

## Семинар "Педагогика ИИ"

Дата и время: 10.12.2025, 10.00

Место проведения: НИУ «МЭИ», г. Москва, ул. Красноказарменная, д. 13с3, аудитория М-101.

Участники: Павел Юрьевич Анучин (НИУ «МЭИ»), Шамиль Алиевич Оцоков (НИУ «МЭИ»), Эдуард Артурович Челышев (НИУ «МЭИ»), Владимир Владимирович Чистяков (НИУ «МЭИ»), Роман Сергеевич Куликов (НИУ «МЭИ»), Мария Александровна Дурова (НИУ «МЭИ»), Иван Андреевич Акинфиев (НИУ «МЭИ»).

Повестка:

- 1) Тема основной дискуссии «Мультиагентные системы»:
  - Доклад «Мультиагентные технологии: от децентрализации к LLM и обратно»;
- 2) Выбор тематики следующего семинара.

Павел Юрьевич Анучин:

Коллеги, всем добрый день. Это 10-й юбилейный семинар педагогики ИИ. Он у нас регулярно проходит, является открытым для всех желающих из организаций. Сегодня у нас будет доклад на тему: «Мультиагентные технологии: от децентрализации к LLM и обратно». Докладчиком у нас выступает Иван Акинфиев, он сейчас представится и начнет свой доклад. А перед этим мы по уже сложившейся традиции представимся. Я думаю, что начнем мы с очных участников и продолжим участниками в дистанционном формате. Кратко предлагаю рассказать о себе, и чем вы занимаетесь в сфере искусственного интеллекта. Меня зовут Павел Юрьевич Анучин. Я ассистент кафедры радиотехнических систем в ИРЭ. Мы занимаемся разработкой прикладных систем на базе искусственного интеллекта для коммуникации проектов инженеров.

Владимир Владимирович Чистяков:

Здравствуйте, коллеги. Меня зовут Чистяков Владимир. Я являюсь сотрудником кафедры РТС МЭИ. Я занимаюсь внутренним развитием искусственного интеллекта как в технической, так и в научной индустрии.

Иван Андреевич Акинфиев:

Добрый день, коллеги. Меня зовут Иван Акинфиев. Я представляю СПбГУ. Я являюсь сотрудником кафедры Систем Управления и Информатики.

Эдуард Артурович Челышев:

Меня зовут Эдуард Челышев, ассистент кафедры вычислительных машин, систем и сетей. Интересуюсь теорией, практикой применения и разработкой искусственного интеллекта.

Шамиль Алиевич Оцоков:

Здравствуйте, коллеги. Шамиль Алиевич Оцоков – профессор кафедры ВМСС.

Мария Александровна Дурова:

Добрый день, Дурова Мария.

Павел Юрьевич Анучин:

Спасибо. Давайте тогда переходим к следующей части нашего семинара, это доклад.

Иван Андреевич Акинфиев:

Меня зовут Иван, еще раз представляюсь. Я сегодня буду рассказывать про мультиагентную технологию, мне очень понравился общий обзор того, как методы эволюционировали от начала самых алгоритмических до состояния распределенных методов, которые мы имеем сегодня. Я занимаюсь методами распределенной оптимизации, в том числе стабилизации, но в техническую часть тоже погрузимся, в том числе у нас будет простой эксперимент, который был поднят из степеней, на этом примере покажем, что это довольно актуальная история, в том числе и робототехники и системы управления контролем. История важная, в том числе и в робототехнике и то, как сейчас развиваются лингвистические модели, где термин «мультиагентная технология» используется не только как робототехника. Начнем мы с самых основ, почему эти мультиагентные технологии стали возможными с точки зрения теории и практики. Данный класс алгоритмов появляется, начинает развиваться активно в системе, в нейтральных программах, но появляется в системе 30-40-х годов. Что такое неравенство Нэша? Неравенство Нэша — это неравенство, которое показывает для нас, что каждый агент должен выбрать стратегию такую, чтобы не дать победить в игре определенной математической другим агентам или не дать им выигрышную стратегию. То есть он должен занять такое положение, что смена его для этого конкретного агента будет невыгодна по сравнению с остальными агентами. То есть, если, например, мы играем в камень, ножницы, бумагу, у нас это более сложная игра, чем для обычного игрока. Если у нас есть камень, а не бумага, мы пытаемся поменять. Самое главное, у нас есть очень хорошая стратегия, если мы играем в игру, что мы выбираем камень-ножницы-бумага, и в принципе это является стратегией, которая оптимальна, просто потому что никто не выигрывает, и значит система находится в равновесии. Но подобные сложные задачи уже относятся к неравенству Нэша, которое показывает как раз, что мы берем не сами это состояние, которое у нас в дискретном пространстве находится, например, если у нас есть сложные какие-то стратегии, мы выбираем математическое ожидание и говорим, что нахождение подобной стратегии по математическому ожиданию для нас наиболее выгодно. Это показано на втором неравенстве. Что это нам дает? Подобное определение очень хорошо позволяет говорить о том, как агенты между собой взаимодействуют, в первую очередь какие

