Санкт-Петербургский государственный университет

**ФИРСАНОВА Виктория Игоревна**

**Выпускная квалификационная работа**

**Подходы к построению диалоговой системы о расстройствах аутистического спектра с помощью архитектуры Трансформер**

Уровень образования: магистратура

Направление 45.04.02 «Лингвистика»

Основная образовательная программа ВМ.5805. «Компьютерная и прикладная лингвистика»

Научный руководитель:

доцент, Кафедра математической лингвистики,

Митренина Ольга Владимировна

Рецензент:

доцент, ФГАОУВО «Национальный исследовательский университет «Высшая школа экономики»», старший научный сотрудник, ФГБОУВО «Воронежский государственный педагогический университет»

Паничева Полина Вадимовна

Санкт-Петербург

2022

Оглавление

[Введение 2](#_Toc103705868)

[Глава 1. Калейдоскоп диалоговых систем 16](#_Toc103705869)

[1.1. Развитие диалоговых систем 16](#_Toc103705870)

[1.2. Современные модели и трансферное обучение 21](#_Toc103705871)

[1.3. Базовые модули диалоговых систем 31](#_Toc103705872)

[1.4. Архитектура Transformer 38](#_Toc103705873)

[Глава 2. Материал исследования 40](#_Toc103705874)

[2.1. Источник материала 41](#_Toc103705875)

[2.2. Структура набора данных 43](#_Toc103705876)

[2.3. Краудсорсинг 46](#_Toc103705877)

[Глава 3. Подходы к построению системы 48](#_Toc103705878)

[3.1. Работа с моделью 49](#_Toc103705879)

[3.2. Работа с данными 57](#_Toc103705880)

[3.3. Оценка системы 61](#_Toc103705881)

[3.3.1. Автоматические метрики оценки 63](#_Toc103705882)

[3.3.2. Ручная оценка 68](#_Toc103705883)

[3.4. Влияние лингвистических особенностей материала 70](#_Toc103705884)

[Заключение 74](#_Toc103705885)

[Список источников 84](#_Toc103705886)

[Приложения 91](#_Toc103705887)

[А. Образец набора данных 91](#_Toc103705888)

[Б. Тексты заданий для краудсорсинга 91](#_Toc103705889)

[В. Образец анкеты для ручной оценки 93](#_Toc103705892)

[Г. Методологические рекомендации 93](#_Toc103705893)

# Введение

В 2019 году на Всемирном экономическом форуме (ВЭФ) была представлена характеристика так называемых «Маяков», предприятий-«пионеров» четвертой промышленной революции [Всемирный экономический форум, 2019, с. 7-8]. Четвертая промышленная революция, так же известная как «Индустрия 4.0», предполагает внедрение и использование «умных» технологий, которые, на мой взгляд, способствуют интеграции физического и цифрового миров. К таким технологиям можно отнести виртуальную и дополненную реальность, аналитику больших данных, искусственный интеллект и другие разработки.

Сегодня, в 2020-е годы, мы наблюдаем за тем, как технологии «Индустрии 4.0» покидают научные лаборатории и начинают служить людям. Например, в коммерческой сфере популярностью пользуется виртуальная и дополненная реальность: проектирование дизайна интерьера в VR-очках (VR — англ., *virtual reality*, «виртуальная реальность») и примерка обуви с помощью видеокамеры смартфона не покажутся активным пользователям «умных» приложений чудом. Презентация товара или услуги в игровой форме, яркий интерфейс, интерактивность превратили использование виртуальной и дополненной реальности в эффективный инструмент маркетинга.

Технологии искусственного интеллекта (ИИ), такие как машинное обучение (англ., *machine* *learning*) и формирование рассуждений (англ., *machine reasoning*), применяются в коммерческих и некоммерческих организациях, муниципальных и государственных учреждениях. В России инструменты ИИ внедряются в системы информирования граждан, обрабатывают заявки на услуги, фильтруют фальшивые новости. «Умные» системы научились давать ответы на часто задаваемые вопросы, собирать обратную связь от клиентов и пользователей и упрощать некоторые процедуры (например, запись к врачу). Нередко работу таких систем обеспечивают алгоритмы обработки естественного языка.

Обработка естественного языка — это область искусственного интеллекта на стыке компьютерных технологий и лингвистики, направленная на создание инструментов для автоматического распознавания и генерации текстов и речи. Рассмотрим несколько успешных примеров подобных систем, разработанных в России.

Виртуальный ассистент «Маруся» [Маруся — голосовой помощник от VK, 2019], выпущенный компанией VK в 2019 году, позволяет пользователю вызвать такси, заказать доставку продуктов питания на дом, произвести поиск в интернете и многое другое с помощью простого голосового запроса. Количество навыков «Маруси» не ограничено: пользователи вправе самостоятельно расширять базу так называемых «скиллов» виртуального ассистента, а владельцы бизнесов могут интегрировать систему в собственные продукты [Маруся | Быстрый старт | VK для разработчиков, 2022]. На данный момент компания VK объединяет множество социальных сетей, приложений и сервисов. Таким образом, «Маруся» представляет своего рода распределительный центр с дружелюбным голосовым интерфейсом для пользователей.

