v_{w_c}$.

Однако, используя отдельное векторное представление для каждого слова, skipgram модель игнорирует внутреннюю структуру слов. Теперь предположим, что нам дан словарь из n-грамм размера G. Для слова w обозначим через $G_w \subset \{1, ..., G\}$ множество n-грамм, фигурирующих в w. Мы связываем векторное представление z_g с каждой n-граммой. Тогда векторное предсталвние слова будет представляться в виде суммы векторных представлений его n-грамм. Таким образом, мы получаем функцию оценки:

$$s(w, c) = \sum_{g \in G_w} z_g^\top v_c \tag{4.8}$$

Такая достаточно простая модель позволяет лучше передавать смысл слов в виде векторных представлений.

4.2.2 Функция ранжирования

Для того, чтобы оценить релевантность документов из ленты запросу пользователя, необходимо представить текст документа из ленты и текст запроса в виде векторных представлений.

Основываясь на выше описанном подходе n-грамм, если у нас есть корпус n-грамм G, для которых определены их векторные представления $v(g)$, а каждый документ d представить в виде набора n-грамм $G_d \subset G$, то тогда можем получить следующее векторное представление документа d:

$$v(d) = \frac{1}{|G_d|} \sum_{g \in G_d} v(g) \tag{4.9}$$

Если у документа кроме самого содрежания m еще известны его название n, а также теги h, то векторное представлние всего документа можно представить как взвешенную сумму векторных представлений всех его составных частей:

$$v(d) = \alpha v(m) + \beta v(n) + \gamma v(h) \tag{4.10}$$

В таком случае оценка релевантности документа может основываться на косинусной близости векторов запроса Q и документа d:

$$r(d) = sim\Big(v(Q), v(d)\Big) = \frac{\sum_i v(Q)v(d)}{(\sum_i v(Q))^{\frac{1}{2}}(\sum_i v(d))^{\frac{1}{2}}} \tag{4.11}$$

4.3 Тематические векторные представления

4.3.1 Задача вероятностного тематического моделирования

Пусть D – коллекциятекстовых документов, W – множество употребляемых терминов в документе (причем, в качестве терминов могут выступать слова, биграммы, буквенные и словесные n-граммы). Любой документ $d \in D$ представляет собой последовательность слов из словаря: $(w_1, ..., w_{n_d}) \subset W$, где каждому термину соответствуетчисло его вхождений n_{dw}. Таким образом, матрица частот F для текстовой коллекции D будет выглядеть так:

$$F = (f_{wd})_{W \times D} \tag{4.12}$$

$$f_{wd} = \frac{n_{wd}}{n_d} \tag{4.13}$$

Предполагается, что что существует конечное множество тем T, и каждое появление токена w в документе d связано с темой $t \in T$, которая заранее не известна. Коллекция документов рассматривается как случайнаяи независимая выборка троек $(w_i, d_i, t_i), i = 1..n$ из дискретного распределения $p(w, d, t)$ на конечном вероятностном пространстве $W \times D \times T$.

Вероятностная тематическая модель будет выглядеть следующим образом [2, 19]:

$$p(w|d) = \sum_{t\in T} p(w|t)p(t|d) = \sum_{t\in T} \varphi_{wt}\theta_{td},\ d \in D, w \in W, \tag{4.14}$$

где $p(w|t)$ – вероятность встречания термина $w \in W$ в теме $t \in T$, $p(t|d)$ – вероятность встречания темы $t \in T$ в документе $d \in D$. Параметры $\varphi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$ образуют матрицы $\Phi = (\varphi_{wt})_{W\times T}$ – дискретные распределения слов для тем и $\Theta = (\theta_{td})_{T\times D}$ – дискретные распределения тем для документов. Из этого следует, что эти матрицы являются стохастическими (каждый столбец представляет собой дискретное распределение).

Тогда задача тематического моделирования звучит следующим образом: по заданной коллекции D найти множество тем T и оценить неизвестные матрицы Φ и Θ. Чтобы найти неизвестные элементы матриц Φ и Θ по наблюдаемой коллекции документов, максимизируем логарифм правдоподобия наблюдаемой коллекции:

$$\mathcal{L}(\Phi, \Theta) = \sum_{d\in D}\sum_{w\in W} n_{dw} \ln p(w|d) \to \max_{\Phi,\Theta} \tag{4.15}$$

С учетом выражения для $p(w|d)$ и стохастичности матриц получаем задачу условной оптимизации:

$$\sum_{w\in W}\sum_{d\in D} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} \to \max_{\Phi,\Theta} \tag{4.16}$$

$$\sum_{w\in W} \varphi_{wt} = 1, \varphi_{wt} \geqslant 0, \tag{4.17}$$

$$\sum_{t\in T} \theta_{td} = 1, \theta_{td} \geqslant 0 \tag{4.18}$$

4.3.2 Аддитивная регуляризация тематических моделей

Задача тематического моделирования в том виде, в котором она была поставлена в 4.16, является некорректно поставленной по Адамару, так как в общем случае множество её решений является бесконечным. Для устранения недоопределенности используют подход, который называется регуляризацией. В рамках аддитивной регуляризации тематических моделей (ARTM) [2, 19] дополнительные регуляризаторы рассматриваются в виде линейной комбинации, добавляемой к основному функционалу:

$$\sum_{w\in W}\sum_{d\in D} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} + \sum_i \tau_i R_i(\Phi\Theta) \to \max_{\Phi\Theta}, \tag{4.19}$$

где τ_i – неотрицательный коэффициент регуляризации.

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

$$R(\Phi, \Theta) = \beta_0 \sum_{t \in T} \sum_{w \in W} \beta_w \ln \varphi_{wt} + \alpha_0 \sum_{d \in D} \sum_{t \in T} \alpha_t \ln \theta_{td} \tag{4.20}$$

В силу свойств кросс-энтропии максимизация данного функционала приводит к тому, что распределения $\vec{\varphi}_t$ и $\vec{\theta}_d$ становятся похожи на фиксированные распределения $\vec{\beta}$ и $\vec{\alpha}$ соответственно. При выборе равномерных распределений этоприводит к сглаживанию тем.