состояния им наиболее выгодно выбирать. Впоследствии мы увидим, что в принципе большинство алгоритмов консенсуса — это либо эгоистические истории, как со следующим алгоритмом, который будет представлен, либо это попытка усреднить состояние между агентами при увеличении их числа. Давайте перейдем к следующему состоянию. В современной теории считается, что муравьи сами по себе отдельно могут выполнять какие-то минимальные задачи, но именно интеллект самого роя приобретается, когда их становится много. Впервые этот тезис алгоритмически был выставлен в 1992 году Марко Дориго в его диссертационной работе. Это представляет довольно важный момент. Переход парадигмы от централизации к парадигме децентрализации. Решение задач на уровне мультиагентных систем, систем роя, систем, где нет централизованной власти, потому что до этого в основном рассматривались генетические алгоритмы построения, то есть даже на самом деле муравьиные истории рассматривались только как то, что матка дает команду для остальных муравьев. Группа муравьев обладает коллективным интеллектом, который образуется внутри одного кластера и направлен на выполнение определенной задачи. Собственно, процесс принятия решения описывается такой вот сложной формулой, на самом деле здесь ничего такого прямо сложного нет, это просто выбор стратегий того, как муравей дальше будет идти. У муравья есть основная стратегия, которой он пользуется для того, чтобы проложить путь для других своих собратьев. И каждый раз, когда муравей проходит, пытается сделать какой-то выбор, при этом он оставляет за собой следы, которые помогут подцепиться следующему муравью, и так они, в принципе, могут выбирать или искать свои решения. В принципе, мы будем работать именно так. В задаче классического построения есть несколько городов, между которыми мы находимся. Давайте быстро пробежимся, если было непонятно. Сначала появляется такая вещь, как уравнение Нэша, которая описывает то, какие состояния выгодно приобретать каждому агенту в рое. Появляется такая история, как уравнение Нэша. Это 40-е, 50-е, 60-е годы. Оно позволяет показывать, какие положения выгодно занимать конкретным агентам. Это нам позволяет в децентрализованных сетях очень хорошо показывать то, что выбор состояния агентов не случаен, а происходит по какому-то определенному правилу. И что мы можем благодаря этому правилу, которое определено и которое доказано, что им выгодно занять определенное состояние, задавать каждому агенту через какое-то воздействие другие состояния или через свод каких-то определенных правил. То есть, например, в камень, ножницы, бумага, я тут с коллегами пример переводил, мы сейчас опять на троих раскинем. Самое удачное состояние, как бы ни казалось, это выбрать, что у каждого из нас камень, ножницы и бумага. Когда мы все втроем проигрываем, мы в принципе все довольны, потому что никто не победил.

Шамиль Алиевич Оцоков:

Можно в этой игре выгодно выбирать стратегию равновероятную. Если вы начинаете с того, что бумага выше у кого-то игрока, то сразу другой начинает выбирать ножницы. Поэтому равновероятная там выгода.

Иван Андреевич Акинфиев:

Да, это правда с точки зрения одного агента, что равновероятный выбор состояния — это, действительно, наиболее лучший сценарий, но в плане именно равновесия системы, когда все проигрывают и никто не выигрывает, эта система становится равновесной и, собственно, это происходит с каким-то определенным временем, а правила выбора для конкретного агента действительно это то, которое вы сказали, это равновесный выбор любой возможной, либо камня, либо ножниц, либо бумаги. И важно еще заметить, раз на этот момент сделали акцент, что подобная история уже описывается не просто неравенством Нэша, а неравенством Нэша, которое идет по математическому ожиданию, потому что стратегии в камень-ножницы-бумага все-таки сложные, а не дискретные. Так, к муравьям. Быстро еще раз про муравьиные алгоритмы. Почему они настолько интересны? Это выглядит как мультиагентная система для математиков и для биологов. Это интересно изучать с точки зрения взаимодействия муравьев. Потому что вроде бы это что-то очень близкое к тому, что может человек, и, наверное, самое близкое из природы, что может человек с точки зрения и социального взаимодействия, и уровня интеллекта конкретного отдельного индивидуума, как может показаться. Поэтому их начинают изучать, и самый первый большой довольно прорыв — это 92-й год. Марко Дориго в 1992 году защитил диссертацию на тему того, что он предлагает то, как муравьи могут делать выборы, и то, что эта система на самом деле является распределенной, а раньше подобные алгоритмы рассматривали только в контексте того, что есть матка, есть структура, иерархии определенные, и по этой иерархии спускаются команды вниз. И Марко Дориго предполагает, что на самом деле это уровень эмерджентного интеллекта, как только какая-то группа муравьев кластеризуется, она начинает выполнять какую-то свою отдельную задачу. Что приходит от парадигмы генетических алгоритмов, например, к парадигме децентрализованных алгоритмов. Давайте тогда изменим рассказ. Выбор правила в случае выбора какого-то конкретного муравья, он уже здесь случайный, потому что, если выбирается какое-то определенное число от 0 до 1, если это число больше порога, то муравей действует по жадному правилу, которое прописано сверху. Если оно меньше порога, то он выбирает какое-то случайное правило, например, вокруг поискать какую-нибудь новую территорию. В чем фишка конкретно муравьев, что они оставляют за собой следы, и по этим следам уже выбирают идти или не идти остальные муравьи. По сути, даже можно сказать, что своими следами они рисуют такую карту выборов для остальных муравьев, потому что какие-то муравьи пройдут по нормальному распределению больше в одном месте, какие-то пройдут меньше. И, выбор этих путей уже будет показан. Можно просто на какой-то картинке изобразить, например, то, как муравьи выбирали путь. Пути муравьи выбирают по формуле, которая показана выше. Здесь эта формула описывает классическую задачу муравьиного мультиагентного алгоритма. Это поиск пути из одного города в другой город, ну или из одного... муравейника в другой муравейник, и тут уже выбор стоит между муравьями в таком порядке, что они либо пытаются пойти по жадному пути, либо пытаются смотреть, насколько след в какой-то определенной местности был близок к жадному. Будь это условно муравей видит цель, не видит

препятствий, что хорошо. Собственно, подобные алгоритмы на самом деле до сих пор используются во многих математических постановках задач. Например, reinforcement learning начинал заходить на мультиагентной системе. Теперь к консенсусным алгоритмам или более уже сложным системам, а именно, на первый взгляд, на самом деле, более сложным системам, а именно к консенсусным алгоритмам и задачам распределенной оптимизации. В принципе, любой консенсусный алгоритм описывается, как поиск минимума или максимума распределенной функции на  $n$  агентов. То есть если у нас есть какая-то задача, мы хотим, чтобы у нас либо найти глобальный минимум для суммы функций каждого агента, это может быть, например, функция выгоды.

