ВВЕДЕНИЕ

В жизни современного человека существует огромное множество организаций, на прием к которым обязательна предварительная запись либо же постановка клиента в очередь непосредственно на месте. Примером таких организаций служат МФЦ, государственные поликлиники и т.д., которые, конечно, имеют свои собственные системы онлайн очередей. К сожалению, подобные решения отсутствуют в случаях, когда нужно организовать собственную очередь здесь и сейчас, как например, при сдаче экзамена по практической части в ГИБДД.

Есть хорошее решение для этой ситуации — электронная очередь. Но классическая электронная очередь — это экран и принтер талонов, затраты на оборудование и плюс не понятно каким образом это можно вынести на улицу. Поэтому важное значение для сокращения времени ожидания имеет разработка универсальной системы онлайн очереди, которой в идеале могли бы пользоваться абсолютно любые организации и не только.

Актуальность темы исследования обусловлена тем, что несмотря на обилие методик и технологий, которые уже есть на данный момент, к сожалению, они не адоптированы под большинство ситуаций, где требуется СУО. Требуется поиск инновационных решений, которые окажут положительное воздействие на сокращение времени, которое человек проводит, находясь в очереди.

Цель научно-исследовательской работы – изучить методы и алгоритмы, которые будут применяться при разработке СУО.

Задачи работы:

- обосновать актуальность научно-исследовательской работы;
- представить методы решения выявленных недостатков для будущей реализации системы онлайн очереди.

Объектом исследования является организация онлайн очереди.

Предметом исследования является анализ технологий и методов, которые лежат в основе организации сервиса онлайн очереди.

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области

Всюду, где есть возможность образования очереди, может оказаться целесообразным применение систем управления очередями (СУО). Системы управления очередями, или электронная очередь — это взаимосвязанные модули, обеспечивающие управление очередностью предоставления услуг и потоком клиентов для создания комфортной атмосферы для персонала компании и ожидающих людей. Кроме того, СУО дают возможность получать различные статистические данные о работе персонала и оценивать эффективность его работы.

По оценкам Минэкономразвития России, сделанным в 2010 г., только по пяти наиболее массовым государственным услугам и только в очередях население теряет около 1925 млн руб. ежегодно, а потери ВВП составляют свыше 2620 млн руб. Подобное или еще худшее положение дел существует во многих странах мира. Например, взрослые жители крупных городов Великобритании тратят на очереди ежегодно по 67 часов. Стояние в дорожных пробках, ожидание у лифта и примерочных магазинов одежды занимают 41 минуту в месяц.

Для данного исследования была собрана статистика, в основе которой 52 человека ответили на вопрос «Ожидали ли вы свою очередь на сдачу экзамена более нескольких часов?». 71,2% ответили «да» и «нет» ответили 28,8% соответственно.

Как это решается на данный момент: один из сотрудников физически самостоятельно контролирует и оповещает очередь. Результатом такого подхода являются минус один сотрудник из команды; скопление неинформированных посетителей, которые на одном месте могут ждать своей очереди достаточно продолжительное время.

ГЛАВА 2. ПОДБОР АЛГОРИТМА ДЛЯ РЕАЛИЗАЦИИ

2.1 Подходы и методы для решения поставленной проблемы исследования

Для реализации именно универсальной очереди будут собраны векторные пространства, а далее поиск сходств, на которых построится базовая модель, а на расхождениях дообучаться и строиться специальные ранжирования под каждое учреждение. Для этого понадобится нейронная сеть.

Нейронная сеть — это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Это тип процесса машинного обучения, называемый глубоким обучением, который использует взаимосвязанные узлы или нейроны в слоистой структуре, напоминающей человеческий мозг.

Также нужно подобрать алгоритм ранжирования, так как нужно учесть, наличие льгот у лиц, находящихся на пенсии, ветеранам ВОВ, ветеранам труда, инвалидам 2 и 3 групп, чтобы минимизировать участие человека в работе системы. Таких людей нужно пропускать вне очереди, однако также необходимо предусмотреть поднятие людей в очереди на более высокие позиции, которые пропустили n количество людей перед собой.