Регуляризатор разреживания имеет такую же форму, что и сглаживающий регуляризатор 4.20, но только со знаком минусом перед коэффийиентами β_0 и α_0. Разреживающий регуляризатор максимизирует кросс-энтропию, заставляя распределения $\vec{\varphi}_t$ и $\vec{\theta}_d$ становится непохожими на распределения $\vec{\beta}$ и $\vec{\alpha}$ соответственно.

Регуляризатор декоррелирования используется для уменьшения похожести (корреляции) тем между собой. Для этого миними-зируется сумма попарных ковариаций между всеми парами тем:

$$R(\Phi) = -\tau \sum_{t,s \in T} \sum_{w \in W} \varphi_{wt} \varphi_{ws} \tag{4.21}$$

Доказано,что комбинация этих трех регуляризаторов повышает интерпретируемость тем [20, 1].

Обучение Пусть функция $R(\Phi, \Theta)$ непрерывно дифференцируема. Известно [2, 19], что в таком случае локальный максимум задачи 4.16 удовлетворяет следующей системе уравнений со вспомогательными переменными $p_{tdw} = p(t|d, w)$:

$$p_{tdw} = \operatorname{norm}_{t \in T}(\varphi_{wt} \theta_{td}); \tag{4.22}$$

$$\varphi_{wt} =_{w \in W} \left(n_{wt} + \varphi_{wt} \frac{\partial R}{\partial \varphi_{wt}} \right); \qquad n_{wt} = \sum_{d \in D} n_{dw} p_{tdw}; \tag{4.23}$$

$$\theta_{td} =_{t \in T} \left(n_{td} + \theta_{td} \frac{\partial R}{\partial \theta_{td}} \right); \qquad n_{td} = \sum_{w \in d} n_{dw} p_{tdw}; \tag{4.24}$$

где оператор $\text{norm}_{t\in T}(x_t) = \frac{\max\{x_t,0\}}{\sum_{s\in T}\max\{x_s,0\}}$.

Решение данной системы уравнений с помощью метода простой итерации эквива-лентно EM-алгоритму, где на E-шаге мы пересчитываем значения p_{tdw}, а на M-шаге вычисляем φ_{wt} и θ_{td}.

4.3.3 Иерархическое тематическое моделирование

Одним из расширением тематических моделей является построение тематических иерархий. 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. Для построения иерархическим моделей вводятся дополнительные регуляризаторы [3].

Регуляризатор межуровневых связей На верхнем уровне иерархии строится обычная плоская тематическая модель. Рассмотрим процесс построения следующих уровней иерархии. Пусть модель l-того уровня с множеством тем T уже построена, и требуется построить модель уровня $l+1$ с множеством дочерних тем S и большим числом тем, $|S| > |T|$. Потребуем, чтобы родительские темы t хорошо приближались вероятностнымисмесями дочерних тем s:

$$\sum_{t\in T} n_{tw}\Big(p(wt)\|_{s\in S}p(ws)p(st)\Big) = \sum_{t\in T} n_{tw}\Big(\frac{n_{wt}}{n_t}\|_{s\in S}\varphi_{ws}\psi_{st}\Big) \to \min_{\Phi,\Psi}, \tag{4.25}$$

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 $l+1$ так, чтобы родительские темы φ_t^l аппроксимировались линейными комбинациямидочерних тем φ_s с коэффициентами ψ_{st}:

$$R(\Phi,\Psi) = \tau\sum_{t\in T}\sum_{w\in W} n_{wt}\ln\sum_{s\in S}\varphi_{ws}\psi_{st}. \tag{4.26}$$

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

Регуляризатор разреживания межуровневых связей Рассмотрим редположение, что каждая тема дочернего уровня $s \in S$ имеетнебольшое число связей с те-

мами родительского уровня $t \in T$. В частности, еслиvсе распределения $p(ts)$ вырождены, то есть каждая тема s имеет только однородительскую тему t, о вся иерархия приобретает вид дерева. Применим кросс-энтропийный регуляризатор для разрежи-вания распределений $p(ts)$. Выражаем $p(ts)$ через ψ_{st}:

$$R(\Psi) = -\tau \sum_{s \in S} \sum_{t \in T} \frac{1}{|T|} \ln p(ts) = -\frac{\tau}{|T|} \sum_{t \in T} \sum_{s \in S} \ln \frac{\psi_{st} n_t}{\sum_z \psi_{sz} n_z}. \tag{4.27}$$

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

$$\psi_{st} =_{s \in S} \left(n_{st} + \tau \Big(p(ts) - \frac{1}{|T|} \Big) \right) \tag{4.28}$$

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

4.3.4 Функция ранжирования

В данном случае векторным представлением текста d будет соответствующий тематический вектор $\vec{\theta}_d = p(t|d)$. Для документов из ленты $d \in D$ эти вектора извлекаются из предпосчитанной тематической моделью матрицы Θ_D. Для входного запроса сначала строится тематическое представление текста Φ_Q и Θ_Q, затем берется соответствующий вектор $\vec{\theta}_q$ из матрицы Θ_Q.

