Семинар "Педагогика ИИ"

Дата: 29.10.2025 10:00

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

Участники: Павел Юрьевич Анучин (НИУ «МЭИ»), Константин Александрович Петров (НИЦ "Курчатовский институт" - НИИСИ), Павел Романович Варшавский (НИУ «МЭИ»), Шамиль Алиевич Оцоков (НИУ «МЭИ»), Александр Павлович Еремеев (НИУ «МЭИ»), Данияр Альбекович Альжанов («Диасофт»).

Повестка:

1) Тема основной дискуссии «Проектирование ИИ-ассистентов инженеров»:
- Доклад «Обзор способов и инструментов создания обучающей выборки для проектирования ИИ-ассистента»;
2) Выбор тематики следующего семинара.

Павел Юрьевич Анучин:

Всем доброе утро! Меня зовут Павел Анучин, я ассистент кафедры радиотехнических систем МЭИ. Сегодня провожу семинар. В очном формате у нас сегодня «очень мало» людей, поэтому можно сказать, что сегодняшняя встреча пройдет в полностью дистанционном формате. Обсудим способы создания обучающих выборок для проектирования и работы ассистентов. Рассмотрим нашу практику и интересные примеры зарубежных коллег. Темы способов и инструментов создания выборок для обучения нейросетей на одном из наших семинаров уже касался мой коллега Владимир Чистяков, когда рассказывал о своей практике применения RAG моделей. Там уклон был больше про дообучение уже существующих моделей, какие результаты это может дать и при каких выборках это дает наилучший результат при меньших объемах самих выборок. Доклад будет в форме дискуссии: я зачитаю важные выдержки и предложу обсудить их. После этого выслушаем мнения коллег. Возможно, у нас получится интересное обсуждение различных практик создания выборок и датасетов.

Какую роль будет играть обучающая выборка при создании будущего ассистента? Например, ассистента, который специализируется на кожных заболеваниях. Я объясню позже, почему сделан такой выбор. Или, возможно, это будет ассистент по

планированию или скраму. Также можно рассмотреть ассистента по юридическим вопросам. В любом случае у него должен быть определенный профиль, которому он будет соответствовать как этически, так и с точки зрения предоставляемой информации.

Роль обучающей выборки и ассистента не ограничивается простой моделью. Это модель, которая объединяет данные, на которых она обучалась, и методы проверки, использованные при валидации. Важно, чтобы она вела себя именно так, как требуется для конкретной задачи. Например, если ассистент работает консультантом в банковской сфере, он не должен давать лишнюю информацию или галлюцинации. Процесс обучения должен быть построен таким образом, чтобы минимизировать возможность выдуманных ответов. Вместо этого ассистент должен предоставлять четко сформулированные ответы, основанные на базе знаний, соответствующей учреждению и его услугам, или на карточке пользователя. Таким образом, обучающая выборка — это не просто материал для обучения, а спецификация того, каким будет ассистент.

Для определения роли обучающей выборки сначала нужно понять, как будет использоваться ассистент. От этого зависят вопросы и задачи, а также ограничения и инструменты, которые мы будем применять для формирования его ответов. На основе этих сценариев мы составим набор эталонных запросов и ответов, так называемую Ground Truth. Желательно, чтобы это сделали эксперты, чтобы выборка максимально соответствовала нашему идеальному представлению об ответах. В Ground Truth мы также определим типы данных, которые будут использоваться: вопросы-ответы и последующие ответы. Если наш ассистент — это LLM, которая должна формулировать предложения под контекст с правильными ответами, то вопросы и ответы будут представлять собой текстовые предложения с определённым смыслом. Если же мы работаем с визуальной или графической информацией, например, с ассистентом по кожным заболеваниям, то ответы будут включать изображения и описания.

После постановки диагноза система должна сначала обработать изображение, которое ей предоставил пользователь. Это включает этапы детекции, сегментации и классификации. На каждом этапе система делает выводы. На первом этапе система определяет, обнаружено ли что-то потенциально опасное. Затем она сегментирует изображение, выделяя область, где может быть заболевание. Пока система не знает, что именно она нашла. Затем она обращается к своей базе данных (если обучение проводилось с её использованием) или использует ранее полученные знания, чтобы понять, что именно она увидела. На основе этого система принимает решение о том, какие рекомендации дать пользователю.

После классификации система может сообщить пользователю, что было обнаружено, поставить диагноз и объяснить, почему было сделано такое предположение. Это описание работы консультанта.

Скрам-мастер и планировщик работы группы должны опираться на конкретные данные из приложения-планировщика или списка сотрудников. Они не могут просто придумывать фамилии или задачи, а должны обращаться к четко определенным каталогам для получения информации. Их задача — давать краткие и ясные ответы, возможно, с комментариями, но без излишней сложности. На этапе Ground Show важно четко определить специфику и не включать все подряд, а сосредоточиться на том, что действительно необходимо. Ответы должны ссылаться на надежные источники, что можно поручить экспертам. Также стоит предусмотреть инструменты для проверки достоверности информации и предотвращения ошибок. Эталонные пары «вопрос-ответ» будут служить обучающими и эволюционными примерами. В результате мы получим две группы данных, которые помогут структурировать процесс и улучшить качество работы.