Технологии распознавания и генерации речи также вошли в основу виртуальных ассистентов компании СберДевайсы — дочернего предприятия Сбера [СберДевайсы — СберБанк, 2019; Виртуальные ассистенты Салют, 2022]. Семейство голосовых помощников «Салют» — это три многофункциональных бота. Каждый бот обладает собственным тембром сгенерированного голоса и стилем общения с пользователем. По задумке разработчиков, пользователь может выбирать бота в зависимости от своего настроения и целей использования умных систем.

Пожалуй, наиболее известной российской системой, разработанной с применением методов обработки естественного языка, является голосовой помощник «Алиса» [Алиса — голосовой помощник от компании Яндекс, 2022], выпущенный компанией Яндекс в 2017 году. Система быстро завоевала популярность. Например, всего за несколько лет «Алиса» стала надежным спутником автолюбителей (систему можно встроить в бортовой компьютер автомобиля [Яндекс.Авто, 2022]). Функционал «Алисы» стал предметом исследования ученых: лингвисты изучают возможности использования системы от Яндекса для преподавания русского языка как иностранного [Аль-Кайси и др., 2019, с. 240-244; Ратников, Чафонова, 2019, с. 65-66], а специалисты в области компьютерных наук разрабатывают для нее новые навыки [Дубельщикова, Тутова, 2020, с. 93-95; Беломутов, 2020, с. 257-259].

Сфера применения таких систем как «Алиса», «Маруся» и «Салют» не ограничивается одной предметной областью. Виртуальные ассистенты могут облегчать рутинные дела дома, в офисе — выступать в роли системы информационной поддержки, а в автомобиле — подсказывать дорогу. Однако едва ли мы сможем использовать подобные системы для решения узкоспециализированных задач, таких как консультация по вопросам медицинского и социального обслуживания или извлечение информации из базы данных по конкретной научной области.

Виртуальные ассистенты, рассмотренные выше, относятся к системам с открытой предметной областью (или открытым доменом). Такой тип диалоговых систем можно назвать универсальным: модели могут поддерживать диалог на свободную тему и решать широкий спектр задач [Adiwardana et al., 2020, с. 1]. Системы с закрытой предметной областью (или закрытым доменом) приходят на помощь, когда появляется необходимость в тематической специализации (например, нужен онлайн-консультант по вопросам финансов или медицины).

Как правило, разработка диалоговых систем с закрытым доменом для решения узкоспециализированных задач предпочтительнее использования аналогичных моделей с открытой предметной областью. В системах с закрытым доменом используются наборы данных (или датасеты), которые содержат информацию по конкретным темам. Для построения датасетов могут использоваться аннотированные диалоги [Su et.al., 2020, с. 6606-6607], графы знаний [Soufyane et al., 2021, с. 142-143] и другие типы данных.

Диалоговые системы с закрытой предметной областью позволяют удовлетворить информационную потребность пользователя по конкретной теме. Специализированные наборы данных в основе таких моделей позволяют системам предсказывать пользовательские запросы. Например, если диалоговая система и ее обучающий датасет содержат информацию о заболеваниях печени, вероятность того, что пользователь спросит о болезнях сердца, низка.

В настоящем исследовании мы сфокусируемся на разработке систем с закрытым доменом и рассмотрим подходы к построению моделей, которые смогут найти применение в сферах медицины и социального обслуживания. В данной работе описывается применение техник машинного обучения к построению вопросно-ответных систем на тему расстройств аутистического спектра для русского языка.

Исследование методов построения диалоговых систем для взрослых и детей в спектре аутизма не является новшеством, а разработка чат-ботов для сферы здравоохранения стала трендом второго десятилетия XXI века. Однако наши работы по созданию диалоговых систем об аутизме (по данным Google Scholar [Google Scholar, 2022]) являются первыми описанными опытами для русского языка в этой предметной области. Обратимся к существующим научным публикациям о применении диалоговых систем в социальной и медицинской сфере.

В 2003 году, когда внедрение диалоговых систем в медицинскую сферу считалось нетривиальной задачей, М. Беверидж и Д. Миллворд предложили мультимодальную диалоговую систему с возможностями генерации HTML-страниц и голоса для извлечения информации из медицинской онтологии [Beveridge, Millward, 2003, с. 341-343]. Набор функций немногочисленных чат-ботов тех лет был ограничен [Bickmore, Giorgino, 2004, с. 2], но с годами развитие технологий искусственного интеллекта позволило исследователям расширить сферу применения медицинских диалоговых систем.

Современные медицинские чат-боты, снабженные алгоритмами искусственного интеллекта, способны извлекать информацию из баз данных и знаний, анализировать запросы пользователей и выступать в роли своеобразных рекомендательных систем. Например, с помощью обработки текстовых описаний симптомов такие системы могут направить пользователя к конкретному медицинскому специалисту [Divya et al., 2018, с. 3]. Другие модели способны консультировать пользователей о применении и дозировках определенных медицинских препаратов [Madhu et al., 2017, с. 245].

Мировые тенденции в области здравоохранения и актуальные проблемы общества мгновенно находят отражение в разработках специалистов. Например, вызовы последних лет, связанные с пандемией новой коронавирусной инфекции, вдохновили исследователей на создание вопросно-ответных наборов данных для построения информационных систем о COVID-19 [Möller et al., 2020, с. 1; Wang et al., 2020, с. 1-2]. В последние годы в социально-медицинской сфере растет интерес к обеспечению инклюзивной среды и повышению информационной осведомленности о том, как взаимодействовать с людьми, имеющими инвалидность [Pendergrass et al., 2019, с. 2-3].