Предположим, что уже построена тематическая модель с двумя уровнями иерархий. Тогда для каждого документа d известны его тематические векторные представления на каждом уровне иерархий – $\vec{\theta}_d^0$ и $\vec{\theta}_d^1$. Оценка релевантности документа из коллекции запросу происходит в два этапа:

1. На первом шаге происходит предварительный отсев: $r(d)$ присваивается 0 для тех документов, в которых тематики запроса практически не присутсвуют, т.е. $\theta^0_{di} < \Delta,\ \forall i, \theta^0_{qi} > 0$;

2. На втором шаге оценивается релеватность запросу оставшихся документов, которая, как и в 4.2.2 описывается косинусной близостью векторов:

$$r(d) = sim(\vec{\theta}_q, \vec{\theta}_d) = \frac{\sum_t \theta_{tq} \theta_{td}}{(\sum_t \theta^2_{tq})^{\frac{1}{2}} (\sum_t \theta^2_{td})^{\frac{1}{2}}}. \tag{4.29}$$

5 Эксперименты

В данной разделе приведены результаты экспериментов с предлагаемой системой разведочного поиска, а также результаты оценки системы ассесорами по методике, которая описана в главе 3.3.

Для того, чтобы оценить, как и зачем люди из разных профессиональных сообществ используют поиск, а также удовлетворены ли они результатом работы имеющихся систем, был проведен социальный опрос. В нем приняло участие 50 человек, средний возраст опрашиваемых был 20-25 лет. Среди них были представители разных сфер ИТ – front-end и back-end разработчики, специалисты по анализу данных, менеджеры и HR. Опрашиваемым необходимо было ответить на следюущие вопросы:

1. Перечислить новостные ресурсы, которые человек читает вообще;
2. Перечислить новостные или образовательные ресурсы, которые человек читает именно для саморазвития и самообразования;
3. Необходимо ли ему искать новую информацию по работе и для профессионального развития;
4. Считает ли себя исследователем или любит получать информацию уже в готовом обработанном виде;
5. Сколько времени тратит на поиск в интернете в день;
6. Сталкивается ли с проблемами при поиске и усвоении информации, и, если да, то с какими.

В результате опроса получилась такая статистика:

- 100% опрашиваемых необходимо проводить исследования – искать новую информацию, изучать новую тему – по работе;
- 50% опрашиваемых следят за новостями в своей профессиональной сфере;
- 50% опрашиваемых сталкиваются с разнообразными трудностями при поиски информации;
- в среднем, пользователи тратят около 3 часов в день на поиск.

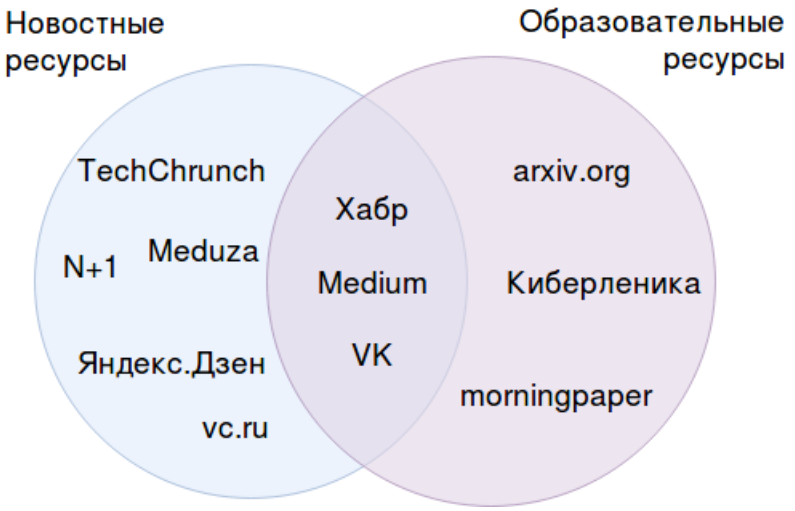


Рис. 3: Диаграмма категоризации ресурсов, которые пользователи используют в повседневной жизни.

Что касается выбора новостных ресурсов, то, сопоставив результаты по двум пунктам, можно получить следующие результаты см. рис. 3.

Наиболее частым ответом среди участников опроса в обеих категориях были Хабр (`https://habr.com`), Medium (`https://medium.com`), а также новостная лента Вконтакте (`https://vk.com`), где пользователи могут состоять в разных сообществах и читать посты новостного и образовательного содержания.

Половина всех опрашиваемых отметила, что, не смотря на все нынешнее разнообразие ресурсов и возможностей поиска, они все равно ежедневно сталкиваются с проблемами поиска нужной информации и, даже потратив много времени, не всегда достигают нужного результата. Это говорит о том, что есть пользовательская необходимость в новых концепциях поиска. И разведочный поиск может стать одним из решений.

5.1 Описание и предобработка данных

Эксперименты проводились на текстовой коллекции новостного ресурса Хабр. Новости с данного ресурса имеют техническую направленность, наиболее полно представлены обзоры о новых технологиях, IT-сфере стартапах и т.д. Вся коллекция состоит из 136618 статей. Информация по каждой статье состоит из:

- заголовка статьи;
- атвора статьи;

- основного текста статьи без картинок и вставок кода;
- даты публикации;
- хабы (теги), к которым отнесена статья.

На этапе предобработки данных из текста была исключена информация, все слова из заголовка, текста и тегов статьи были приведены к нижнему регистру и к начальной форме (с помощью библиотеки pymorphy2), а также из текста удалялись стоп-слова и слова длиной меньше 5 симовлов.

Было сделано предположение выдавать в рекомендательной ленте всегда ответ длиной только в 50 статей, т.е. для всех вычсляемых метрик $K = 50$.

5.2 Результаты

В эксперименте по оценка качества работы с системой приняло участие 15 ассесоров. Перед началом эксперимента ассесоры проходили анкетирования, в результате которого выяснилось: 9 человек были возраста 21-25 лет, 4 человека возраста 26-30 лет, а также 2 человека возраста 15-21 год. 2 человка получили уровень образования аспирантура, 7 человек – увроня магистратуры, 5 человек – бакалавриата, 1 человек – среднее (общее) образование. Все ассесоры являлись представителями сферы IT разных профессий: специалисты по анализу данных, разработчики мобильных приложений, технические писатели, тестировщики. При этом, тематики интересов ассеров были очень разнообразные – ИТ-бизнес, стартапы, математика, дизайн, технологии будущего, физика, путешествия, музыка, психология.