Шамиль Алиевич Оцоков:

Извиняюсь, немного непонятно определение. Консенсус — это другая, как я понимаю. Когда несколько есть агентов, каждый из них принимает решение, и путем голосования определяется истинный результат. Когда большинство «за», достигается консенсус. В общем, как в блокчейне, когда есть несколько майнеров. Там есть алгоритм консенсуса, когда они достигают решения, что принять или отклонить ту или иную заявку или решение. Используется алгоритм, называется византийский генерал, например, для консенсуса. Такой известный в теории, как в распределенной системе. А какая связь минимума-максимума, то есть нахождение минимума-максимума, непонятно, объясните, пожалуйста.

Иван Андреевич Акинфиев:

Задача поиска минимума или максимума — это задача постановки именно оптимизации, чтобы достичь наиболее выгодного состояния системы. Это не обязательно какой-то минимум или максимум с точки зрения функционала. То есть у вас есть какой-то функционал риска или функционал качества, который у вас определяет то, как система работает. И задача поиска минимума или максимума — это именно задача поиска максимума или минимума этого функционала в зависимости от вашей постановки задач. Если что, я расскажу об этом поподробнее на конкретном примере, поэтому можете сейчас задать, можете попозже задать.

Шамиль Алиевич Оцоков:

Ещё одно замечание. В теории игр, которую вы рассматриваете, есть, как я понимаю, неравенство Нэша. Там несколько иная постановка задачи. Дело в том, что то, что хорошо для одного игрока, не обязательно хорошо для всех. Ситуацию нужно рассматривать в целом.

Существуют различные типы игр. Вы начали с примера «камень-ножницы-бумага» — это статическая игра с определёнными стратегиями игроков. Также существует Парето-оптимизация, которая как раз ищет решение, выгодное для всех участников. А вот здесь у вас встречается слово «консенсус». Честно говоря, я не совсем понимаю, как оно связано с поиском минимума или максимума. Впрочем, ладно, давайте не будем сейчас на этом останавливаться.

Иван Андреевич Акинфиев:

Сейчас, отвечу тогда на вопрос. В случае, если мы говорим сейчас про теорию игр, с точки зрения одного агента действительно часто выгодно представлять, например, то, что с точки зрения одного агента, с точки зрения игры, мафии, выгодно мафии, чтобы она победила. Но, например, если рассматривать систему мафия, маньяк и мирный житель, где каждый представляет собой три стороны, то с точки зрения множества прогонов, с точки зрения именно системы и то, как игроки реагируют на какие-то действия других игроков, выгодно, например, чтобы победил мирный, потому что мафия или маньяк не будут в этом случае тогда в обиде, хотя для них, в принципе, проиграть невыгодно. Но с точки зрения системы усредненное состояние будет то, что мирный победил. С точки зрения минимума, почему это постановка задачи поиска консенсуса? Это поиск минимума или максимума определенного? Это, потому что в случае рассматривания системы мы рассматриваем какой-то функционал риска либо функционал качества. И в этом случае ставится постановка задачи найти минимум риска или максимум качества в этом случае. Поэтому это чуть попозже я расскажу, и давайте тогда продолжать. Если останется вопрос, готов обсудить с удовольствием. Зачем вообще современной истории нужны алгоритмы консенсуса или алгоритмы для мультиагентных систем, где они используются? Например, эта история была популярна в 15-20-х годах. Это децентрализованные операционные системы, это системы живого моделирования, что на самом деле очень важно, потому что алгоритмы консенсуса в системах живого моделирования позволяют делать такую штуку, как распределенное онлайн-моделирование, которое очень сильно, например, зависит от синхронизации времени. Дальше. Собственно, это сама синхронизация времени между различными системами, где требуется сохранять единицы времени, например. Или для децентрализованных систем с туманными вычислениями. Это различные робототехнические применения для обмена данными, для усреднения положений, помех, всего остального, что есть в приложениях робототехники, например, до сих пор. Все шоу-дроны — это не мультиагентные системы, это системы, которые изначально запланированы и летают по определенному маршруту, то есть в случае, например, потери у одного дрона ГНСС, то он просто упадет, он не будет обмениваться сообщениями с соседними дронами. И любые другие задачи, где требуется обмен данными между агентами и не хочется иметь какой-то централизованный сервер, тот же, например, задача блокчейна. Рассмотрим работу подобного алгоритма и то, как он устроится на проблеме синхронизации времени в робототехнической системе. В принципе, ее можно рассмотреть, как вопрос синхронизации консенсуса. Зададим граф, по которому будут общаться у нас агенты. Граф у нас может меняться во времени, связи могут рушиться, связи могут появляться. Топология в принципе довольно в этом плане вариативна в данной постановке задачи. Что мы хотим сделать в случае того, как мы синхронизируем время. У нас есть, например, 5 агентов, дальше это будет представлено, между которыми мы хотим засинхронизировать время. Например, нам нельзя использовать метку времени агента с наибольшей частотой. То есть в классических алгоритмах синхронизации времени используется агент с большей частотой, так как он имеет наименьшую ошибку, за его метку времени берется за то, что на самом деле сейчас происходит случай отсутствия соединения с каким-то глобальным сервером. Подобная постановка задачи, она, конечно, хорошая, она применяется, но у нее есть проблема, если мы хотим сделать какую-то онлайн-симуляцию, где нам нельзя терять метки времени при десинхронизации. Например, если один агент улетает вперед, потом он синхронизируется с агентом с большей частотой, оказывается, что его время идет вперед, он теряет метку времени, потому