Возникает необходимость в новом программном обеспечении, которое поможет выстроить коммуникацию между людьми с инвалидностью и без нее. Например, в 2018 году в Австралии был разработан чат-бот для детей с расстройствами аутистического спектра, в котором взаимодействие с пользователем производится с помощью картинок и символов. Специалисты могут предложить ребенку программу в рамках терапии, а родители — в качестве инструмента для выработки базовых навыков общения и взаимодействия [Cooper, Ireland, 2018, с. 65-67]. Годом позже был представлен аналогичный чат-бот на китайском языке. Диалоговая система предназначена для выработки навыков коммуникации и сбора исследовательских данных для врачей [Li et al., 2019, с. 524].

Мы рассмотрели научные труды, методологически близкие нашей работе. Это позволяет нам выявить исследовательские пробелы, существующие в сфере построения диалоговых систем для решения проблем медицинского и социального обслуживания. Первый и наиболее очевидный пробел — отсутствие диалоговых систем на русском языке, информирующих о расстройствах аутистического спектра. Результаты поиска в системе индексирования научных изданий Google Scholar показали, что на данный момент исследования, посвященные разработке русскоязычной диалоговой системы об аутизме (кроме наших исследований, ссылки на которые будут приведены ниже, в разделе об апробации результатов исследования) не проводились [Google Scholar, 2022].

Лингвистических наборов данных о медицине и социальном обслуживании на русском языке на данный момент существует немного. Наиболее полно такой текстовый материал представлен для английского и китайского языков [Blinov et al., 2022, с. 3]. Обычно социальные или медицинские данные для русского языка собираются исследователями в процессе решения конкретных задач. Например, в ходе разработки программного обеспечения ученые могут обратиться за помощью к научным медицинским организациям [Danilov et al., 2020, 164], сделать запрос в архив или собрать языковые данные экспериментально.

В контексте исследований на стыке социальной и медицинской сфер сбор и систематизация материала для обработки русского языка представляет собой нетривиальную и актуальную задачу. Результаты нашего литературного обзора показывают, что в области разработки систем искусственного интеллекта для русского языка обнаруживается исследовательский пробел.

Исследования, с которыми мы ознакомились [Möller et al., 2020; Wang et al., 2020; Cooper, Ireland, 2018; Li et al., 2019], описывают процесс разработки программного обеспечения для языков, отличных от русского. Вместе с тем исследования подходов к построению русскоязычных медицинских диалоговых систем показывают, что методы, получившие распространение в англоязычном сегменте (например, использование фреймов, машинного обучения и онтологий), столь же актуальны и для русского [Vatian et al., 2019, с. 179-182].

Следующий исследовательский пробел, который мы обнаружили, связан с недостаточностью описания аспектов разработки социальных и медицинских диалоговых систем. В процессе изучения литературы по этой теме у нас возникло три исследовательских вопроса. Рассуждения, приведенные в выводах к данной работе, не дадут нам однозначного ответа на все вопросы. Однако в будущем мы планируем пригласить целевую аудиторию к дискуссии, которая, вероятно, приблизит нас к новым решениям обозначенных проблем.

Первый вопрос заключается в следующем: как помочь человеку с расстройствами аутистического спектра, не навредив ему? Целевой аудиторией социально-медицинских систем могут быть люди с особыми потребностями. Например, целевой аудиторией диалоговых систем, которые мы исследуем, являются, в том числе люди с расстройствами аутистического спектра (в частности, с высокофункциональным аутизмом, синдромом Аспергера). Люди в спектре аутизма обладают повышенной чувствительностью. Любое слово, которое они считают недобрым, может вызвать у них тревожность или спровоцировать агрессивное поведение. Поэтому исследование безопасных, контролируемых систем должно быть в приоритете при создании продуктов с такой целевой аудиторией.

Опасность систем машинного обучения заключается в том, что они с трудом поддаются контролю. В особенности это касается генеративных алгоритмов — выдачи таких моделей сложно предсказать. Они могут напугать пользователя в спектре аутизма и вызвать тем самым нежелательные последствия. Использование более примитивных методов, например, создание систем на правилах, может решить эту проблему. Но тогда нам потребуется значительно больше времени и ресурсов на разработку гибкой системы с большой базой знаний.

Проблема контролируемости моделей машинного обучения стала особенно актуальна в последние годы в связи с увеличением вычислительных мощностей компьютеров и распространением генеративных алгоритмов. Например, в 2021 году была представлена контролируемая диалоговая система для диагностики заболеваний — генеративный чат-бот, задающий вопросы о самочувствии пользователя. Модули контроля чат-бота позволяют модели придерживаться точности в медицинских суждениях и обеспечивают лексическое разнообразие для комфортного взаимодействия с пользователем [Compton et al., 2021, с. 111]. Однако исследования о контролируемости медицинских диалоговых систем не учитывают проблемы разработки продуктов для людей в спектре аутизма. В связи с этим мы считаем, что этому вопросу необходимо уделить внимание в настоящей работе.