5.2.1 Оценка работы алгоритмов поиска

Первым заданием для ассесоров было оценить качества работы алгоритмов поиска. Ассесоры оценивали качество работы алгоритма по времени и по AP@K. Каждый ассесор построил по 3 подборок каждого размера – из 1, 5, 10, 15 и 20 статей. Результаты работы предсталвены в таблицах табл. 4 и табл. 5. Также на рис. **??** представлены более подробные резульаты оценок AP@K для каждой модели и для подборки каждого размера.

Хотя модель TF-IDF показала хорошие результаты, работает она значительно дольше, чем все остальные модели. Это связано с недостатоком моделей «мешка

|Q|	TF-IDF	BM25	Fasttext	Topic
1	25	10	**2**	5
5	600	80	**10**	12
10	980	195	**12**	20
15	1800	220	**18**	25
20	2100	350	**20**	27

Таблица 4: Среднее время работы (в секундах) каждого алгоритма поиска на подборках разных размеров.

|Q|	TF-IDF	BM25	Fasttext	Topic
1	0.55	**0.56**	0.43	0.48
5	0.6	**0.61**	0.43	0.51
10	0.6	0.53	0.44	**0.61**
15	**0.65**	0.6	0.36	**0.65**
20	**0.71**	0.6	0.39	0.59

Таблица 5: Занчение MAP@50 для каждого алгоритма поиска на подборках разных размеров.

слов» – они имеют большую размерность, что приводит к очень долгим вычислениям. BM25 тоже обладает таким недостатком, но он проявляется гораздо меньше. Кроме того, эта модель показала хорошие результаты для подборок небольших размеров, всего немного опередив TF-IDF. Тематическая модель проигрывает на подборках размеров 1 и 5, но показывает себя лучше на подборках больших размеров. И, самое главное, не имеет такого недостатка по скорости работы, как TF-IDF и BM25. Fasttext модель, не смотря на свое значительное превосходство по скорости работы, показывает самые худшие результаты по точности рекомендаций.

5.2.2 Оценка процесса разведочного поиска

Во втором задании каждому ассесору необходимо было создать стартовую подборку, а затем на протяжении нескольких итераций получать ответ от системы и пополнять свою подборку, пока ассесор не решит прекратить поиск. Каждый ассесор создал по три таких стартовых подборки.

Задание 1 На первом этапе ассесору предлагалось работать только с одной моделью до конца эксперимента с подборкой. Было оценено: медиана количества статей,

которые были добавлены на каждом шаге (табл. 6) и распределение финального размера подборки, который получился у ассесора по завершению эксперимента (рис. 4).

	Шаг1	Шаг2	Шаг3	Шаг4	Шаг5	Шаг6	Шаг7
TF-IDF	3	3	5	3	0	0	0
BM25	5	3	4	2	2	1	1
Fasttext	1	1	2	0	0	0	0
Topic	3	3	2	3	1	1	2

Таблица 6: Медианное количество статей, которые ассесоры добавляли на каждой итерации при работе с каждым алгоритмом.

В этом эксперименте на работе ассесоров с системо сказались все недостатки каждого алгоритма ранжирования. При работе с TF-IDF ассесоры не хотели работать дальше подборки размеров 4-6 статей, так как этот алгоритм ранжирования работает очень долго, что полностью нивелирует достаточно неплохую релевантность статей его рекомендательной выдачи. Fasttext, как и при первом анализе, показывал плохую точность, что не позволяло ассесорам набрать нужное количество статей в подборку по своей тематике. Topic показывал разные результаты – для некоторых тем он показывал приемлемую точность на протяжении всей работы ассесора с системой, для некоторых точность была слишком низкая, чтобы набрать статьи в подборку. BM25 на первых итерациях показал себя лучше всего – ассесоры добавляли статьи в подборку на каджой итерации. Однако, после того, как из подборка достигала размера 7-10 статей, алгоритм ранжирования работал заметно медленнее и рекомендательная выдача была не очень релеватной, что заставляло ассесоров останавливать поиск на этом шаге.

Задание 2 Потом ассесорам было предложено выбирать любую модель на любом шаге для пополнения подборки. Было оценено: количество раз алгоритм ранжирования был выбран на каждом шаге (табл. 7) и распределение финального размера подборки, который получился у ассесора по завершению эксперимента (рис. 4).

Можно заметить, что данный эксперимент позволил нивелировать недостатки каждой из моделей и позволил ассесорам гораздо дольше проработать с системой. Полученные результаты подтверждают выводы, которые были сделаны ранее. Модель TF-IDF и BM25 показывают себя на небольших подборках гораздо лучше, чем

	Шаг1	Шаг2	Шаг3	Шаг4	Шаг5	Шаг6	Шаг7	Шаг8	Шаг9	Шаг10
TF-IDF	15	20	10	7	8	5	2	0	0	0
BM25	20	18	23	17	16	10	5	4	0	0
Fasttext	3	2	2	10	6	10	5	8	9	7
Topic	7	5	10	10	15	9	15	8	10	10

Таблица 7: Количество раз, которое алгоритм ранжирование был выбран на каждой итерации.

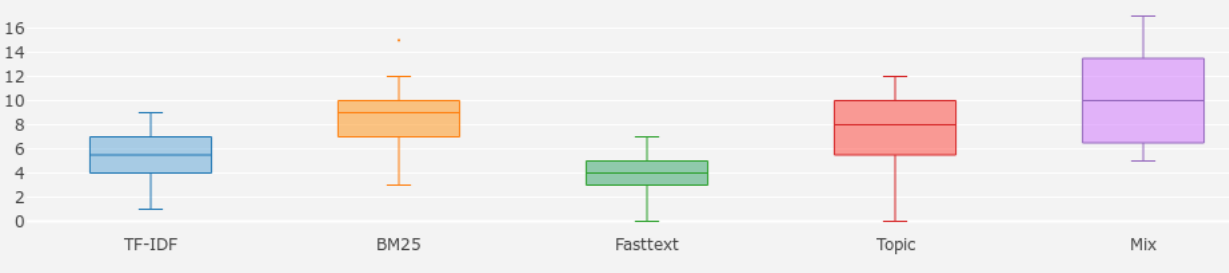


Рис. 4: Распределение размера подборок, которые получились на финальном шаге у ассесоров.