что он принимает время, которое было уже позади него, метку времени, которую он уже прошел. В этом случае у нас есть другой путь, путь того, что мы можем модернизировать чуть-чуть внутреннюю секунду, каждый раз ее замедляя или ускоряя, чтобы к следующему моменту синхронизации внутренняя секунда пыталась синхронизировать ее со временем всей системы. То есть, например, вместо того, чтобы быть реальной одной секунды, до следующего такта синхронизации будет 1.05 от реальной секунды. И вот таким образом, такими небольшими скачками мы, первое, сохраняем метки времени до момента синхронизации, второе, мы имеем какой-то модификатор, который позволяет нам управлять внутренним временем. Это довольно простая задача синхронизации, но очень показательная. В этом случае мы ищем минимальное расхождение во времени между агентами на каждый момент синхронизации под предположением 1.4, это предположение на градиент, на дрейф функции и так далее, и на шум, и на сам граф. Собственно, тогда мы получаем между собой алгоритм, который мы видим справа, эта страшная формула запускается на каждом конкретном агенте, который позволяет сравнивать ему время со своими конкретными агентами сторонними и позволяет синхронизироваться по времени. В этом случае у нас стоит задача минимизации расхождения времени между агентами. То есть это не задача конкретного агента. Постановка задач идет именно на всю систему, когда нам нужно минимизировать расхождение. Это к вопросу о том, почему MinMax в данной постановке задачи. Здесь нужно пояснение тоже. Собственно, здесь идет симуляция между пятью агентами. Лайв-симуляция была. Первые 30 секунд у нас агенты рассинхронизируются, нет никакой синхронизации. Если видно, там плохой очень момент, конечно. Вот здесь вот видно, что черное идет расхождение. Дальше у нас есть контрольная группа из тех же пяти агентов. Если бы они продолжили без синхронизации, мы видим, что собственные времена у них расходятся. И есть группа, которая синхронизируется агентов, она сходится в одну полосочку, одну полосочку, и мы имеем полностью синхронизированную систему по одному агенту. Важный момент, если мы смотрим на агента номер 5 с наибольшей частотой, то благодаря коэффициентам  $p_1$ ,  $p_2$  в алгоритме мы можем синхронизировать, не усреднять по наиболее частому агенту, а по агенту, а просто средний именно по системе. Хотя, например, на агенте с наибольшей частотой у него будет, скорее всего, время поточнее. Но в случае, например, синхронизации и какой-то задачи нам может, например, больше интересовать неправильность времени агента, а именно усредненное по системе. Поэтому с помощью этих коэффициентов мы можем еще и выбирать по сравнению с обычными алгоритмами и не давать ошибки.

Шамиль Алиевич Оцоков:

А можно спросить, это ваш алгоритм или чужой?

Иван Андреевич Акинфиев:

Это мой алгоритм.

Шамиль Алиевич Оцоков:

Просто я хочу сказать, лет 15 назад я слушал один доклад у нас на кафедре по распределенным вычислениям, где упоминался алгоритм синхронизации времени распределенных вычислений. И там было название конкретное. То есть это уже давно, очень давно, я бы скажу, известно. Ничего нового там нет. Я не помню название, но вообще проблема решена.

Иван Андреевич Акинфиев:

Это разные постановки задач, разные подходы к решению. Смотрите, тут такой вопрос. Действительно, задача синхронизации во времени, можно сказать, что решена уже очень давно. Многие, на самом деле, задачи решены очень давно. Зачем, например, решать задачи, которые можно решить с помощью фильтра Калмана, с помощью нейронных сетей? Тут действительно есть такой вопрос, что какие-то задачи определенные, они могут быть решены, но при этом важно понимать, что даже если задача решена, посмотреть на нее с другой стороны тоже полезно, потому что это может открыть какие-то интересные особенности, то ли задачи, то ли применения, которые раньше не были. Например, если мы говорим про большие лингвистические модели, зачем, например, алгоритм компьютерного зрения заменять сейчас большими лингвистическими моделями, если алгоритмы компьютерного зрения уже работают. Но мы же все равно пытаемся применить LLM в этой среде.

Шамиль Алиевич Оцоков:

Я извиняюсь, просто вопрос. Если вы разрабатываете свой алгоритм, то нужно доказать, что этот алгоритм лучше по временным данным, по временному признаку, что у него сложность алгоритма, она ниже, чем сложность существующих алгоритмов. Поэтому он будет быстрее работать, чем существующий алгоритм. Или давать более точную оценку. Вот поэтому нужно сравнение. А так не имеет смысла разрабатывать. Если есть что-то, может быть, он лучше, чем ваш, понимаете.

Иван Андреевич Акинфиев:

Это пример задачи. Я не писал по этой теме диссертацию в плане того, что это было конкретное применение конкретного алгоритма с конкретной постановкой задачи. Мы просто посмотрели, как это работает и написали об этом статью. Не каждая статья должна быть какого-то прорывного уровня.

Шамиль Алиевич Оцоков:

Правило такое, что обычно перед тем, как разрабатывать какой-то алгоритм, нужно подумать, а он лучше или хуже, потому что может быть есть лучше вариант.