Для обучения и валидации мы используем два разных набора данных. Хотя они похожи по контексту и содержанию, это два отдельных набора. На одном мы обучаем модель, а на другом проверяем, насколько хорошо она работает в аналогичных условиях. Это позволяет нам оценить её эффективность. Метрики качества заранее определяются для каждого типа ассистента в зависимости от данных, с которыми он работает. Таким образом, у нас есть модель, данные для её обучения и способ проверки. Существует множество готовых решений для проверки моделей, особенно для сложных задач, таких как анализ рака. Эти решения используют LLM для оценки контекста и соответствия ответов. Они также проверяют, откуда были взяты данные и насколько они логичны. Кроме того, есть базовые общие метрики, которые применяются в различных задачах.

Общие типы задачи, то есть там здравый смысл, рассуждение, многозадачное рассуждение, инструкционное поведение, то есть соответствие инструкциям, которые в него были заложены. Ну и мультимодальные задачи, когда есть у нас несколько, опять же, сущностей помимо LLM, возможно, есть еще какие-то инструменты. Есть, например, какая-нибудь YOLA или UNED. который в параллель будет обрабатывать нам изображение. Ну, в принципе, перечень на слайде есть, мы не будем на нем задерживаться, потому что это общепонятные такие примеры. Говоря в общем о бенчмарке, это почти всегда пара вопрос-ответ, который был составлен экспертом. Не всегда это в текстовом формате. Если у нас, опять же, нейросеть заточена под графическую, это будет у нас картинка, на которой есть целевой объект, и он там не представлен, не обведен, не указан. А есть вторая картинка, это же, где он указан, целевой объект. Или, например, картинка, на которой нет целевого объекта, и, соответственно, вторая картинка идентична ей, где ничего и не указано. На таких картинках происходит обучение, ну и потом валидация в дальнейшем. Пример эталонного ответа на кейсе с планированием, что условно есть у нас какой-нибудь, общие типы задач включают здравый смысл, рассуждение, многозадачное рассуждение, выполнение инструкций и мультимодальные задачи. В последних задействованы несколько сущностей, помимо LLM, такие как YOLA или UNED, которые параллельно обрабатывают изображения. Этот список представлен на слайде, но мы

не будем на нем останавливаться, так как это общеизвестные примеры. Бенчмарк обычно представляет собой пару вопрос-ответ, составленную экспертом. Вопросы могут быть не только в текстовом формате. Например, для графической нейросети это может быть картинка с целевым объектом, который не обозначен, и вторая картинка, где объект отмечен. Или это может быть картинка без целевого объекта и идентичная ей вторая картинка. На таких примерах происходит обучение и дальнейшая валидация. Пример эталонного ответа на кейс с планированием: есть RAG или другой планировщик задач, где руководитель вносит правки, а коллеги отмечают свои задачи. Ассистент должен извлекать из планировщика конкретные пункты и предлагать их в порядке 1, 2, 3. Например, подготовить отчет, проверить корректность данных, назначить исполнителя, команду или дату. Ассистент также может сформулировать предписание, например, о необходимости совещания по проекту "Орбита", если проект не был указан в таблице, но есть в RAG, так как модель изначально была взята для этого проекта. Если ассистент будет писать много воды или пропускать данные из планировщика, это будет плохим знаком, что он плохо проходит бенчмарк или валидацию. Это одна из оценок его работы.

Говоря об оценках и метриках качества, можно выделить три основных класса, которые мы подробно рассмотрим. Эти классы являются формальными и могут быть оценены автоматически с помощью алгоритмов. К таким параметрам относятся security, F1, точность, соответствие и другие. Они измеряются на основе данных, на которых модель была обучена, или выборки, размеченной экспертом. Семантические оценки для ООМ оцениваются другой моделью, например, критиком. Он может определить, была ли мысль выражена правильно, соответствует ли температура ответа заданной и т.д. Критик выявляет логические несоответствия, ошибки и даже галлюцинации в ответах модели. Например, он может заметить, что модель не должна была давать комментарии, а должна была предоставить сухую выжимку информации. Также существуют человеческие оценки, которые проводятся экспертами по критериям пользы, ясности и этики. Без них, вероятно, сложно выявить некоторые аспекты качества ответов модели.

Эксперт должен обратить внимание на узкие аспекты, которые могут быть полезны или значимы для компании. Важно понять, как коллеги воспринимают его работу и насколько он удовлетворяет их потребности. Критерием пользы должно стать то, насколько эксперт понимает и решает задачи, а также ясность его действий. Например, если у нас нет ясности в каком-то аспекте, эксперт должен заранее подготовить метрики, которые помогут оценить его работу. Формальные метрики включают долю правильных ответов в тестовой выборке и точность, то есть долю верно найденных ответов среди всех найденных.

Допустим, человек должен назвать три правильных термина, но называет четыре или два. В этом случае точность его ответа можно оценить как 3 из 4, что составляет 0.86.