Fasttext и Topic. Разница в работе составляет несколько десятков секунд, но это не сильно сказывается на первых этапах на работе с этими моделями. Однако, когда подборка достигает примерно размера в 4-5 статей и больше, TF-IDF работает слишком долго и ассеоры не готовы выделять столько времени на ожидание выдачи.

На подборках размеров 6-9 статей модели BM25 и Topic работают достаточчно хорошо и позволяют ассесорам дальше набирать новые статьи в подборку. Но начиная с этого этапа в выдачи BM25 появляется большое количество дубликатов и много однотипных статей, слишком сильно похожих по содержанию на те, что уже у ассесора в подборке. Тогда ассесоры начинали гораздо чаще добавлять статьи их выдачи от Topic и Fasttext. Было отмечено, что полученные рекомендации от этих моделей не всегда были действительно релеванты к подборке. Но они предлагали статьи, которые были из смежных областей. А это позволяло разнообразить подборку и расширить кругозор пользователя.

В результате эксперимента можно сделать вывод, что для достижения всех целей пользователя по работе с системой стоит использовать комбинацию алгоритмов ранжирования. Пока пользователь задает короткие запросы из пары статей, то лучше работают модели на основе «мешка слов» – они позволяют ему набрать в подборку достаточное количество статей для первого знакомства с темой. Далее же лучше давать рекомендации от моделей, которые в векторных представлениях слов кодируют смыслы и темы, так как это позволяет давать рекомендации из тем, похожих или

смежных с темой пользователя, что для него, после первого ознакомления с темой, будет гораздо полезнее.

5.2.3 Оценка работы системы разведочного поиска

Ассесорам предлогалось оценить, на сколько им была полезна система и смогли ли они за все время получить новые знания. Ассесорам предлогалось оценить достижение своей цели работы с системой разведочного поиска по шкале от 1 до 5, где «5» – «Все было новое по теме», «4» – «Что-то нашлось новое по теме», «3» – «Нейтральное», «2» – «Особо ничего не нашлось», «1» – «Ничего полезного не нашлось». Также ассесорам необходимо было сравнить результаты работы с системой разведочного поиска с поисковой системой, которая есть на сайте Хабра. Результаты представлены на рис 5.

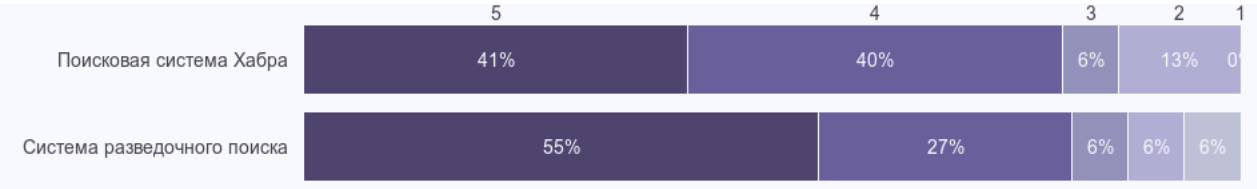


Рис. 5: Ответы ассесоров на вопрос "Смогли вы получить новые знания по вашей теме?".

В конце всего эксперимента, когда они смогли достаточно поработать и с предлагаемой системой разведочного поиска, и с поисковой системой Хабра, ассесорам предлагалось оценить, какой системе они отдадут предпочтение в конечном счете, чтобы продолжить изучать интересующую их тему. Результаты представлены на рис 6.

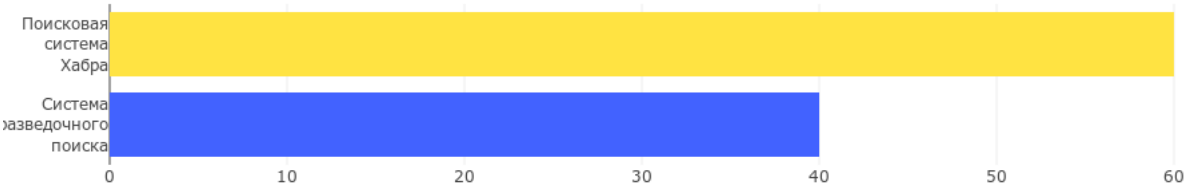


Рис. 6: Ответы ассесоров на вопрос "Какой системе вы отдаете предпочтение?".

Из 15 ассесоров, принимавших участие в оценивании, 8 оценили систему разведочного поиска как очень полезную. Притом, что поиск Хабра очень полезным посчитали 6 человек. 1 ассесор остался совсем не доволен работой системы разведочного поиска. Это было связано с долгой работой моделей TF-IDF и BM25 и что он

ничего не нашел по своей теме. Также он не смог ничего найти и с помощью Fasttext. Тема была достаточно специфической, и есть предположение, что в коллекции Хабра она представлена малым количеством статей.

В конечном итоге, ассесоры в большинстве отдали предпочтение поисковой системе Хабра – 6 человек проголосовало за систему разведочного поиска, 9 человек отдали предпочтение Хабру. Однако, это неплохой результат. Многие ассесоры отмечали, что с помощью предлагаемой системы, в частности, при помощи тематического поиска, они смогли найти статьи по теме, которые помогали взглянуть на нее с другой стороны. Иногда это были статьи на стыке тем или из смежных тем, что позволяло расширить кругозор. В общем, ассесоры отмечали полезность и перспективу развития системы разведочного поиска.