Иван Андреевич Акинфиев:

Даже если есть. Это не постановка задачи с точки зрения индустрии, это постановка задачи с точки зрения посмотреть, как это работает. Если каждый раз говорить о том, что наша постановка будет хуже или лучше, например, мы получили какие-то определенные результаты, и они могут быть хуже, они могут быть лучше. Но если мы не будем просто даже задаваться вопросом, нужны нам эти результаты, потому что задача решена или нет, то у нас просто наука исчезнет или остановится, потому что подобные вопросы часто приводят к интересным каким-то выводам.

Роман Сергеевич Куликов:

Без сравнения с имеющимся результатом, это результат ненаучный.



Иван Андреевич Акинфиев:

К вопросу о лингвистических моделях.

Если сравнивать с известными результатами, то, как писала в своей диссертации одна наша сотрудница, проводившая сравнение с классическими алгоритмами синхронизации, ключевое преимущество рассмотренного подхода в том, что в нём не требуется "ретопологизация времени".

То есть, в отличие от текущих алгоритмов (если говорить без привязки к конкретной статье), которые завязаны на использование времени агентов с наибольшей тактовой частотой, в случае онлайн-моделирования этот этап не нужен. Нет необходимости перестраивать всю временную шкалу событий (ретопологию точек времени) при синхронизации. Это и есть то научное преимущество, которое было продемонстрировано и защищено в той работе.

Что касается лингвистических моделей (LLM), то здесь в современной постановке задач происходит интересная трансформация термина "мультиагентные системы". Он постепенно уходит из сугубо технической сферы распределённых систем в область, где агент — это не отдельный компьютер, а определённая роль или специализированная, предобученная нейронная сеть, которую можно настраивать для улучшения общего результата модели.

Собственно, с появлением больших языковых моделей (LLM) термин "мультиагентный" несколько исказился. Он перестал означать исключительно децентрализацию вычислений в классическом понимании и стал чаще описывать систему из виртуальных агентов-ролей, выполняемых одним или разными экземплярами модели.

Однако сама идея децентрализации никуда не делась. Напротив, она начинает плавно проникать и в мир LLM, трансформируясь. Теперь речь идёт не только о множестве агентов-ролей в одной модели, но и о децентрализации самих агентов по разным вычислительным ресурсам, исходя из потребностей и доступных мощностей.

Эта задача становится особенно актуальной там, где нет возможности создать единый сверхмощный вычислительный кластер, сопоставимый, например, с западными или китайскими аналогами. В таких условиях встаёт вопрос: а что, если вместо обучения одной гигантской модели на одном кластере обучить множество небольших, узкоспециализированных агентов — каждый на своей задаче и своих данных? А затем организовать их взаимодействие для формирования конечного вывода.

Здесь нам на помощь приходят два направления:

Алгоритмы децентрализованного обучения. Они позволяют обучать LLM распределённо, без центрального сервера. Агенты (отдельные узлы) обмениваются между собой градиентами или весами, и в итоге глобальная модель приходит к общему консенсусу.

Направление специализированных моделей и "смеси экспертов" (Mixture of Experts, MoE). Это сейчас очень популярное направление, о котором, в частности, говорил директор центра ISCOLTECH. Идея в том, чтобы обучать отдельные, относительно небольшие модели для конкретных индустрий или задач (это и есть "эксперты"), а затем создавать механизм (микшер), который динамически выбирает и комбинирует их ответы.

В обеих этих парадигмах снова востребованы задачи оптимизации взаимодействия (инференса) между агентами. И здесь распределённые алгоритмы оптимизации и классические методы мультиагентных систем получают новый импульс для развития, так как требуют эффективной работы именно на распределённых, возможно, неоднородных устройствах.

Павел Юрьевич Анучин:

Коллеги, сейчас если есть вопросы, давайте сейчас тогда их зададим и, наверное, перейдем к третьему блоку нашего семинара, это дискуссия. Так понимаю, что у Ивана доклад подходит к концу, и он готов сейчас ответить на недостающие фрагменты, которые вызывают вопросы, и подискутировать по тем моментам, которые были изложены. Спасибо. Если вопросов нет, то я, наверное, задам первый. Иван, а можете рассказать о собственном опыте применения таких стеков, которых рассказали, и где они сейчас применяются, или, например, как вы это делали?

Иван Андреевич Акинфиев:

Да, насчет того же наболевшего алгоритма синхронизации. На самом деле под это была задача определенная. Задача была в синхронизации данных между двумя-тремя STM для онлайн-моделирования. И в этом алгоритм синхронизации сейчас применяется. Я в дальнейшей разработке именно физического воплощения не участвую. Но, в принципе, сам алгоритм в себе рабочий и имеет уже техническое воплощение. А в сравнении с другими аналогами, действительно, тут надо было попросить коллегу прислать свою диссертацию, где она это делала. Но могу сказать, что да, по сравнению с обычными методами, не происходит ретопологии и обратной backpropagation, которая требуется иногда в онлайн-системе.

Владимир Владимирович Чистяков:

У меня тоже есть вопрос. Иван, можете подробно рассказать о mixture of experts? Сталкивались ли с этим?

Иван Андреевич Акинфиев:

Насчёт Mixture of Experts и перехода от эвристик к оптимизации.

Вы правы, говоря о системе, где несколько предобученных агентов обсуждают между собой финальное решение. Часто на первом этапе такое обсуждение реализуется через эвристические системы (например, простые правила голосования, приоритеты, заданные вручную), а не через формальные алгоритмы консенсуса.

Однако в такой постановке действительно возникает задача: как перейти от этих эвристик к более формализованной и предсказуемой процедуре? Ведь когда происходит "обсуждение" между агентами (будь то отдельные модели или перемены ролей внутри одной системы), нужен чёткий механизм его завершения. Сейчас это часто делается внешним "костылём" — например, жёстким лимитом на количество шагов (токенов) или таймаутом, который просто "отрубает" дискуссию в какой-то момент.