Второй исследовательский вопрос звучит следующим образом: как создать систему, которая не будет вводить пользователя в заблуждение? Небольшая неточность, например, пропуск частицы «не» в выдаче диалоговой системы, может привести к порождению ложных фактов и дезинформированию пользователя. Ответ на первый исследовательский вопрос предполагает изучение эмоциональной составляющей выдач чат-бота. Нам предстоит рассмотреть формулировки ответов системы, тональность генерируемого текста, его лексическое наполнение. Второй вопрос приглашает нас к размышлению об информационном наполнении выдач и точности передачи фактов.

Чтобы ответить на второй исследовательский вопрос, некоторые исследователи проводили эксперименты с использованием генеративных или экстрактивных подходов к построению диалоговых систем. Например, выделенный нами вопрос исследовался в контексте создания моделей для понимания прочитанного, то есть для генерации ответов на вопросы по заданному тексту [Luo et al., 2022, с. 4-8]. На выбор между генеративными и экстрактивными алгоритмами могут влиять объемы входных данных, количество задач, решаемых системами, предметная область и другие переменные. Конкретные домены требуют отдельного исследования выбора алгоритма для передачи информации без искажений. В настоящем исследовании мы представим обзор и анализ решений этой проблемы в контексте предметной области «расстройства аутистического спектра».

Третий исследовательский вопрос формулируем следующим образом: как найти баланс между игровой составляющей диалоговой системы и ее информативностью? В нашем исследовании разрабатываемая модель предназначена как для высокофункциональных людей с аутизмом, так и для нейротипичной аудитории (без ментальных особенностей). Желательно, чтобы реплики чат-бота приглашали пользователя к игре. По крайней мере, диалоговая система должна представлять дружелюбный языковой интерфейс. При этом интерактивная составляющая системы должна быть стилистически нейтральной, иначе модель превратится в развлекательный чат-бот.

Сохранить баланс между игрой и информативностью помогут средства компьютерной лингвистики. Например, модель можно снабдить фреймами или правилами для общения с людьми в спектре аутизма. Обучающие данные можно дополнить репликами, обеспечивающими дружелюбное взаимодействие. Удачный пример — использование речевого стиля Мэри Поппинс, персонажа сказочных повестей Памелы Трэверс, в системе «Алиса» компании Яндекс [Академия Яндекса, 2020].

В ходе работы над исследованием мы консультировались с психиатрами, которые работают с детьми и взрослыми с расстройствами аутистического спектра. Это позволило нам изучить перспективы использования чат-ботов в качестве источника информации об аутизме.

Обозначим новизну данной работы. Во-первых, мы рассмотрели проблемы построения контролируемых, психологически безопасных диалоговых систем для людей в спектре аутизма и попытались найти варианты решений. Наши решения предназначены для русского языка и имеют лингвистическое обоснование. Учитывались такие факторы, как лексическое наполнение генерируемых текстов, формулировки высказывания и синтаксическая структура.

Во-вторых, мы рассмотрели этапы разработки информационных диалоговых систем для конкретной предметной области — «расстройства аутистического спектра». В Главе 3 «Подходы к построению системы» мы проанализировали возможные решения для разработки разговорных моделей этого домена. Наши решения были оценены с помощью специальных метрик. Система оценивания и результаты даны в Главе 3.

В-третьих, мы попытались изучить проблему сохранения баланса между игровой и информационной составляющей чат-бота, который впоследствии может быть внедрен в инклюзивную среду. В работе предлагаются решения на стыке компьютерных наук и лингвистики. Обсуждаются варианты модификации дизайна обучающих данных и способы разработки алгоритма. Чат-бот рассматривается как информационный посредник между нейротипичными и особенными людьми.

Актуальность настоящего исследования раскрывается в трех направлениях, которые мы обозначаем терминами «технологии», «лингвистика», «инклюзия». Суть технологического аспекта заключается в использовании нейросетевой архитектуры Transformer. На данный момент, эта система считается наиболее эффективным инструментом для обработки естественного языка. Модели, построенные на этой архитектуре (Bidirectional Encoder Representations from Transformers [Devlin et al., 2018], Generative Pre-trained Transformer [Radford, Narasimhan, 2018] и другие), эффективны в решении нетривиальных задач компьютерной лингвистики. Они находят практическое применение в таких популярных продуктах, как Google Translate [Caswell, Liang, 2020].

Лингвистический аспект актуальности работы заключается в том, что ее результаты включают языковую интерпретацию выдач компьютерных систем. Интерпретация результатов машинного обучения остается актуальной проблемой. Выдачи нейросетевых моделей до сих пор с трудом поддаются объяснению из-за большого количества обучающих параметров. Лингвистическая интерпретация играет важную роль при создании языковых интерфейсов, к которым относится наша диалоговая система. Языковой анализ незаменим, когда речь идет об инструментах для инклюзии. Язык остается основным средством коммуникации, как мост, соединяющий самых разных людей.

Инклюзивный аспект заключается в том, что результаты нашего исследования смогут найти практическое применение в образовательных организациях и учреждениях социальной защиты населения. Сегодня в России начинают активно распространять информацию о расстройствах аутистического спектра. Открываются тренировочные квартиры для людей с ментальными нарушениями, где они учатся самостоятельной жизни, создаются инклюзивные группы в детских садах, классы в общеобразовательных учреждениях. Такой инструмент, как чат-бот об аутизме, обеспечит поддержку для подобных организаций и их участников.