5.3 Пример работы с системой

Приведем иллюстрацию того, как может происходить итеративный разведочный поиск при помощи описываемой системы.

Например, пользователь изучить анализ данных. В качестве старта он выбирает статью «Анализ данных — основы и терминология» с Хабра, где описываются базовые понятия в этой теме. Однако пользователь хочет пойти дальше и узнать больше. Итак, сравним работу алгоритмов на первой итерации (см. рис. 7 табл. 8).

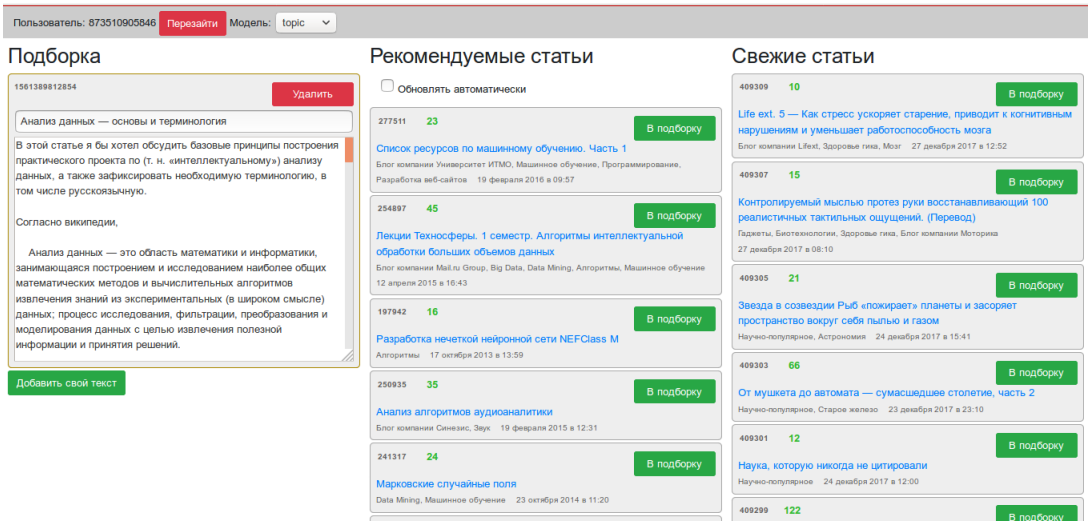


Рис. 7: Пример работы тематического поиска на запрос из статьи про анализ данных.

В принципе, все модели угадали тематику. Только у Fasttext 2 из 3 статей из ТОП не соотносится с темой. И TF-IDF, и BM25, и Topic порекомендовали на самых пер-

Запрос	TF-IDF	BM25	Fasttext	Topic
Анализ данных — основы и терминология	Рециркуляционные нейронные сети, Список ресурсов по машинному обучению, Нейронная сеть как предиктор	Краткий курс машинного обучения, Метод Виолы-Джонса, Детектирование частей тела	Вероятностное программирование, Сравнение технологических подходов, Реальность повторного использования	Список ресурсов по машинному обучению, Лекции Техносферы. 1 семестр, Разработка нечеткой нейронной сети

Таблица 8: ТОП3 выдачи агоритмов поиска по запросу из 1 статьи «Анализ данных — основы и терминология».

вых позициях образовательные ресурсы, которые помогли бы пользователю дальше продвинуться по теме. В данном случае, особо нет разницы, что добавлять в подборку. Для примера, мы добавим статью «Список ресурсов по машинному обучению» из ТОП1 алгоритма Topic. Снова получаем результаты работы системы (см. рис. 8, табл. 9).

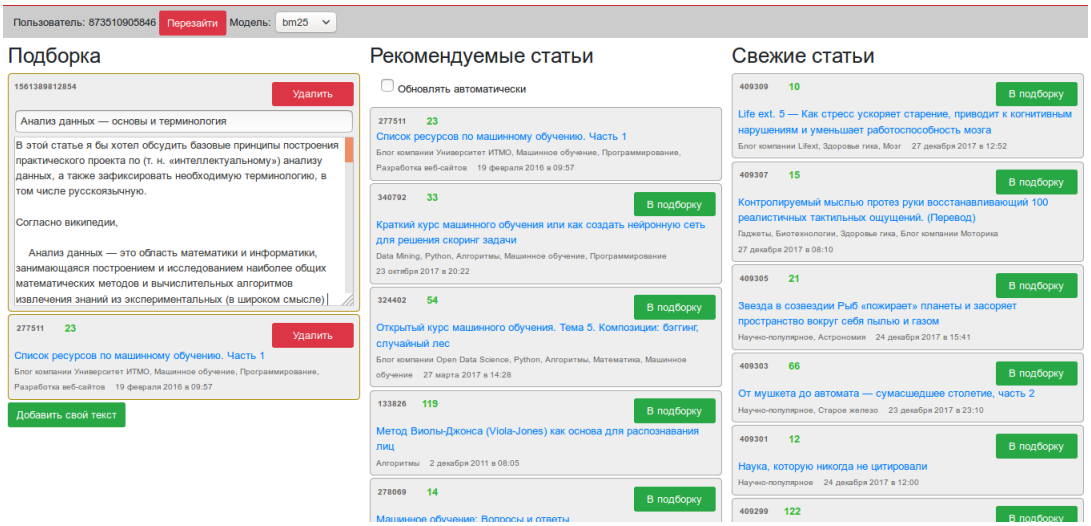


Рис. 8: Пример работы алгоритма BM25 на второй итерации поиска.