В основе базового алгоритма ранжирования лежит система рейтинга. При регистрации посетителя в очереди ему присваивается определенный рейтинг (числовое значение), система сверяет рейтинг нового посетителя с рейтингом первого в очереди. Если рейтинги равны или рейтинг нового посетителя меньше, то новичок ставится в очередь после того, кто пришел раньше. Если начальный рейтинг нового посетителя больше, то система ставит его в начало очереди, при этом меняет его значение вычитая единицу, а рейтинг более раннего на эту же единицу растет. Такое распределение посетителей в очереди гарантирует справедливость в обслуживании посетителей: чем больше человек пропускает впереди себя, тем выше его

текущий рейтинг. А это значит, что, пропустив одного или двух, далее он не будет никого пропускать и вне зависимости от рейтинга нового посетителя пройдет раньше него.

Однако похожее обобщение не универсально. Такое ранжирование не имеет смысла, когда льготы в конкретном учреждении попросту не рассматриваются или же признаки, по которым нужно пропускать людей, являются уникальными. Поэтому планируется использовать несколько методик ранжирования, которые и составят итоговую очередь.

Ключевым параметром, который позволит перейти к умному ранжированию, является поле «торопится ли человек». Если он равен «нет», то вышеприведенный алгоритм отсортирует эту часть очереди и поместит ее в конец итоговой. Если же параметр, который определяет торопится ли человек, равен условному «да», то такие люди попадают в часть с глубоким ранжированием, которая также отсортирует их и поместит этот список в начало итоговой очереди.

Рассмотрим как это будет работать на примере очереди на сдачу экзамена по практической части в ГИБДД и очереди в больницу. Общими признаками у обеих очередей будут: пол; дата рождения; фамилия; id; пароль.

Для составления очереди в ГИБДД также понадобится информация о том, какая коробка передач – механика или автомат, это повлияет на ранжирование, так как пользователей, которые сдают на автомат, обычно ставят в начало из-за быстрого проведения экзамена.

В то время как для записи в больничное учреждение понадобится уже другая информация, например: льготный статус; вид гражданства; требуется ли срочная госпитализация и т.д.

Так вот для части с глубоким ранжированием понадобятся еще такие параметры, как: беременность (такого человека стоит поставить на прием первее); количество людей на иждивении и др.

Для части с глубоким ранжированием понадобится машинное обучение.

Ранжирование машинного обучения — это применение машинного обучения (обычно контролируемого, полуконтролируемого или обучения с подкреплением) при построении моделей ранжирования для информационно-поисковых систем. Данные обучения состоят из списков элементов с частичным порядком между элементами в каждом списке. Этот порядок обычно получается путем присвоения каждому пункту числовой или исходной оценки или бинарного суждения (например, «релевантный» или «нерелевантный»). Целью моделей Learning-To-Rank (LTR) является ранжирование набора элементов-кандидатов для любого заданного поискового запроса в соответствии с критерием предпочтения.

Ранжирование нужно везде, где система предоставляет пользователю выбор из большого числа вариантов:

- ранжирование выдачи поисковой системы;
- ранжирование рекомендаций пользователям (книги, фильмы, музыка, товары интернет-магазина, и т.п.);
- ранжирование вариантов автоматического завершения запроса (Query Auto Completion, auto-suggest);
- ранжирование возможных ответов в диалоговых системах (Question Answering Systems);
- ранжирование вариантов перевода в системах машинного перевода (Machine Translation).

Для того, чтобы все это работало понадобится также использовать утилиту Pipeline, которая помогает автоматизировать рабочие процессы машинного обучения. Конвейеры работают, позволяя объединить линейную последовательность преобразований данных, что завершается процессом моделирования, который можно оценить. Иными словами, эта утилита будет включать нейронную сеть для ранжирования, которая позволит ранжировать очереди с разными данными одинаково хорошо.

Существует огромное количество языков программирования для реализации самых разных задач. Рассмотрим лишь некоторые, а на основе

проведенного анализа выберем тот язык, на котором будет реализован будущий проект.