Семантические метрики нестандартны и удобны для работы с обученной LLM, которая анализирует контекст. Она оценивает, насколько ответ соответствует

вопросу, насколько разнообразны выбранные фрагменты и откуда они взяты: из базы знаний или придуманы ассистентом. Также учитывается, насколько полно ассистент использовал весь контекст вопроса, включая информацию из подключенных приложений. Комплексные человеческие метрики включают полезность, ясность ответов и согласованность отчетов при повторных запросах. При итеративной проверке можно оценить, насколько последовательно ассистент отвечает. Например, не пытается ли он что-то выдумать, когда его об этом не просили.

Согласованность ответов при повторных запросах означает, что если мы попросим найти информацию, которая находится рядом по контексту, то она будет соответствовать нашему запросу. Например, если мы попросим найти информацию о коллегах, то ассистент должен предоставить данные о них, а не перечислять всю выборку задачи заново. Также важно учитывать общую оценку работы ассистента. Это субъективная оценка, но она имеет смысл, если мы сначала выровняем веса по всем остальным критериям. Люди, которые будут использовать ассистента, лучше всего смогут оценить, насколько хорошо он отвечает на их вопросы. Наконец, важно правильно интерпретировать метрики в разных сценариях.

Мы соотнесли метрики с типами сценариев использования ассистентов. Например, для поискового ассистента, который не должен ничего придумывать, а только извлекать информацию из рак-модели, важны ключевые метрики контекста. Важно, правильно ли он опирается на контекст или начинает выдумывать. Диалоговый ассистент должен генерировать мысли в правильном контексте, а не просто извлекать их откуда-то. Учебный ассистент и агент обращаются к приложениям или базам знаний, чтобы найти нужную информацию и ответить на вопрос пользователя. Ground Truth — это эталонная запись, которая содержит идеальные ответы для конкретной конфигурации. Ассистент учится сверяться с этими ответами и сравниваться с экспертным уровнем. Эксперт закладывает эти ответы, чтобы проверить, как ассистент справляется с задачами.

Переходя к следующему вопросу, стоит задуматься о том, что мы можем сделать с полученной информацией. Понятно, что размер выборки должен быть значительным, чтобы все работало эффективно. Если мы говорим о графической и текстовой информации, то их объемы будут существенно различаться. Однако, согласно классическим подходам, для обоих типов данных потребуется большое количество размеченных датасетов. Это включает в себя как вопросы и ответы, так и выделение объектов на изображениях с последующей классификацией. Все эти процессы требуют значительных трудозатрат. Нужно не только собрать данные, но и разметить их, а затем обучить модель. В связи с этим я нашел несколько интересных подходов. Один из них предложен Google, а другой — исследовательским университетом. Первый подход заключается в использовании только высокоточных меток. Это означает, что эксперт оценивает выборку данных и выделяет однозначные варианты, где решение задачи очевидно.

На слайде представлена картинка с синими и оранжевыми точками. В правой части они сближаются, и при их вводе могут пересекаться. Точки пересечения окружностей особенно интересны для обучения, так как они неоднозначны. Гипотеза заключается в том, что если обучить нейросеть на таких неоднозначных примерах, детально показав, как с ними работать, то она сможет решать и более простые задачи. Google подтвердил эффективность этого подхода. Другой интересный метод — абсолютный ноль. Проверялась гипотеза, что существующую модель можно дообучить, если она будет сама генерировать задачи и ответы к ним.

По сути, будет работать как две нейросети. Одна из них, возможно, разной конфигурации, будет придумывать задачи. Важно, чтобы эти задачи не были слишком сложными для решения и не настолько простыми, чтобы каждая последующая задача решалась без усилий. Постановщик должен находить баланс, чтобы задачи были средней сложности. Это приведет к тому, что каждая следующая задача будет чуть сложнее предыдущей, и он не будет зацикливаться на одном уровне сложности, а всегда будет стремиться к усложнению. У этого подхода есть свои плюсы и минусы. Плюс в том, что не нужно искать огромный датасет, размечать его, формулировать и определять ценности. Однако минус в том, что, поскольку задачи ставит нейросеть, мы не можем влиять на то, какие ценности она будет учитывать при создании новых задач. Возможно, это можно будет корректировать по ходу работы.

Смысл работы нейросети в том, что она будет сама придумывать себе новые задачи и пробовать их решать. В процессе выполнения этих операций она будет вырабатывать свои ценности и пути. Всё начинается с того, что мы даём старт, поставив простую задачу из того сектора, для которого планируем использовать нейросеть.

Например, если мы хотим, чтобы она решала математические задачи, то сначала даём ей простой пример, а затем усложняем его. В ходе эксперимента коллеги достигли интересных результатов в различных областях. Они создали графический демонстратор, где точки перемещаются внутри условной фигуры с ограничивающими стенками. Кружки не могут выходить за пределы этих стенок. Для наглядности они добавили физику тел. В эксперименте использовались несколько нейронных сетей. На примере этого демонстратора они обучили с нуля свой кодер, дообучили модели 4O, КВН и получили разные результаты. Нейросети, изначально обученные на большом контексте и не имеющие прямого отношения к этой задаче, справились с ней хорошо. Нейросеть, обученная с нуля и достигшая такого результата, показала стабильный результат. Это подчеркивает важность выбора модели в зависимости от конкретной задачи.