Наша мотивация заключается в личной необходимости в подобном инструменте. Непростой опыт жизни в инклюзивном общежитии, непонимание, трудности в общении с соседкой по комнате — девушкой с инвалидностью — вдохновили нас на создание дружелюбного приложения для нейротипичных и особенных людей.

Перечислим проблемы исследования:

1. психологическая безопасность систем искусственного интеллекта для инклюзивной среды;
2. передача информации по темам предметной области «расстройства аутистического спектра»;
3. построение дружелюбного языкового интерфейса.

Цель исследования заключается в том, чтобы по результатам экспериментов с применением различных подходов к построению диалоговых систем создать вариант методологических рекомендаций по разработке чат-ботов для поддержки инклюзии людей с расстройствами аутистического спектра. Для достижения цели поставлены следующие задачи:

1. собрать набор данных для обучения интеллектуальной диалоговой системы о расстройствах аутистического спектра;
2. провести эксперименты по тонкой настройке генеративных и экстрактивных вопросно-ответных моделей с архитектурой Transformer;
3. провести эксперименты по преобразованию дизайна и структуры обучающего набора данных с целью повышения производительности настроенных моделей;
4. провести автоматическую и ручную оценку полученных систем;
5. провести лингвистический анализ выдач модели для определения того, какие языковые особенности данных из обучающего набора влияют на техническую производительность систем;
6. по результатам исследования составить методологические рекомендации для разработчиков диалоговых систем для поддержки инклюзии людей с расстройствами аутистического спектра.

Материалом исследования послужили тексты с сайта сообщества поддержки людей с синдромом Аспергера и аутизмом, которые мы использовали с согласия администрации данного проекта [Autistic City, 2022]. Подробные сведения о том, как собирался и обрабатывался материал, даны в Главе 2 настоящего исследования «Материал исследования».

Объектом исследования стали диалоговые системы, которые могут использоваться в качестве инструмента поддержки инклюзии людей с расстройствами аутистического спектра. Предметом исследования являются алгоритмы искусственного интеллекта, которые обеспечивают работу таких диалоговых систем. Выделенные нами проблемы (например, обеспечение контролируемости и безопасности систем) были обнаружены в процессе работы с такими алгоритмами. По результатам исследования мы планируем представить методологические рекомендации для разработчиков диалоговых систем.

Методологический арсенал исследования включает общелогические, теоретические, эмпирические и специальные методы. К общелогическим методам относятся анализ и сравнение. В частности, планируется подробное изучение выдач диалоговой системы с точки зрения лингвистических особенностей сгенерированных текстов, а также их сравнение с материалами обучающего набора данных. В качестве теоретического метода будет использована проверка научной гипотезы. В качестве эмпирического метода мы выбрали эксперимент (подробное описание экспериментов представлено в Главе 3 «Подходы к построению системы»). Специальные методы компьютерной лингвистики и смежных дисциплин включают сбор и обработку материала исследования с помощью инструментов науки о данных, моделирование языковых систем, автоматическое и ручное измерение метрик оценки производительности компьютерных программ.

Мы выдвинули несколько гипотез. Во-первых, ожидается, что генеративные модели будут порождать случайные токены и нечастотные коллокации, создавать высказывания, не связанные с предметной областью диалоговой системы. Если контроль подобных выдач будет представлять трудность для разработчиков, мы не сможем рекомендовать использование генеративного подхода в качестве базового для построения диалоговой системы. Низкая предсказуемость и высокая вариативность делают модели небезопасными для людей с ментальными особенностями.

Во-вторых, предполагается, что экстрактивные системы, то есть модели, извлекающие ответ на вопрос из контекста, покажут высокие значения метрик оценки качества. Такие модели ничего не генерируют «от себя», а лишь воспроизводят отрывки текстов, полученных на входе. Мы ожидаем от таких систем высокой точности при извлечении фактов. Вероятно, они окажутся в приоритете в нашем списке рекомендаций. Следует учесть, что такие модели требуют ресурснозатратной подготовительной работы с данными, так как их производительность напрямую зависит от качества обучающих текстов.

В-третьих, предполагается, что преобразование структуры обучающего набора данных в значительной степени повлияет на производительность как генеративных, так и экстрактивных систем. Ожидается прямая зависимость между объемом обучающей выборки и производительностью систем: чем больше вопросно-ответных пар в наборе данных, тем выше будут значения метрик оценки качества независимо от архитектуры и параметров систем. Другие характеристики наборов данных (например, средняя длина ответа на вопрос), вероятно, будут в меньшей степени влиять на результат.

Поскольку продукт, описываемый в настоящем исследовании (диалоговая система для поддержки инклюзии людей с расстройствами аутистического спектра), представляет собой инструмент для благотворительной деятельности, в целях сбора обучающих данных для системы был запущен краудсорсинговый проект. Проект реализуется победителем конкурса «Практики личной филантропии и альтруизма» Благотворительного фонда Владимира Потанина. Детали краудсорсинга приведены в Главе 2 «Материалы исследования».

Эксперименты по модификации набора данных впервые были представлены нами на воркшопе Data-Centric AI в рамках конференции Conference on Neural Information Processing Systems 2021 (NeurIPS 2021) [Firsanova, 2021 (А)]. О результатах тонкой настройки моделей архитектуры Transformer мы рассказали на конференции Digital Transformation of Global Societies 2021 (DTGS 2021) [Firsanova, 2022]. Рассуждения на тему перспектив использования генеративных и экстрактивных моделей для построения чат-ботов были представлены нами на конференции Internet and Modern Society 2021 (IMS 2021) [Firsanova, 2021 (B)].