Это неидеально. И здесь, как вы верно отметили, открывается путь к задаче Blackbox Optimization (оптимизации "чёрного ящика").

Что это значит на практике?

Формализация диалога как оптимизационной задачи. Вместо того чтобы произвольно обрывать обсуждение, можно попытаться смоделировать сам процесс диалога агентов как итеративную процедуру поиска оптимума некоего общего "функционала согласия". Каждый шаг обсуждения (обмен аргументами, токенами) — это шаг алгоритма, приближающий систему к консенсусу.

Гарантии сходимости. Главная цель такого перехода — получить теоретические гарантии сходимости. То есть, ввести некий синтетический (вычислимый) критерий,

который будет показывать, насколько дискуссия ещё полезна. Когда критерий показывает, что дальнейшее обсуждение не даст значимого улучшения (например, значение целевой функции стабилизировалось), процесс завершается автоматически. Это избавляет от необходимости задавать жёсткий лимит токенов "на глаз".

Эффективность и интерпретируемость. Такой подход позволяет понять, когда именно происходит самое важное. Вы привели отличный пример: возможно, для модели вроде Claude самые ценные выводы формируются в первые 20 тысяч токенов обсуждения, а остальные 60 тысяч вносят лишь 2% улучшения. Blackbox optimization могла бы выявить эту точку "убывающей отдачи" и остановиться там, значительно экономя вычислительные ресурсы.

Таким образом, готовый исследовательский путь выглядит так:

Исходное состояние: Система МоЕ с эвристическим, "костыльным" управлением диалогом (лимиты, таймауты).

Задача: Формализовать процесс обсуждения агентов как задачу совместной оптимизации (консенсусной или иной).

Цель: Разработать алгоритм/критерий, который гарантированно сходится за конечное (и предсказуемое) время к решению, близкому к консенсусу, и делает это оптимально с точки зрения затрат (токенов, шагов).

Результат: Более эффективные, предсказуемые и теоретически обоснованные мультиагентные системы для LLM, где мы не гадаем о необходимой длине диалога, а вычисляем её достаточность.

Это перспективное направление, которое лежит на стыке теории мультиагентных систем, оптимизации и Machine Learning.

Владимир Владимирович Чистяков:

Какие, если не секрет, вот Квен, да?

Иван Андреевич Акинфиев:

Квен, DeepSeek, что еще? Но в основном это Квен, потому что у нее есть доступ к инференсу, к функциям внутренним. То есть, например, GPT-модельки такого не позволяют.

Владимир Владимирович Чистяков:

Понятно, спасибо. Павел, есть еще вопросы у вас?

Павел Юрьевич Анучин:

Коллеги, будут еще вопросы в онлайн?

Мария Александровна Дурова:

По поводу, если делать сравнение с другими алгоритмами, чтобы повысить научную ценность. На каких задачах можно будет сравнивать эффективность алгоритма?

Иван Андреевич Акинфиев:

Касательно синхронизации: когда этот алгоритм проигрывает и как правильно ставить задачу. Вы правы. Представленный здесь алгоритм консенсусной синхронизации действительно может откровенно проигрывать в определённых сценариях по сравнению с "жадным" выбором времени от агента с наибольшей частотой (самыми точными часами). Когда жёсткий выбор лидера выигрывает? Это происходит в системах, где главный приоритет — абсолютная точность времени, а не

сохранение целостности временной шкалы событий. Например: В задачах, где нужно просто компенсировать дрейф и накопленную ошибку конкретного датчика или измерителя. В небольших системах (2-3 агента), где простота и гарантированное следование за самым точным источником перевешивают другие соображения. Когда с физической точки зрения важно именно качество самой "секунды" — её минимальное отклонение от эталона, а не согласованность внутреннего контекста системы. Здесь выбор времени от наиболее точного агента — это зачастую наиболее выигрышный вариант. Как правильно ставить задачу для научной работы (диссертация, статья)? Для научной строгости недостаточно показать работу на одном кейсе (например, на 5 агентах). Нужно системно исследовать масштабируемость: Пройтись по разному количеству агентов в сети: 1, 5, 10, 100, 1000. Это стандартная практика. Показать, что алгоритм хорошо работает и на больших масштабах. Консенсусные алгоритмы часто как раз и рассчитаны на это. Как вы верно заметили, для данного алгоритма это подтверждено другими статьями. Главное — чётко определить нишу, где ваш подход незаменим. В чём ключевое преимущество консенсусного подхода (без ретопологии)? Оно раскрывается в задачах, где нельзя терять или "перескакивать" метки времени, даже если они немного неточны. Важен сохранённый причинно-следственный контекст. Вы привели отличный пример — распределённый рендеринг видео. Задача: Изображение разбито на части, каждый агент рендерит свой сегмент. Проблема: Важно не столько абсолютное время ("сколько сейчас секунд по атомным часам"), сколько согласованность времени внутри системы. Чтобы все сегменты, отрендеренные для момента T, были собраны в цельный кадр, а не размазаны по разным моментам из-за рассинхронизации. Что важнее консенсуса: Сохранение целостности изображения во времени. Каждый "тик" внутренних часов системы должен быть одинаковым для всех агентов, чтобы сборка кадра была корректной. Допустима ли задержка? В такой постановке — да, критически важна внутренняя синхронность, а не абсолютная синхронизация с миром. Не страшно, если вся система в целом отстаёт или спешит на 5 секунд относительно глобального времени. Страшно, если один агент уже рендерит кадр 10, а другой — ещё кадр 5. Итог постановки задачи: Ваш алгоритм консенсуса — не универсальная замена классической синхронизации. Это специализированный инструмент для узкого, но важного класса задач, где: Требуется сохранять целостную временную шкалу событий (без ретопологии). Внутренняя согласованность важнее абсолютной точности. Система должна быть отказоустойчивой и не зависеть от единой точки отказа (агента-лидера с "самыми правильными" часами). Таким образом, для научной работы нужно честно показать: Где он проигрывает (малые системы, приоритет абсолютной точности). Где и почему он выигрывает (задачи, подобные распределённому рендерингу, онлайн-симуляции). Как он масштабируется (производительность на 10, 100, 1000 агентах). Это создаёт полную и убедительную картину.