Для нашей задачи подойдет нейросеть, которая будет полностью сосредоточена на ней и лишена лишних блоков, способных вызвать больше ошибок, чем пользы. Я прикреплю ссылки на источники к презентации, которую выложу. Можно будет ознакомиться. Могу также продублировать их в чате, у них есть GitHub. Все эти примеры можно будет посмотреть и изучить. У нас есть доступ. Хотел бы поделиться опытом применения подходов на ассистентах, работающих с графической

информацией. Например, у нас есть ассистент по кожным заболеваниям. Он выглядит примерно так. Есть некий эталон, который пользователь передает нам на вход. На этом примере мы видим, как работает наша система. Пользователь задал вопрос: «Всё ли у меня в порядке?» Его изображение сначала поступает в модель U-Net, обученную на задачах детекции и сегментации. U-Net определяет наличие признаков, на которых его обучали, и выделяет интересующую область, предположительно связанную с заболеванием. Слева синим цветом показана маска, наложенная на исходное изображение. Это результат сегментации U-Net, который он выделил как область, вызывающую подозрения. Затем активируется модель YOLA, обученная накладывать другую маску. Она также выполняет детекцию выделенной области. На основе значений обеих масок U-Net строит рамку вокруг подозрительной области и накладывает центральную маску. Это этап верификации, где проверяется корректность работы U-Net. В завершение система выделяет внутренние структуры выделенной области.

Этот метод работы ассистента помогает определить асимметрию, которая может быть скрыта за маской. С помощью LLM мы можем сформулировать три заключения: есть что-то, нет чего-то, или есть атипичная сеть. По маске можно понять наличие асимметрии или синих тел. В данном случае синих тел нет, и ассистенту это известно. LLM формулирует мысль, что в этом месте есть типичная сеть с симметрией или другими признаками. Затем она обращается к базе данных, где уже есть таблица с признаками и диагнозами. На основе пересечения признаков ассистент рекомендует или не рекомендует диагноз, объясняя причины.

На стыке U-Net, YOLO и LLM с RAG проявляются признаки, которые соответствуют первоначальному запросу пользователя. Запрос формулируется текстом, а результат сопровождается картинкой. Эти признаки складываются в конечный итог. Возьмем другой пример. Здесь нет асимметрии. Еще один показательный пример — когда мы говорим о строительстве. У нас есть план этажа, и нам нужно найти на нем стены, лифты, лестницы и другие элементы. Я делал коллегам инструкцию для этого. Говоря об инструментах, есть приложение с открытым софтом. Его можно использовать на компьютере или как веб-решение на серверах компании, предлагающей подписку. Этот софт позволяет вставлять графическую информацию, задавать количество классов и гибко размечать их на изображении.

Шамиль Алиевич Оцоков:

Павел, можно задать вопрос по поводу задачи? У меня был студент, который занимался анализом изображений участков кожи для определения вида кожного заболевания. Здесь не нужно искать, где именно находятся пятна. Это задача классификации, а не обнаружения объекта. Для этого не требуется LLM, так как по пятнам можно однозначно определить заболевание. Почему вы выбрали такую задачу? И зачем нужно обнаруживать объекты?

Павел Юрьевич Анучин:

Спасибо за ваш вопрос. Я думаю, что задачи были разными. Первая задача была выполнена для компании, занимающейся разработкой метаизделий. Они хотели, чтобы информация с эндоскопов и других приборов, позволяющих снимать в макрорежиме, обрабатывалась с помощью нейросети и передавалась пользователю или врачу в виде сводки. Мы подошли к этой задаче комплексно и разбили её на части. Вторая задача заключалась в создании чат-бота для работы с клиентской базой. Пользователи могли задавать вопросы в свободной форме, отправляя текстовые описания, картинки с текстовыми описаниями или только картинки. Для решения этой задачи была использована LLM с RAC-моделью. Модель могла анализировать признаки, указанные пользователем, и делать предположения о том, что изображено на картинке, даже если эти признаки не были явно видны.
Если текстовым запросом задали вопрос, модель на основе данных и знания может ответить. Задача комплексная: мы должны что-то делать в любом случае, независимо от запроса. Если запрос не даёт данных для однозначного ответа, мы не даём ответ. Со студентом я сейчас активно работаю. Это демо для будущего проекта. Задание интересное, но свободных рук немного. Если появятся новые возможности, будем рады. Возвращаясь к созданию датасета для графических задач, можно использовать этот инструмент. Он не единственный, но подход у всех аналогов одинаковый. Инструмент предоставляет интерфейс, где можно выбрать классы и назначить объекты. Эти объекты будут размечены вручную или с помощью полуавтоматических сервисов.