2.2.1 Подбор языка программирования для будущей реализации веб-сервиса

Для реализации веб сервиса, как уже рассматривалось ранее, понадобится поддержка языком: работы с одной из СУБД; работы с нейронными сетями.

Java – объектно-ориентированный язык программирования общего назначения.

Преимущества языка Java:

- универсальность. Многие современные системы и приложения разработаны с помощью языка Java, т.е. преимуществом является способность интегрировать методы науки о данных непосредственно в существующую кодовую базу;
- Java – это высокопроизводительный, скомпилированный язык общего назначения. Это делает его пригодным для написания эффективного производственного кода ETL (извлечение-преобразование-загрузка), а также алгоритмов машинного обучения с использованием вычислительных средств.

Недостатки:

- «многословность» языка Java делает его не лучшим вариантом для проведения специальных анализов и разработки более специализированных статистических приложений;
- Java не имеет большого количества библиотек для передовых статистических методов, что важно для данной работы.

MATLAB - пакет прикладных программ для решения задач технических вычислений.

Преимущества языка MATLAB:

- MATLAB, предназначенный для численных вычислений, хорошо подходит для использования количественного анализа со сложными математическими требованиями, такими как обработка сигналов, преобразования Фурье, матричная алгебра и обработка изображений;
- визуализация данных. MATLAB имеет ряд встроенных возможностей построения графиков и диаграмм.

Недостатком данного языка является специфичность. Это не язык программирования для общего назначения, ни о какой поддержке тех же СУБД, например, не может быть и речи.

Python – широко используемый язык программирования общего назначения.

Преимущества языка Python:

- он имеет обширный набор специально разработанных модулей и широко используется разработчиками. Многие онлайн-сервисы предоставляют API для Python. Это хорошо в рамках этой работы для выгрузки веб сервиса на облако;
- такие программные пакеты как pandas, scikit-learn и т.д., делают Python надежным вариантом для современных приложений в области машинного обучения;
- имеет встроенную СУБД SQLite, что также является преимуществом в контексте реализации веб приложения.

  По умолчанию стандартная библиотека Python уже содержит модуль sqlite3, никаких дополнительных действий программисту для работы делать не нужно;
- имеет удобные фреймворки для создания веб приложений.

  В частности, Flask - минималистичный каркас веб-приложений, предоставляющий лишь самые базовые возможности. Далее созданный макет программы уже можно будет выгрузить на сервер.

Недостатки:

- типобезопасность. Python – это динамически типизированный язык, т.е. часто могут возникать ошибки несоответствия типов.

По итогу сравнения выиграл язык программирования Python. Он является хорошим вариантом для целей науки о данных. Большая часть науки о данных сосредоточена вокруг процесса ETL. Эта особенность делает Python идеально подходящим языком программирования для таких целей. Разные библиотеки, разработанные специально для этого языка, делают Python очень интересным для работы в области машинного обучения.

2.3 Подбор алгоритма ранжирования людей в очереди

По поводу конкретной модели до данных говорить рано. Однако для того, чтобы учесть все нюансы будет реализован собственный алгоритм ранжирования пользователей, включающий подходы sota-ranking алгоритмов, таких как PiRank, VNS-Rank, SetRank и Mulberry.

Рассмотрим подробнее в чем заключается вышеприведенные подходы.

2.3.1 Что такое sota-ranking?

SOTA - это аббревиатура от State-Of-The-Art. В контексте искусственного интеллекта (ИИ) это относится к лучшим моделям, которые могут быть использованы для достижения результатов в задаче, специфичной для ИИ.

Основные преимущества:

1. Повышает точность задачи.

   Прежде всего, нужно проверить, какие параметры определяют модель SOTA, такие как полнота, точность и площадь под кривой. Можно выбрать любую метрику. Затем можно определить ее значение SOTA для каждой выбранной метрики. Если эти метрики имеют высокую точность производительности (около 90%-95%), они помечены как

SOTA. Эти модели очень точны. Так задачи ИИ максимально приближены к тому, что нужно делать пользователю;

2. Повышает надежность.