Мария Александровна Дурова:

А агенты могут решать чуть-чуть разные задачи?

Иван Андреевич Акинфиев:

Да, но в случае алгоритма консенсуса да, есть более generalized version этого алгоритма, который позволяет именно task distribution делать, и там как раз задача в том, чтобы распределять задачи.

Мария Александровна Дурова:

Тогда встает отдельная задача к тому, чтобы как-то нормировать значения между собой. То есть если они решают разные задачи, то там будет разный выход.

Иван Андреевич Акинфиев:

Распределение тасков, простите за англицизм, чтобы задачи два раза не говорить. Распределение тасков, там общая задача системы – это минимизировать время, когда вы хотите, чтобы распределить загрузку между какими-то конкретными агентами, вы хотите, чтобы у вас каждый агент с каждым агентом примерно завершал задачу в одно и то же время, потому что это будет говорить о том, что у вас задача правильно выполнена. Там как раз будет задача ставиться глобальная, минимизировать расхождение времени выполнений между агентами. То есть в этом плане они будут выполнять разные задачи, но нам без разницы, на самом деле, что за задачи они выполняют, пока они их могут выполнять. В математической постановке они просто их могут выполнять, потому что там это задачи атомарные, они делимые. И, собственно, как только происходит это распределение, каждый агент занимается своим делом, но наша задача именно следить за тем, чтобы они были одинаково нагружены, как руководители в каком-то подразделении. Ну, то есть, чтобы условно у вас проект подходил к какому-то финалу с определенным временем.

Мария Александровна Дурова:

Получается, если вернуться к началу презентации, когда у нас возникло обсуждение вокруг консенсуса и как к консенсусу относится минимизация. Так вот, получается, минимизация — это поиск консенсуса не по значению, а по времени отклонения от среднего времени выполнения, грубо говоря.

Иван Андреевич Акинфиев:

Ну, в этом случае, да, это минимизация. Имеется в виду постановка задачи с точки зрения минимизации выполнения времени каждого агента, расхождения между каждым агентом в этот важный момент.

Мария Александровна Дурова:

Тогда немножко более понятно, как относится к консенсусу.

Иван Андреевич Акинфиев:

Рассмотрим задачу распределённой оптимизации с целью достижения консенсуса. Каждый агент в системе стремится минимизировать своё локальное расхождение — например, по времени выполнения задач или загрузке — с соседними агентами. Для этого он может передавать часть своих задач другим участникам сети. Если мой локальный «бэклог» (оценочное время до завершения) больше, чем у соседа, я передаю ему задачи, чтобы сравнить наши показатели. Глобальная цель всей системы

— прийти к усреднённому, сбалансированному состоянию, которое и является искомым консенсусом.

С математической точки зрения это формализуется через консенсусное усреднение: состояние каждого агента итеративно корректируется на основе состояний его соседей по графу взаимодействий. В устойчивом состоянии, при связном графе, все агенты сходятся к одному и тому же среднему значению. Этот подход обладает ключевыми преимуществами: он устойчив к динамическим изменениям в сети (агенты могут подключаться и отключаться), полностью децентрализован и хорошо масштабируется, так как требует только локального обмена данными.

Однако эта элегантная модель сталкивается с фундаментальной проблемой, которая часто опускается в теоретических построениях. Алгоритмы консенсусной минимизации идеально работают с бесконечно делимым ресурсом (например, вычислительным временем) или с однородными, атомарными задачами, которые можно свободно передавать между агентами.

Проблема возникает, когда задача является «беременной женщиной» — то есть обладает следующими свойствами:

Неделима (её нельзя разбить на части для распределения между агентами).

Уникальна (имеет специфические требования, выполнимые не каждым агентом).

Стохастична по времени выполнения (точная длительность неизвестна).

В такой ситуации базовый принцип консенсусной минимизации даёт сбой. Два агента не могут «усреднить» выполнение одной неделимой задачи. Агент, взявший её, будет иметь принципиально иную локальную метрику (например, время до завершения), чем его соседи. Попытки алгоритма выровнять состояние путём перераспределения окажутся тщетными, создавая стойкий дисбаланс. Система либо не сойдётся к истинному консенсусу, либо будет сходиться крайне медленно, постоянно «спотыкаясь» об этот дисбаланс.

Таким образом, для научной или инженерной работы необходимо чётко оговорить область применимости модели, указав, что она работает для задач делимых, взаимозаменяемых или эффективно разбиваемых на атомарные подзадачи.

Если же система должна работать с «беременными женщинами» — неделимыми задачами — требуется изменить постановку задачи. Возможные пути:

Сменить оптимизируемый функционал. Минимизировать расхождение не по текущему времени выполнения, а по ожидаемой будущей загрузке или по способности принять новую задачу.

Ввести иерархию или классы задач. Чётко разделить задачи на «делимые» (балансируемые консенсусным алгоритмом) и «неделимые» (управляемые отдельным диспетчером или специальными правилами очередей).