Вот пример: мы выбираем класс и фигуру для разметки. Это может быть многоугольник, прямоугольник или овал, в зависимости от удобства. Затем указываем количество точек для разметки. Если это прямоугольник, нужно определить, сколько точек будет задействовано. Например, две точки будут обозначать ширину и высоту, а четыре точки создадут полигон. Я рекомендовал использовать четыре точки, потому что в логе будут записаны все четыре координаты объекта. Если это многоугольник, его кортеж будет состоять из множества координат. При объединении всех объектов в один датасет не возникнет проблем из-за различий в записи простых геометрических фигур по двум точкам и многоугольников, которые имеют другой принцип записи. Кортеж из координат каждой точки угла многоугольника обеспечит согласованность данных.
Возьмем, к примеру, сложный объект, состоящий из двери и стены с нишей. Эта конструкция несимметрична и не очень красива. Здесь много точек, образующих полигон. Внутри этого полигона находится простой объект — стена, а в ней — дверь. Дверь выделена другим классом и четырьмя точками, она находится внутри стены. Это особенность разметки. При дальнейшей работе с этим объектом важно решить, нужно ли включать дверь внутрь стены или же заканчивать стенку в начале и конце двери. Это важный момент, поскольку от него зависит, как будут выглядеть маски, создаваемые YOLA или UNED. Что касается метрик, то мы могли бы разметить 10 или даже 100 изображений своими силами, но это трудоемко и сложно, если у нас нет большого коллектива, который может этим заниматься. Рассмотрим пример: на картинке видны окна, стены, лестницы и лифты.

Мы будем отмечать фурнитуру не по отдельности, а одним классом. Все это нужно разметить по разным классам, чтобы использовать для обучения. Для этого мы используем инструмент, называемый аугментациями. Он позволяет брать уже размеченные изображения и изменять их разными способами. Мы можем искажать их по горизонтали и вертикали, менять цвета, затемнять или осветлять.

Например, из небольшой выборки в 30 изображений мы можем получить около 300 с помощью аугментаций. Важно следить за тем, чтобы искажения не повторялись слишком часто, иначе это может негативно сказаться на обучении. Мы создаем примерно 30-50 изображений с искажениями на одно исходное. Это позволяет значительно увеличить количество данных для обучения без дополнительных затрат времени и усилий.

По результатам работы мы получаем ключевой показатель — матрицу. На примере объекта можно выделить классы, которые задавались и находились. По вертикали представлены классы, которые есть в выборке. Последний класс, background, повторяется. Предпоследний класс — это класс, а последний background — это не класс, а фон. Всё остальное, что не вошло как класс в изначальную разметку, попало в фон. По горизонтали показано, что искала нейросеть и с какой успешностью. В данном случае это был U-Net. На пересечениях видно, что стена со стеной совпадала в большинстве случаев.

Когда мы говорим о пересечениях с другими классами, мы имеем в виду фон — подложку объекта. Это то, что окружает объект и не является его частью. Например, полка — это объект, а всё остальное вокруг неё — фон. Мы выделили класс "ground" как основной, а всё остальное отнесли к фону. Эксперименты показали, что для достижения высокой точности на конкретный класс лучше использовать одну нейросеть, которая будет обрабатывать все классы на картинке. Это поможет минимизировать неизвестные моменты, с которыми нейросеть может спутать объекты. Такой подход эффективнее, чем распределение классов по разным моделям. В итоге у нас будет один класс и две сущности: сам класс и его отсутствие. Например, если на картинке есть стена, то она будет классом, а всё остальное — фоном. В таком случае точность будет максимальной при хорошей выборке данных.
На примере этой матрицы мы рассмотрим виды ошибок. Здесь можно увидеть, как разделить ошибки первого и второго рода. Это могут быть ложноположительные и ложноотрицательные результаты, а также случаи, когда объект есть, но система его не обнаружила. Последняя ошибка особенно критична, потому что после фильтрации, если система что-то пропустила, исправить это уже невозможно. Если же система обнаружила что-то ошибочно, это можно исправить на этапе доработки. Например, она может принять окна за двери. Но если система вообще ничего не обнаружила, особенно в случаях с короткими заболеваниями, которые не всегда очевидны, то исправить ситуацию будет сложно. Поэтому лучше, чтобы система хотя бы указала на наличие объекта, а эксперт уже решит, стоит ли его учитывать.
Он также может утверждать, что что-то является правильным, а что-то — нет. Это

будет ложноположительный результат, как с дверями. Это еще один пример, который я привел. Кроме того, существуют и другие метрики, которые я описывал в презентации. Здесь мы видим классы и процент соответствия. Когда мы тестировали модель на выборке, не участвовавшей в обучении, можно было определить, какой класс работает лучше, а какой — менее точно. Также есть графики, которые я не показал. Они демонстрируют прогресс модели в зависимости от эпох обучения. Если обучающая выборка построена правильно, то на первых этапах наблюдается резкий прирост, затем наступает период, когда прогресс замедляется. Это не плато, а просто медленное увеличение. В какой-то момент можно заметить, что прирост либо замедляется, либо, наоборот, ускоряется. Обычно прогресс идет равномерно, и на определенном этапе прирост начинает снижаться.

В обратную сторону обучение происходит редко и не всегда, в зависимости от структуры учебного материала. Поэтому по этой шкале мы можем понять, что, скажем, 100 эпох будет достаточно, а 500 — избыточно. Мы уже получим приемлемый результат на 150 эпохе, и дальнейшее обучение не принесет пользы без явных изменений в обучающей выборке. На этом я завершаю первую часть доклада. Я хотел показать презентацию и предоставлю ссылки на источники после семинара. Если кому-то нужно, могу отправить их в частном порядке. Есть ли вопросы или предложения для обсуждения? Может быть, вернемся к какому-то моменту и рассмотрим его подробнее или обсудим?