Структура используемого в настоящей работе набора данных впервые была представлена на конференции Dialogue 2021 в рамках студенческой сессии [Firsanova, 2021 (C)]. Метрики оценки производительности моделей, включая новую методику оценивания выдач диалоговых систем вручную, были описаны в одной из наших статей [Firsanova, 2021 (D)]. Впервые концепция проекта создания диалоговой системы о расстройствах аутистического спектра была представлена автором на воркшопе Human-AI Interaction в рамках 9th Conference on Artificial Intelligence and Natural Language (AINL 2020) [Firsanova, 2021 (E)]. Страница проекта представлена на сайте сообщества аналитиков данных Open Data Science (ODS) [Autism Spectrum Disorder QA, 2022].

# Глава 1. Калейдоскоп диалоговых систем

## 1.1. Развитие диалоговых систем

Диалоговые системы сегодня используются в бизнес-проектах, обеспечивают информационную поддержку в благотворительных организациях, отвечают на вопросы граждан о государственном социальном обслуживании, а иногда разрабатываются энтузиастами ради развлечения. Например, среди молодежи популярностью пользуются чат-боты для случайных встреч со студентами и выпускниками одного университета. Современные диалоговые системы поддерживают непринужденную беседу на свободную тему, информируют, извлекают из баз данных факты или мультимедийные объекты, даже сочиняют истории в процессе совместного творчества человека и машины. Наиболее успешный пример решения последней задачи — компьютерная игра в формате чат-бота AI Dungeon, которая предлагает пользователю сочинить собственное приключение в жанре фэнтези [Hua, Riley, 2020, с. 1].

Первые диалоговые системы появились в 60—70-х годах XX века. Прародительницей всех чат-ботов считается система ELIZA [Weizenbaum, 1966]. Модель ELIZA имитировала поведение психотерапевта, который придерживался метода Карла Рождерса (клиент-центрированная психотерапия). Суть метода заключается в том, что специалист задает клиентам вопросы, перефразируя их собственные высказывания. По мнению разработчиков ELIZA, реализация этого метода не требует знаний о мире, а значит идеально подходит для создания автоматической системы без доступа к внешнему источнику информации [Jurafsky, Martin, 2021, с. 7].

В основе модели ELIZA лежит алгоритм сопоставления с образцом (англ., *pattern matching*), набор правил для преобразования высказываний пользователя в вопросы (англ., *transform* *rules*) и список ранжированных ключевых слов (англ., *keywords*), то есть частотных выражений, которые могут встречаться во входных данных. Кроме того, система снабжена заготовками ответов на случай, если шаблон (образец) пользовательского запроса распознать не удалось [Jurafsky, Martin, 2021, с. 7-9].

Ключевые слова, заложенные в систему ELIZA, ранжированы. Ранг ключевого слова влияет на то, какой токен выберет ELIZA для перефразирования пользовательского запроса. Система использует ранги в случае, если ключевых слов во входных данных оказалось несколько. Ранг ключевого слова зависит от того, насколько вероятно получить от пользователя специфическую информацию при использовании этого токена для преобразования входных данных в вопрос. Слова с высокими рангами — в приоритете [Jurafsky, Martin, 2021, с. 7-9].

Рассмотрим пример использования ранжированных ключевых слов в системе ELIZA. По мнению Джозефа Вейценбаума, разработчика ELIZA, местоимение первого лица единственного числа должно иметь низкий ранг, так как при преобразовании утвердительного высказывания в вопрос оно потребует простой замены на местоимение второго лица. Например, если пользователь напишет *«я боюсь мышей»*, система, скорее всего, переспросит: *«Ты боишься мышей?»* В свою очередь, слово *«все»* в запросе пользователя потребует уточнения. Например, в ответ на запрос *«все меня ненавидят»* ELIZA выдаст: *«Кого ты подразумеваешь под всеми?»* Следовательно, слово *«все»* будет иметь более высокий ранг, и именно его ELIZA выберет для перефразирования, если в запросе пользователя оно встретится в связке с местоимением первого лица единственного числа [Jurafsky, Martin, 2021, с. 7-9].

Технология сопоставления с образцом, используемая в ELIZA, долгое время считалась одним из самых распространенных подходов к построению диалоговых систем [Devakunchari et al., 2019, с. 36] и других моделей обработки естественного языка. Например, этот метод использовался в системе SHRDLU [Winograd, 1972], одной из первых программ для понимания текстов на естественном языке. В основе SHRDLU — идея о неком пространстве блоков разных цветов и форм, которые пользователь может передвигать с помощью запросов к программе на естественном языке.

Другим примером использования метода сопоставления с образцом, является чат-бот A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) [Jurafsky, Martin, 2021, с. 9]. Работу системы A.L.I.C.E. обеспечивает язык разметки искусственного интеллекта AIML (Artificial Intelligence Markup Language). AIML представляет собой диалект (подмножество) расширяемого языка разметки XML (Extensible Markup Language) и предназначен для разработки чат-ботов, работающих по принципу «стимул-реакция» (stimulus-response). Разработчики заявляют, что A.L.I.C.E. является расширением системы ELIZA. При этом отличие A.L.I.C.E. заключается не только в использовании усовершенствованной технологии, но и в объеме базы знаний системы. Кроме того, A.L.I.C.E. снабжена возможностью выхода в сеть, которая необходима для сбора языковых данных в неограниченном объеме [Wallace, 2009, с. 182].