Разработать гибридную модель. Консенсусный алгоритм работает на верхнем уровне, распределяя поток потенциальных задач, в то время как выполнение конкретных неделимых задач регулируется иными механизмами, которые могут временно «исключать» занятого агента из процесса глобальной балансировки.

Это понимание — не недостаток консенсусных алгоритмов, а точное определение границ их применимости. Глубокий анализ таких ограничений, как проблема неделимой задачи, показывает зрелость подхода и превращает теоретическую модель в практический инструмент для построения надёжных распределённых систем.

Мария Александровна Дурова:

Ну да, 9 детей за 9 месяцев.

Иван Андреевич Акинфиев:

Да, да. Тут в этом плане есть такая история. Но в плане именно распределения это хороший пример того, как это работает.

Павел Юрьевич Анучин:

Хорошо, я вижу, что мы нашли в этом тоже логику, и это радует. Поэтому, в принципе, считаю, что на вопросы ответы получены. Коллеги, будут ли еще вопросы по тематике данного семинара? Если их нет, то предлагаю либо обсудить следующую тему семинара, который будет, если есть желающие предложить ее, или, например, мне анонсировать ее самостоятельно уже в формате чата. Да, если есть темы, интересующие вас, которые, например, по силе задач актуальных для вас конкретно сейчас стоят, мы можем поискать специалистов, которые могут на них подискутировать или рассказать о них.

Мария Александровна Дурова:

Еще всегда интересно по рекомендательным системам послушать, что случилось. Новое и актуальное. Там всегда идут обсуждения о том, как решать проблему холодного старта. И вроде как там тоже гибридные системы всюду применяются. Интересно было бы это послушать. Конкретно настолько заказывать я понимаю, что это нереалистично. Но вообще просто что по рекомендательным системам может.

Павел Юрьевич Анучин:

Можно, если это возможно, какую-то профилизацию, вы конкретно к чему бы, точку приложения, рекомендательные системы, в чем бы применялись?

Мария Александровна Дурова:

А там, по сути, это даже не принципиально. Или про мой?

Павел Юрьевич Анучин:

Ну, ваш кейс, да, просто для понимания, например, какой у вас интерес с точки зрения приложения.

Мария Александровна Дурова:

Так как я сейчас в основном работаю либо над собственными какими-то абстрактными проектами, либо со студентами, то вот как раз со студентами, когда делают проекты рекомендательных систем, там всегда такой чит-код, что изначально уже заполнены интересы пользователей. То есть там и всегда обозначено, что потенциальный рост проекта — это решение проблемы холодного старта, гибридации систем. Но в рамках бакалаврской, да даже и магистерской работы это особо не решается, потому что это все-таки тоже большую базу требует. Поэтому вот интересно было послушать, как это на самом деле должно решаться.

Павел Юрьевич Анучин:

Хорошо.

Мария Александровна Дурова:

Как раз неопределенная область не имеет особого значения, я думаю, то есть это может быть как онлайн кинотеатры, библиотеки, просто магазины, это не сильно важно.

Сейчас же еще это те же самые магазины добавляют функционалы распознавания образов, то есть рекомендация не только по истории предыдущей активности, но и по тому, что пользователь запрашивает. Он говорит, хочу чего-то такого, похожего на вот это, но конкретно вот этого, скорее всего, в магазине не будет, и поэтому становится задача подобрать что-то похожее.

Павел Юрьевич Анучин:

То есть это поиск признаков условно.

Мария Александровна Дурова:

Да.

Павел Юрьевич Анучин:

Ну и сюда еще, наверное, таргетинг.

Владимир Владимирович Чистяков:

Я помню, когда раньше в WhatsApp написал про кроссовки.

Мария Александровна Дурова:

Вот, вот. Это таргетинг. Ну, это же, в общем-то, в этих задачах и используется. Другое дело, что насколько это все четко работает. То есть, если это на уровне просто кроссовки и рекламы кроссовок, то это довольно как раз не очень высокий уровень. Чаще всего это рекламу просто раздражает. То есть задача в идеале — это так, чтобы тебе вот эта навязчивая реклама лезла, но ты не мог бы с этим справиться и все-таки переходил по этой ссылке. Вот это идеальное.

Павел Юрьевич Анучин:

Как у Яндекс.Браузера, где у тебя просто плашка рекламная уже почти в строке поиска.

Владимир Владимирович Чистяков:

Я просто отключила все. Ты вообще не отключай рекламу.

Павел Юрьевич Анучин:

Ты не можешь ее отключить даже. Владимир, а вот по, например, игре сможем, если что? Небольшой блок, это не полноценный доклад, мы не будем делать.

Владимир Владимирович Чистяков:

В целом сможем, но больше не в этом месяце.

Павел Юрьевич Анучин:

В этом месяце, наверное, да, коллеги, насчет следующего семинара сейчас пока не готов сказать, да, анонсирую в ближайшее время в чате опять же. По датам я имею в виду, потому что сейчас будут уже предновогодние дни, потом праздники. Мы пока думаем, как это будет. Да, возможно, это будет промежуточное мероприятие какое-то между следующим семинаром, который пройдет уже в следующем году. Поэтому ждем анонса в чате. А тут просто хотел узнать, что сможем ли мы, например, к следующему году, к следующему году, да, дать конкретику немножко по теме игры.



Владимир Владимирович Чистяков:

Мы сможем, только в том случае, если это до сих пор актуально.

Павел Юрьевич Анучин:

Коллеги, тогда на этом предлагаю завершать. Спасибо всем за участие, кто смог присоединиться в онлайн и прийти в очном формате. Очень ценим ваше время, уделенное нашему семинару. Поэтому, если вопросов больше нет, предлагаю как раз заканчивать и ждем вас на новых встречах. Спасибо.