Константин Александрович Петров:

У меня есть несколько вопросов. С самого начала говорилось о необходимости исключить галлюцинации в системе. Но я не совсем понял, как это достигается. Насколько я понимаю, есть фиксированный выбор, который система обучается использовать. Это, например, второй и третий слайды презентации.

Александр Павлович Еремеев:

Я насколько понимаю, что принцип... По сути, галлюцинация.

Константин Александрович Петров:

Просто галлюцинация контролируемая. Вот.

Александр Павлович Еремеев:

Конечно. Павел, можно показать третий слайд?

Александр Павлович Еремеев:

Различные метрики, которые он может применить. Если он найдет их удачно, то примет правильное решение. Если неудачно — это приведет к ошибке. Нейронная сеть работает по такому же принципу, поскольку теоретически доказано, что она всегда обладает определенной степенью неустойчивости. Чем сложнее сеть, тем выше

эта степень. Что это значит? Мы всегда обучаем и тестируем сети на конечных выборках. Например, при работе с кожными заболеваниями на вход сети может поступить вектор из бесконечного множества параметров. Какие-то из них могут измениться, наложиться друг на друга, и сеть будет пытаться найти ответ. Первые версии сетей, такие как GPT-2, могли ответить «не знаю», если у них не было данных для ответа. Однако современные большие сети фактически не используют такой ответ. Они пытаются сгенерировать ответ, даже если он неверный. Галлюцинации или неверные ответы могут выявить только серьезный специалист или сеть, называемая критиком, то есть сеть более высокого уровня. Когда мы говорим о моделях, которые хотим создать, стоит обратить внимание на последние доклады на конференции в Киеве, представленные профессорами.

Анна Безельтер, профессор, и другие специалисты обсуждают переход к узконаправленному искусственному интеллекту (narrow AI). Если мы решаем задачу, связанную с обучением или диагностикой, например, поведения турбины или её функционирования, то нужно обращаться к системе, агенту или модулю, который максимально соответствует этому направлению. Эксперт конкретной предметной области может добавить свои знания, которые наиболее проработаны и корректны.

Когда мы используем общий инструментарий, это аналогично работе врача общей практики. Он может определить заболевание и направить пациента к узкому специалисту. Это дилемма широкого специалиста, который знает обо всём, но не углубляется в детали, и узкого специалиста, который знает всё о конкретной области, но не видит общей картины.

Поэтому, когда мы говорим о серьёзных вещах, включая построение обучающих моделей, как упоминал Шамиль Алиевич, лучше использовать инструментарий, который специально разработан для данного класса задач или конкретной предметной области.

Тогда и эффект этот галлюцинации будет менее выявлен раз. И во-вторых, контекст сети, с которой мы работаем, он будет заточен на предметную область, ту, которая нужна разработчику. Вот то, что я хотел добавить. То есть галлюцинация – это не то, что сеть сознательно врет, хотя… Сами руководители AI Open говорят, что на отдельных задачах вот эта общая сеть GPT может давать до 60% неправильных ответов. Почему? Ну, потому что она не может, так сказать, использовать все свои метрики, то есть контекст, по-видимому, недостаточно глубок для вот такой узкой задачи, система начинает генерировать то, что она считает правильным по своим метрикам и дает ответы. По сути, для специалиста неверный ответ, ну или недостаточно точный. Поэтому вот я еще раз говорю, что вот сейчас идет все-таки переход, если нужна действительно помощь специалиста в узких специализированных задачах на уровне профессионала, то надо использовать инструментарий, именно заточенный под эту предметную область. Ну вот мой такой вот большой комментарий.

Константин Александрович Петров:

Нет, я считаю, что ответа на вопрос я не получил. Речь идет о снижении количества галлюцинаций. А в начале презентации речь шла о гарантии отсутствия этих галлюцинаций. То есть получается, что об отсутствии галлюцинаций мы все-таки говорить не будем. Мы будем говорить о их максимальном снижении, правильно?

Павел Юрьевич Анучин:

Если мы говорим про задачу, где у нас есть планировщик задач и нам не надо от нейросети ничего, кроме того, чтобы она парсила оттуда данные и выдавала их в отформатированном виде, то здесь, наверное, мы как раз будем говорить о том, чтобы она вообще не галлюцинировала и не придумывала ничего сверх того, что нам надо получить из этой базы, помимо форматирования ее в структурированный вид. Если мы говорим в принципе об ассистентах и тех задачах, которые сегодня обсуждались, то здесь стоит задача снизить. Как было сказано, мы снижаем те эффекты, которые получаются у больших языковых моделей, когда сложно отличить что она по делу сказала, а что нет.

Константин Александрович Петров:

Если мы ограничиваем её применение только парсингом данных, то зачем нам нейросеть? В моей практике в НИИ несколько раз случалось, что после анализа задачи становилось ясно: её проще решить без нейросети, используя более простые алгоритмы. Это могут быть стандартные библиотеки языков программирования, такие как Python, или алгоритмы оптимизации на основе генетики, которые проще, чем нейросети. Возникает вопрос: нужна ли нейросеть, если её задача так ограничена?