Не менее значимой диалоговой системой-первопроходцем является LUNAR [Woods, 1973]. Разработка LUNAR связана с космической миссией «Аполлон-11». Система принимала на вход вопросы пользователей о лунных горных породах на естественном языке (например, *«Какие образцы содержат X?»*) и выводила информацию из соответствующей базы знаний в виде конечного ответа или списка возможных результатов [Fergencs, 2020, с. 5].

Алгоритм в основе системы LUNAR — расширенная сеть переходов (Augmented Transition Network) — представлял собой одну из первых попыток автоматического синтаксического анализа (парсинга). Разбор структуры предложения машинными методами до сих пор остается трудоемкой задачей. Расширенная сеть переходов была предложена разработчиком LUNAR Вильямом Вудсом [Woods, 1970]. Вудс описывал свой алгоритм как эквивалент представления глубоких и поверхностных структур, представленных в теории трансформационной грамматики Хомского [Chomsky, 1965].

Алгоритм Вудса представлял собой автомат с магазинной памятью (англ., *pushdown* *automaton*), то есть конечный автомат, состояния которого сохраняются в стек (стек позволяет извлекать только последний, «верхний», из сохраненных элементов за один запрос) [Woods, 1970, с. 591]. Конечный автомат — это система с одним входом и выходом, которая принимает одно состояние из множества в заданный момент времени. Переходы из одного состояния в другое происходят при определенных условиях. Если нашей задачей является построение синтаксического разбора, то при достижении конечного состояния мы сможем получить результат парсинга — дерево [Белоусов, Ткачев, 2006, c. 495-521].

Расширенная сеть переходов — это рекурсивная графовая система. Узлами графа являются состояния, а ребрами — переходы. Расширенная сеть переходов использует процедуру нисходящего парсинга, то есть стратегию, при которой система сначала анализирует вершину синтаксического дерева, а затем спускается по нему и движется слева направо, используя правила формальной грамматики. В процессе парсинга сеть принимает на вход предложение и начинает свое прохождение по его структуре с вершины высказывания, то есть из стартового состояния. Затем сеть переходит к первому токену (к первому слову в предложении) и проходит далее, зацикливаясь на элементах, образующих группы слов (например, именные группы). Алгоритм завершает работу, когда сеть доходит до последнего слова в последовательности токенов [Woods, 1970, с. 591-596].

Как и алгоритм сопоставления с образцом, расширенная сеть переходов не утратила своей актуальности со временем. Изменилась лишь сфера применения этой модели. Например, в двухтысячных годах сеть переходов использовалась в качестве механизма контроля диалога в разговорных моделях. В частности, расширенная сеть переходов использовалась как базовый алгоритм работы движка, который связывал выходные данные языкового процессора системы с действиями виртуального персонажа, озвучивающего сгенерированные реплики. Движок контролировал использование истории диалогов с пользователем и знаний лингвистического процессора с помощью семантического интерпретатора [Guinn, Hubal, 2006, с. 397]. Описанная система задействовала одновременно несколько модулей и типов данных: алгоритмы делали запросы к базам знаний, генерировали аватары чат-ботов и порождали реплики. Расширенная сеть переходов использовалась в качестве связующего механизма внутри мультимодальной системы.

Адаптация старых алгоритмов под новые задачи, их преобразование и доведение до совершенства позволяет добиться впечатляющих результатов. Например, настоящим прорывом в области обработки естественного языка стало применение сверточных нейронных сетей (англ., *convolutional neural network*) к задаче классификации предложений [Kim, 2014]. Сегодня сверточные нейросети используются для фильтрации спама (бинарной классификации электронных сообщений), определения тональности интернет-комментариев (классификации текстов по наличию в них лексики или других элементов, передающих настроение автора) и других задач.

Сверточная нейросеть использует математическую операцию (свертку), чтобы после преобразования к числовому виду изображений, текстов, аудио или какой-либо еще информации упростить структуру входных данных и выявить повторяющиеся или сходные элементы, характерные для определенного класса. Такой тип нейронных сетей известен с 80-х годов XX века [Homma et al., 1988]. Первая сверточная нейросеть решала лингвистическую задачу — она использовалась для автоматического распознавания фонем.

## 1.2. Современные модели и трансферное обучение

Благодаря технологическому прогрессу увеличились доступные вычислительные мощности. Это позволило воплотить в жизнь мечты математиков и программистов XX века. Однако разработка новых моделей, таких как архитектура Transformer, не теряет актуальности.

Архитектура Transformer [Vaswani et al., 2017] была предложена в 2017 году в качестве модели для решения задачи машинного перевода. Сегодня Transformer — одна из самых популярных архитектур для обработки естественного языка. Архитектура Transformer используется для разработки вопросно-ответных систем, классификаторов текстов, моделей машинного перевода и других приложений.

Гибкость архитектуры Transformer обусловлена возможностями трансферного обучения (англ., *transfer* *learning*). Наиболее полное описание возможностей этого подхода к построению систем искусственного интеллекта, на наш взгляд, было представлено на мастер-классе конференции 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019) [Ruder et al., 2019].