   Как упоминалось выше, высокая точность модели SOTA также повышает надежность задач ИИ. Если это задача машинного обучения или задача глубокой нейронной сети, то результаты в значительной степени такие, какими они должны быть. Они надежны и не считаются случайным тестом;

3. Обеспечивает воспроизводимость.

   Чтобы продукт искусственного интеллекта был гибким и экономичным, есть возможность быстро отправить минимально жизнеспособный продукт на тестирование клиентам. Затем можно перейти к получению отзывов пользователей и итеративному улучшению. Поэтому воспроизводимость в модели SOTA может считаться хорошей практикой. Это помогает с алгоритмическими компромиссами.

4. Сокращает время генерации.

   Поскольку модель SOTA помогает в воспроизводимости алгоритма или продукта, она также помогает сэкономить время, когда весь процесс «ложится» на конвейерную ленту. Это означает, что можно сделать продаваемый продукт из прототипа за меньшее время, чем когда был сделано тот же продукт с нуля. Все, что нужно, это воспроизвести алгоритм на параметрах, на которых он должен быть протестирован, уже есть, так что да, вы экономите много времени при генерации продукта.

Ключевой проблемой, связанной с подходами машинного обучения к ранжированию, является разрыв между интересующими показателями производительности и суррогатными функциями потерь, которые могут быть оптимизированы с помощью методов, основанных на градиенте. Этот пробел возникает из-за того, что показатели ранжирования обычно включают операцию сортировки, которая не поддается дифференциации по параметрам модели.

Предпочтение перед элементами задается с помощью меток релевантности для каждого кандидата. Фундаментальная трудность LTR заключается в том, что интересующие нижестоящие показатели, такие как нормализованный дисконтированный совокупный выигрыш (NDCG) и средняя позиция релевантности (ARP), зависят от рангов, индуцируемых моделью. Эти ранги не дифференцируемы по отношению к параметрам модели, поэтому показатели не могут быть оптимизированы непосредственно с помощью методов, основанных на градиенте.

Чтобы решить вышеуказанную проблему, популярный класс LTR использует методы сопоставления элементов с вещественными оценками, а затем определяет суррогатные функции потерь, которые работают непосредственно с этими оценками. Суррогатные функции потери, в свою очередь, могут принадлежать к одному из трех типов. Модели LTR, оптимизированные с помощью точечных суррогатов, представляют ранжирование как проблему регрессии/классификации, в которой метки элементов задаются их индивидуальными метками релевантности. Такие подходы напрямую не учитывают какие-либо взаимозависимости между ранжированием.

Также рассмотрим какие подходы лежат в основе методов ранжирования. Это listwise и Pairwise подходы.

listwise-подход, когда оценивается все ранжирование целиком. Такие подходы не могут эффективно учитывать “связи” из-за предварительной перестановки всего списка;

Pairwise-подход — это попытка задать функцию отношения на множестве объектов. То есть, модель получает на вход два объекта и должна выдать вероятность того, что первый из них больше подходит пользователю, чем второй. Вообще говоря, при таком подходе мы не получаем математически выверенной операции, могут быть накладки в виде невыполнения ассоциативности, транзитивности или других свойств — методы обучения Pairwise модели не всегда гарантируют их соблюдение.

Однако, как показано на рисунке 1, если существуют пары предпочтений элемента “A>B” и “C>D” для пользователя 1, пары “A>D” и “C>B” также должны существовать для пользователя 1 из-за двоичного значения неявной обратной связи. Другими словами, мы имеем p (A> D, C> B | A> B, C> D) = 1 в практическом процессе построения пары, что нарушает независимость между парами и, таким образом, влияет на результат оптимизации потери по парам.