Модели Topic и BM25 опять повторили свои выдачи и рекомендуют образовательные темы. Это подходит по тематике, но хочется большего разнообразия. Fasttext не показывает ни одной релевантной статьи по теме. Для того, чтобы разнообразить подборку и спектр подтем, добавим статьи «Рециркуляционные нейронные сети» и «Генерация всех произвольных деревьев» из выдачи TF-IDF. Получаем результаты

Запрос	TF-IDF	BM25	Fasttext	Topic
Анализ данных — основы и терминология, Список ресурсов по машинному обучению	Рециркуляционные нейронные сети, Генерация всех произвольных деревьев, Нейронная сеть как предиктор	Краткий курс машинного обучения, Открытый курс машинного обучения Метод Виолы-Джонса	Элементы функционального программирования, Введение в компонентно-ориентированный подход, Диаграмма цикличной причинности	Лекции Техносферы. 1 семестр, Марковские случайные поля, Спектральный анализ сигналов

Таблица 9: ТОП3 выдачи агоритмов поиска по запросу из 2 статьей.

работы алгоритмов (см. рис. 9, табл. 10).

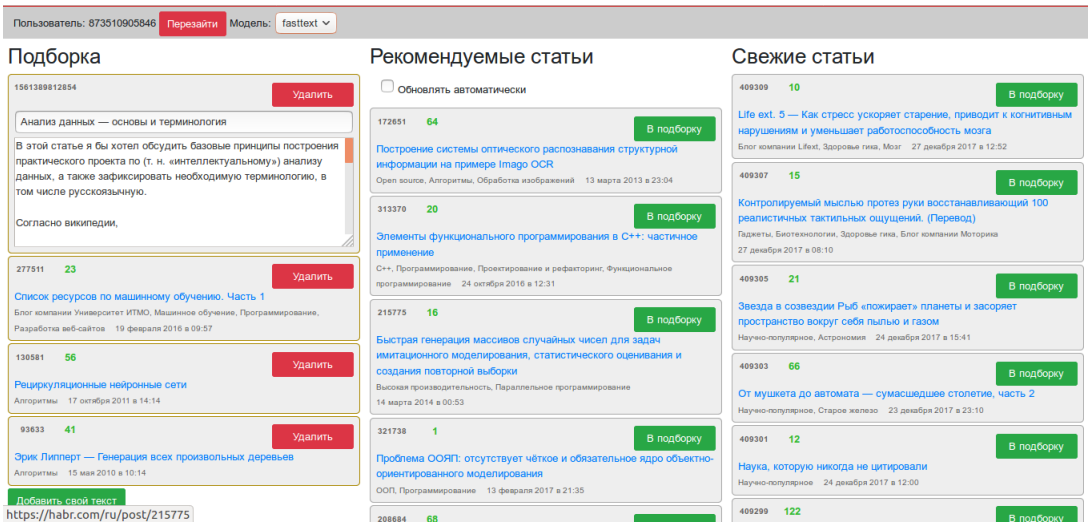


Рис. 9: Пример работы алгоритма Fasttext на третьей итерации поиска.

На 4 итерации алгоритм BM25 практически не сменил свою выдачу, TF-IDF стал рекомендовать больше про нейронные сети, Fasttext дает неточные рекомендации, но темы очень смежные с анализом данных, Topic частично повторяет предыдущие выдачи, но здесь можно заметить попытку охватить и тему нейронные сети, и обучение в анализе данных. Здесь стоит отметить, что на предыдущих итерациях время работы алгоритмов было несколько секунд и не особо заметным. Сейчас время работы алгоритма TF-IDF составило 10 минут, что значительно сказывается на впечатлении от выдачи, в то время как BM25 отработал за 40 секунд, Fasttext и Topic за пару

Запрос	TF-IDF	BM25	Fasttext	Topic
Анализ данных — основы и терминология, Список ресурсов по машинному обучению, Рециркуляционные нейронные сети	Краткий анализ решений в сфере СОВ, Самое главное о нейронных сетях, Логика сознания	Краткий курс машинного обучения, Открытый курс машинного обучения Метод Виолы-Джонса	Построение системы оптического распознавания, Элементы функционального программирования, Быстрая генерация массивов случайных чисел	Спектральный анализ сигналов, Список ресурсов по машинному обучению, Machine Learning. Курс от Яндекса

Таблица 10: ТОП3 выдачи агоритмов поиска по запросу из 4 статьей.

секунд. Поэтому возникает желание на следующих итерациях вообще не рассматривать работу TF-IDF, а набирать подборку только из Fasttext или Topic, которые могут дать более разнообразные варианты для дальнейшего чтения.

Как можно заметить, разведочные поиск по теме «Анализ данных» на данном этапе еще не закончен. Можно дальше и дальше продолжать получать рекомендации. Мы промоделировали итерационный процесс поиска, как пользователь может взаимодействовать с системой и какие может получать результаты.

6 Заключение

6.1 Итоги работы

В рамках проведенного исследования была достигнута поставленная цель и решены сформулированные в начале исследования задачи. Подведем итоги по проделанной работе:

1. Разработана система разведочного поиска и запущен прототип на текстовой коллекции новостного ресурса Хабр;

2. Разработаны алгоритмы поиска на основе векторных представлений для такого вида поиска;

3. Предложена методика оценки алгоритмов поиска, процесса и всей системы разведочного поиска;

4. Были проведены эксперименты и собраны ассесорские оценки для оценки качества работы алгоритмов поиска процесса и всей системы разведочного поиска;

5. В результате анализов эксериментов было выявлено, что каждый метод обладает своими преимуществами и недостатками. Наиболее полезным для ассесоров было использование чередование алгоритмов на каждой итерации поиска – для старта лучше себя показывают алгоритмы TF-IDF и BM25, в том время как на больших подборках и после нескольких итерация Fasttext и Topic работают гораздо быстрее и позволяют найти еще новые статьи с другими взглядами на тему;

6. Ассесоры отметили пользу сервиса, а также то, что он позволил им действительно получить новые знания. Рекомендательная выдача сервиса интереснее поисковой выдачи новостного ресурса Хабра, на данных которых проводился эксперимент. Однако, только лишь 40% ассесоров отдала бы предпочтение предложенному сервису в противовес обычным поисковикам. Это неплохой результат, который говорит о преспективе дальнейших исследований и необходимом продвижении в данном направлении.