Павел Юрьевич Анучин:

Если задача действительно ограничена и решается предложенными методами, то нейросеть не нужна. Например, в кейсе с планировщиком возникает специфическая проблема. Она заключается в том, что сотрудники, которые используют этот планировщик для получения задач и информации, могут записывать туда плохо сформулированные мысли или неформализованные данные. Если нейросеть будет выдавать на выходе такие же плохо структурированные данные, это создаст проблемы. Такой инструмент будет сложно использовать, что негативно скажется на восприятии и работе с ним.

Константин Александрович Петров:

Спасибо, Павел. Теперь ответ понятен полностью. У меня еще был вопрос. Можно шестой слайд показать? Да, метрики оценки качества. Здесь речь идет про формальные, это, видимо, какие-то простые методы. Семантические, опять-таки, какая-то обученная модель, ну и человеческие, понятно, обученный человек. Вот вопрос к семантическим. А, собственно, на чем в этом случае обучается модель-

критик? На каком массиве данных, чем он отличается от того, на чем обучается нейросеть, выполняющая задачу?

Павел Юрьевич Анучин:

Здесь может быть поставлена задача, в которой одна модель использует RAG, а другая — нет. Например, при обучении берется модель, которая будет использовать раг и выполнять роль критика. Затем мы обучаем другую модель, которая в будущем сможет работать в той же парадигме с примерно такой же точностью, но уже без RAG. Таким образом, задача формулируется следующим образом.

Константин Александрович Петров:

Хорошо, спасибо, я понял. Для меня еще вопрос был такой, не конкретному слайду, а более общим. В общем, говорилось о нескольких применениях, да, это и ассистент, как бы там, руководителя, это и поиск на картинках, либо схемы домов, либо кожные заболевания. Вопрос подачи выборки в нейросеть в случае, если мы говорим о приложении ассистента-руководителя. То есть у нас есть некая система управления проектами. И нам нужно будет из этой системы управления проектами по какому-то API выкачать и обучить нейросети? То есть у нас еще стоит задача некой прослойки между API и нейросетью? Или как это выглядит в принципе? Вот то, с чем вы сталкивались с точки зрения ассистентов, как это выглядело?

Павел Юрьевич Анучин:

API и промежуточный слой, через который данные из базы поступают в нейросеть для использования, — это не нейронный инструмент, а отдельный модуль. Он отправляет запросы и получает ответы. На стороне нейросети находится графовая сеть. Она структурирует информацию, объединяя данные по темам. Когда поступает запрос, графовая сеть выделяет релевантные блоки и отбрасывает нерелевантные. Это не дает стопроцентного результата, но позволяет передать LLM не весь объемный документ, а только его выжимку, соответствующую контексту. LLM будет работать с этим материалом, определяя, что важно, а что нет.

Допустим, у нас есть база данных, которую можно редактировать. Она находится отдельно, и мы подключаемся к ней через канал. В базе хранятся диагнозы и метрики для их постановки. Мы можем добавлять или изменять данные.

Когда нейросеть обрабатывает запрос, она обращается к этой базе. Даже если мы изменили данные, нейросеть использует актуальную информацию. Это удобно, потому что не нужно переобучать нейросеть каждый раз при изменении данных. Мы можем гибко обновлять базу, и нейросеть будет использовать новые данные по мере необходимости.

Павел Романович Варшавский:

Коллеги, у меня больше не столько вопрос, сколько комментарий в дополнение к тому, что вы уже спрашивали. Когда мы говорим об обучающих выборках, их роль становится ключевой для работы ассистентов и нейросетей. Как уже отметили, у ИИ могут возникать галлюцинации и ошибки. Однако гарантировать безошибочную работу ассистентов можно только в узких предметных областях, где данные статичны и не меняются. В таких случаях можно обойтись без нейросетей и создать достоверную систему вопрос-ответ. Но если мы допускаем возможность ошибок или гипотез, то обучающая выборка должна быть не только репрезентативной, но и полной. Для серьёзных задач это может быть проблематично, как отметил Александр Павлович. Кроме того, в наборах данных часто встречаются противоречивые данные, что также нужно учитывать. Коллеги, хочу добавить к нашему обсуждению. Здесь уже можно говорить о медицинской тематике. На консилиумах часто сталкиваются с противоположными точками зрения на одну и ту же проблему. Важно понимать, какой уровень ассистента будет использоваться. Он будет предлагать решения или аргументировать выбор в той или иной ситуации? Также стоит учитывать, на кого ориентирован ассистент. В любой предметной области есть пользователи с разным уровнем знаний. Например, студент, который только начинает обучение, и эксперт, который уже глубоко разбирается в теме. Для первого ассистента может предлагать простые и понятные решения, а для второго — более сложные и детализированные. Эксперт может быть более снисходительным к ошибкам ассистента, если видит, что в ответе есть полезная информация. В то же время, он может ожидать более точного и детального анализа от ассистента.

Действительно, можно столкнуться с непростой задачей, которая для эксперта может быть неоднозначной. Подготовка обучающих выборок — это тоже большая проблема. Нужно уделить внимание их разметке, обеспечению отсутствия шумов, полноты и непротиворечивости. Эти аспекты важны во многих предметных областях. Их необходимо учитывать.