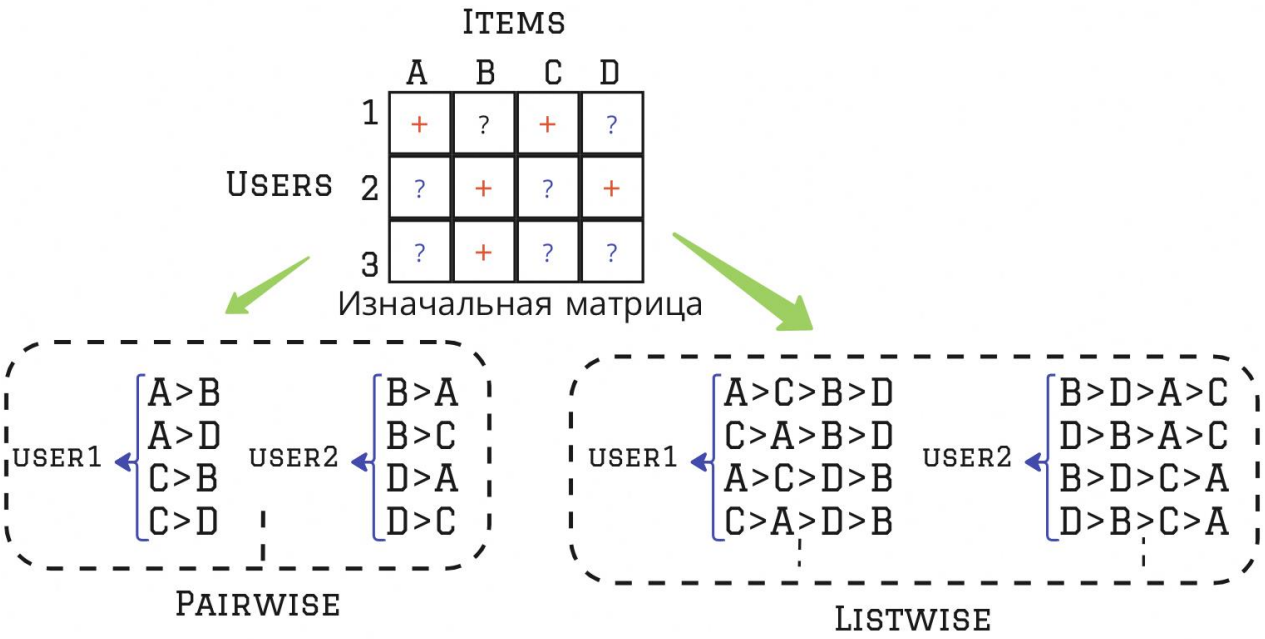

Рисунок 1 - схематический набросок структур предпочтений в различных подходах к совместному ранжированию, где ”>” представляет порядок предпочтений

Рассмотрев преимущества и недостатки, виды подходов, теперь рассмотрим, что лежит в основе конкретных алгоритмов.

PiRank – алгоритм масштабируемого обучения ранжированию с помощью дифференцируемой сортировки. Это новый класс дифференцируемых суррогатов для ранжирования, которые используют непрерывную, контролируемую температурой релаксацию к оператору сортировки на основе сортировки по нейронной сети (NeuralSort).

В основе PiRank лежит подход listwise, в котором оценки изучаются с помощью глубоких нейронных сетей, а суррогатные потери получаются с помощью дифференцируемой релаксации к оператору сортировки. В

частности, мы выбираем в качестве строительного блока контролируемую температурой релаксацию NeuralSort для сортировки и специализируем ее для часто используемых показателей ранжирования, таких как NDCG и ARP. Конечная цель обучения для PiRank сводится к точной оптимизации метрики ранжирования в пределе нулевой температуры и компенсирует смещение для снижения дисперсии в оценках градиента при высокой температуре. Кроме того, PiRank масштабируется до реальных промышленных сценариев, где размер списков товаров очень велик, но интересующие показатели ранжирования определяются лишь небольшим набором товаров с наивысшим рейтингом. Масштабирование обеспечивается новой стратегией "разделяй и властвуй", похожей на сортировку слиянием, где мы рекурсивно применяем расслабление сортировки к вложенным спискам меньшего размера и распространяем только верхние элементы из каждого вложенного списка для дальнейшей сортировки.