6.2 Дальнейшие исследования

Среди возможных направлений продолжения исследования и усовершенствования предложенной системы разведочного поиска можно выделить такие:

1. Сделать возможность для пользователя делать несколько подборок, сформировнных в папке по тематикам. В рекомендательной ленте отображать, какая статья рекомендуется к какой теме. Это позволит пользовател. изучать несколько тематик одновременно. Кроме того, пользователь может включать или выключать некоторые темы из участвия в запросе на рекомендации.

2. Расширить пользовательский интерфейс для проведения разведочного поиска не только по текстам документов, но и по тематикам коллекции в целом. Добавить возможность посмотреть карту тем всей коллекции, углубиться в какую-то тематику, перейти в соседние темы.

3. Провести сравнение работ алгоритмов поиска по типу A/B тестирования – ассесор не будет знать, с какими моделями он сейчас работает, а в рекомендательной ленте будут перемешаны выходы разных моделей. Это позволит более точно оценить, выходу каких моделей ассесоры отдадут большее предпочтение.

4. Добавить оценку статей не только по релевантности, но и по порядку для чтения. Пользователи отмечали, что таких оценок не хватает и не понятно, с чего лучше начать изучение темы. Эту проблему можно было бы решить, если бы на всей коллекции была введена метрика порядка чтения, которая позволила бы рекомендовать пользователям статьи для их уровня знаний по теме.

5. Далее шаги 2 и 3 повторяются до тех пор, пока пользователю есть что добавлять из рекомендательной ленты и он чувствует необходимость в пополнении своих знаний.

6. Провести эксперименты на нескольких коллекциях документов, том числе и на мультиязычных.

Список литературы

[1] Additive regularization for topic modeling in sociological studies of user-generated texts / M. Apishev, S. Koltcov, O. Koltsova et al. // Mexican International Conference on Artificial Intelligence / Springer. — 2016. — Pp. 169–184.

[2] Bigartm: Open source library for regularized multimodal topic modeling of large collections / K. Vorontsov, O. Frei, M. Apishev et al. // International Conference on Analysis of Images, Social Networks and Texts / Springer. — 2015. — Pp. 370–381.

[3] *Chirkova N., Vorontsov K.* Additive regularization for hierarchical multimodal topic modeling // *Journal Machine Learning and Data Analysis.* — 2016. — Vol. 2, no. 2. — Pp. 187–200.

[4] *Diriye A. M.* Search interfaces for known-item and exploratory search tasks: Ph.D. thesis / University College London (University of London). — 2012.

[5] Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // *arXiv preprint arXiv:1301.3781.* — 2013.

[6] *Ellis D.* A behavioural approach to information retrieval system design // *Journal of documentation.* — 1989. — Vol. 45, no. 3. — Pp. 171–212.

[7] Enriching word vectors with subword information / P. Bojanowski, E. Grave, A. Joulin, T. Mikolov // *Transactions of the Association for Computational Linguistics.* — 2017. — Vol. 5. — Pp. 135–146.

[8] Exploratory search framework for web data sources / A. Bozzon, M. Brambilla, S. Ceri, D. Mazza // *The VLDB Journal—The International Journal on Very Large Data Bases.* — 2013. — Vol. 22, no. 5. — Pp. 641–663.

[9] Exploratory search through visual analysis of topic models / P. Jähnichen, P. Oesterling, G. Heyer et al. // *Digital Humanities Quarterly (special issue).* — 2015.

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

[11] Is exploratory search different? a comparison of information search behavior for exploratory and lookup tasks / K. Athukorala, D. Głowacka, G. Jacucci et al. // *Journal of the Association for Information Science and Technology*. — 2016. — Vol. 67, no. 11. — Pp. 2635–2651.

[12] *Karpathy A.* Arxiv sanity preserver. — `https://github.com/karpathy/arxiv-sanity-preserver`. — 2016.

[13] *Kelly D. et al.* Methods for evaluating interactive information retrieval systems with users // *Foundations and Trends® in Information Retrieval*. — 2009. — Vol. 3, no. 1–2. — Pp. 1–224.

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

[15] Mendeley: Creating communities of scholarly inquiry through research collaboration / H. Zaugg, R. E. West, I. Tateishi, D. L. Randall // *TechTrends*. — 2011. — Vol. 55, no. 1. — Pp. 32–36.

[16] *Mirizzi R., Di Noia T.* From exploratory search to web search and back // Proceedings of the 3rd workshop on Ph. D. students in information and knowledge management / ACM. — 2010. — Pp. 39–46.

[17] Reading tea leaves: How humans interpret topic models / J. Chang, S. Gerrish, C. Wang et al. // Advances in neural information processing systems. — 2009. — Pp. 288–296.

[18] A survey of definitions and models of exploratory search / E. Palagi, F. Gandon, A. Giboin, R. Troncy // Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics / ACM. — 2017. — Pp. 3–8.

[19] *Vorontsov K., Potapenko A.* Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization // International Conference on Analysis of Images, Social Networks and Texts / Springer. — 2014. — Pp. 29–46.

[20] *Vorontsov K., Potapenko A., Plavin A.* Additive regularization of topic models for topic selection and sparse factorization // International Symposium on Statistical Learning and Data Sciences / Springer. — 2015. — Pp. 193–202.

[21] *Webber F. D. S.* Semantic folding theory and its application in semantic fingerprinting // *arXiv preprint arXiv:1511.08855.* — 2015.

[22] *White R. W., Marchionini G., Muresan G.* Evaluating exploratory search systems // *Information Processing and Management.* — 2008. — Vol. 44, no. 2. — P. 433.