И последний комментарий. Кажется, что большие языковые модели, которые показали хорошие результаты в общих задачах, могут допускать ошибки. Сейчас их пытаются адаптировать для использования в конкретных областях.

Практика показывает, что если добавить к нейросети жесткие требования и не допускать ошибок, то можно получить статическую систему с однозначными и достоверными результатами. В таком случае возникает вопрос: нужен ли ассистент на базе нейросети? Это тема для обсуждения, но то, о чем мы говорили сегодня, напоминает именно это.

Александр Павлович Еремеев:

Спасибо. Мне кажется, что Константин прав: не всегда нужно использовать нейронные сети. Если задача узкая, есть хорошие эксперты и накопленная информация, то можно обойтись системами на основе продукционных правил, прецедентов и других методов. Плюс таких систем в том, что они имеют объяснительные компоненты. По

продукционным правилам мы строим дерево, которое показывает ход рассуждений. В случае нейронных сетей это проблематично. Поэтому сейчас активно разрабатываются доверенные сети, которые включают объяснительные компоненты или продукционные правила. Также создаются каскадные сети, где первый каскад определяет класс нарушения или патологии, второй каскад классифицирует эти патологии, третий каскад выявляет конкретные патологии и так далее. Это делается для того, чтобы понять и интерпретировать решение сети. Например, если нейронная сеть сообщает, что вы проходите или не проходите, то непонятно, почему было принято такое решение.

Она не объясняет. Поэтому если решение с простыми моделями и объяснительной компонентой действительно подходит, его нужно использовать, а не применять сложную многокаскадную сеть, которая теоретически неустойчива.

На основе теоретических моделей, например, теории Клумогорова-Тихонова, обучение всегда ограничено, а выборки на входе из бесконечного множества. То есть система никогда не дает стопроцентной надежности. Продукционные модели, напротив, более прозрачны и дают это, как уже обсуждалось.

Действительно, обучающая выборка очень важна для нейронных сетей. Режим аугментации и зашумления, подготовка к более качественной выборке, преобразования, которые занимают до двух лет, направлены на удаление второстепенной информации, заставляющей сеть переобучаться. Однако это не всегда необходимо. Важно использовать инструменты, подходящие для вашей предметной области и задачи. И не всегда это именно вот эти мощные нейронно-сетевые универсальные инструментарии. Но это вот как дополнение, к тому о чем мы дискутировали.

Павел Юрьевич Анучин:

Спасибо. Да, это действительно так. На первых семинарах, Павел Романович, вы уже говорили об этом. Было ясно, что при наличии хорошей президентской базы работать с ней комфортно. Всё объяснимо, всё доверено, и ничего неожиданного не произойдёт без причины.

Есть ли предложения по теме следующего доклада? Или, возможно, вы знаете, кто мог бы выступить? Также можно предложить тему для доклада на следующий семинар. У меня есть несколько идей, но сначала хотелось бы услышать ваши мысли. Мы параллельно проводим семинары с Олегом Вадимовичем, поэтому важно учитывать все аспекты.

Константин Александрович Петров:

Мы рассматриваем возможность разработки ассистента и уже начали прорабатывать техническое задание. В процессе поиска решений мы обнаружили множество

коллективов, которые работали над аналогичными проектами, как на теоретическом, так и на практическом уровне.

Мы планируем провести семинар в НИИСИ через две недели. На семинаре выступят докладчики из разных регионов, возможно, из Петербурга или Петрозаводска. Обсудим с ректором возможность их участия. Также рассматриваем вариант проведения внешнего семинара под эгидой семинаров по педагогике ИИ.

Однако может возникнуть проблема с публикацией полных версий презентаций. Есть ещё одна идея — лекция для студентов о применении искусственного интеллекта в обучении, науке и разработке микроэлектроники.

Я тоже постепенно готовлю лекцию, думаю, что через 2-3 недели она будет готова в каком-то виде. Поскольку я беру на себя много задач при разработке такой лекции, я бы хотел сначала представить ее тезисно на семинарах по педагогике ИИ. Это позволит получить критику от коллег.

Павел Юрьевич Анучин:

Приветствуем оба. Спасибо. По-первому, про работу я в курсе, обсудим возможность проведения мероприятия под эгидой нашего семинара. Все вопросы, касающиеся записи и не записи, решаемы. Решим, что запишем, а что нет. Это организационные моменты. Послушаем коллег из отрасли, которые уже достигли результатов и готовы поделиться опытом. Это всегда интересно и ценно для нас. Я передам Олегу Вадимовичу эту информацию для дальнейшего обсуждения. По второму вопросу - будем рады вас заслушать и покритиковать в хорошем ключе, подискутировать. Спасибо вам за инициативу. Думаю, что ближе к следующему семинару мы подготовим новый анонс, в котором то, что было сказано, уже пропишем как более актуальную информацию, что у нас будет на следующем семинаре, кто доложится. На этом семинар завершаем.

Константин Александрович Петров:

Коллеги, спасибо за интересный семинар! Я уже побегу на следующее совещание.