Variable Neighbourhood Search for Learning to Rank Problem – listwise подход. Переменный поиск соседства в 2 новых методологиях в ИИ для обучения рангу.

VNS используется для оптимизации решений эвристических задач (задач, вызывающих познавательную активность). Решения основаны на систематических изменениях соседних решений. Изменения вносятся во время фазы возрастания для получения локального оптимума и фазы возмущения для получения глобальных оптимальных решений. Процедуры разведки и эксплуатации выполняются с использованием различных размеров шага мутации. Цель состоит в том, чтобы выбрать лучшее потомство для перехода к следующему эволюционирующему поколению.

Алгоритм поиска по переменной окрестности является разновидностью алгоритма локального поиска (LS). Но это преодолевает проблему застревания в локальных оптимальных решениях, с которой сталкивается LS. Он используется для достижения цели поиска глобальных оптимальных решений. Это может быть достигнуто путем более тщательного изучения различных

решений окрестности с недетерминированными и нерегулярными размерами шагов в развивающихся итерациях. Чтобы найти локальный оптимум, для задач оптимизации использовались методы локального поиска. Они локально повторяют исходное решение, улучшая значение целевой функции каждый раз для получения локальных оптимальных решений. Это делается до тех пор, пока не будут сделаны дальнейшие улучшения после получения локальных оптимальных решений. Улучшенное решение x’ в окрестности N(x) текущего решения x может быть получено на каждой итерации. Примерами таких методов, которые были рассмотрены в других исследованиях для других проблемных областей, являются: генетические алгоритмы (GAs), эволюционные стратегии (ES), эволюционное программирование (EP) и эволюционные алгоритмы (EAs).

Более того, алгоритмы VNS ранее не использовались при решении задачи LTR.

SetRank – также listwise подход. Многомерная функция ранжирования перестановки, которая кодирует и ранжирует элементы с сетями самовнимания. По своей сути этот алгоритм учитывает характеристики неявной обратной связи в рекомендательной системе. В частности, SetRank нацелен на максимизацию апостериорной вероятности новых сравнений предпочтений по множеству и может быть реализован с помощью матричной факторизации и нейронных сетей.

Mulberry – это гибрид liswise и pairwise подходов. Изучает политики ранжирования, максимизирующие множество показателей по всему набору данных.

2.4 Подбор нейронной сети для реализации универсальной очереди

Для реализации потребуется алгоритмическая либо же нейронная preprocessing утилита, которая будет изменять данные для их подачи в нейронную сеть, с помощью которой получится составлять универсальные

очереди для разных задач. Это нужно, так как вручную подбирать параметры для матрицы входных данных под каждую задачу физически невозможно.

Для реализации была выбрана рекуррентная нейронная сеть.

Рекуррентные нейронные сети (Recurrent neural network) — вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки. RNN могут использовать свою внутреннюю память для обработки последовательностей произвольной длины.

ЗАКЛЮЧЕНИЕ

Инновационностью предлагаемого подхода является настраивание параметров среды для индивидуализации очереди под конкретное учреждение.

Универсальная очередь требует использования векторных пространств и нейронных сетей, чтобы научить компьютеры обрабатывать данные способом, аналогичным человеческому мозгу. Алгоритм ранжирования необходим для учета людей с особыми потребностями, а рейтинговая система используется для распределения посетителей в очереди. Для особых случаев, таких как в ГИБДД и больнице, где принимаются во внимание различные атрибуты, необходимы несколько методик ранжирования. Машинное обучение используется для создания моделей ранжирования поисковых запросов.

Итогами научно-исследовательской работы является решение поставленных задач, а именно: были рассмотрены алгоритмы, которые будут лежать в основе будущего веб сервиса, а также их теоретические аспекты.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ГИБДД – государственная инспекция безопасности дорожного движения

МФЦ – многофункциональный центр

СУБД – система управления базами данных

LTR – learning-to-rank

VNS – variable neighbourhood search

SOTA – state-of-the-art

СУО – система управления очередью

ИИ – искусственный интеллект

ETL – извлечение-преобразование-загрузка

ЭО – электронная очередь

LS – local searching