Трансферное обучение проводится в два этапа. На первом этапе модель обучается с нуля (англ., *pre-training*), а на втором — производится ее тонкая настройка (англ., *fine-tuning*). Для обучения с нуля подбираются обобщенные задачи (например, автоматическое заполнение пропусков в предложении) и данные открытой предметной области (например, тексты из Wikipedia). В процессе тонкой настройки знания, приобретенные системой, переносятся (англ., *transfer*) на конкретную задачу, другой (обычно более сложный для обработки) естественный язык или специфическую предметную область. Суть тонкой настройки заключается в дообучении модели на новых данных с минимальными изменениями архитектуры. При этом дизайн наборов данных для тонкой настройки может значительно отличаться от структуры наборов, использовавшихся для обучения с нуля.

Реализация трансферного обучения в моделях с архитектурой Transformer заключается в дообучении готовой модели на новых данных и при необходимости добавлении к нейронной сети одного или нескольких слоев для преобразования информации, получаемой на выходе, под конкретную задачу. Архитектура Transformer является базовой для некоторых популярных моделей обработки языка. Например, Generative Pre-trained Transformer (GPT) использует Transformer для реализации модели-декодировщика. Обучающая задача GPT заключается в традиционном моделировании языка или генерации текста [Radford, Narasimhan, 2018]. Bidirectional Encoder Representations from Transformers (BERT) представляет собой модель-кодировщик. Задачи BERT — классификация предложений и предсказание пропущенного слова в тексте [Devlin et al., 2019].

Не менее популярны модели, которые объединяют кодировщик и декодировщик, повторяя оригинальное строение Transformer. Например, Bidirectional Autoencoder Representations from Transformers (BART) — это модель-автокодировщик, обученная распознавать «шум» в данных, то есть признаки, которые препятствуют обнаружению закономерностей в выборке [Lewis et al., 2019]. Другой пример — многозадачная Text-to-Text Transfer Transformer (T-5) [Raffel et al., 2020]. Эта модель способна решать несколько задач без дообучения, например, производить машинный перевод, классифицировать предложения и реферировать текст. Модель распознает задачу по префиксу, заданному пользователем (например, «*translate*» для перевода или «*summarize*» для реферирования).

В данный момент на платформе Hugging Face, которая представляет собой открытый репозиторий систем с архитектурой Transformer, опубликовано более 30 000 модификаций этих и аналогичных моделей [Models - Hugging Face, 2022]. Модификации, представленные в Hugging Face, были дообучены и настроены для решения специфических задач, работы с закрытыми предметными областями и конкретными языками. Поскольку архитектура Transformer выбрана нами для проведения экспериментов настоящего исследования, мы вернемся к обсуждению ее характеристик в разделе 1.4 данной главы «Архитектура Transformer».

Первоначально Transformer разрабатывался для решения задачи машинного перевода, однако возможности трансферного обучения сделали эту модель практически универсальной для обработки естественного языка. Например, архитектура Transformer применяется при разработке диалоговых систем. Рассмотрим возможности применения моделей-кодировщиков, декодировщиков и автокодировщиков с архитектурой Transformer для решения этой задачи.

К моделям-кодировщикам относятся системы, аналогичные Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2019]. По словам разработчиков BERT, модель предназначена для построения двунаправленного векторного представления как одного высказывания, так и двух связанных предложений, например, вопросно-ответной пары. Векторное представление позволяет построить для входных данных (например, списка предложений) пространство векторов, которое отображает закономерности совместной встречаемости токенов. Двунаправленное представление означает, что модель при построении вектора заданного слова учитывает его правый и левый контекст (словесное окружение).

Два свойства BERT — двунаправленность и возможность устанавливать наличие связи между двумя предложениями — обусловлены задачами, которые модель решала в ходе обучения [Devlin et al., 2019, с. 4174-4175]. Первая задача — маскированное моделирование языка (англ, *masked language modelling*) — использовалась для построения двунаправленных представлений. В ходе обучения модель получала на вход текст, в котором замаскировано 15% токенов, и училась предсказывать пропуски. Мы полагаем, что знания, полученные в ходе решения этой задачи, связаны с возможностью BERT извлекать информацию из текста, а именно искать ответы на вопросы.

Вторая задача BERT — предсказание следующего предложения (англ., *next sentence prediction*), по сути, представляет собой задачу бинарной классификации. Модель учится предсказывать, является ли высказывание X продолжением предложения Y, то есть отвечать на вопрос, может ли X следовать за Y в неком контексте. Знания, полученные в ходе решения этой задачи, наделяют систему «пониманием» того, какие признаки сигнализируют о наличии связи между двумя предложениями [Devlin et al., 2019, с. 4174-4175]. Знания о контексте токенов и наличии связи между предложениями позволяют BERT извлекать из заданного отрывка цитату, отвечающую на вопрос пользователя. Такой тип вопросно-ответных систем называется экстрактивным (от английского «*extract*»*,* то есть «*извлекать*»).

Реализация экстрактивного подхода к построению вопросно-ответных систем невозможна без обучающих данных с адаптированной под эту задачу структурой. Необходимые наборы данных должны содержать контекст, вопрос и образец ответа, чтобы модель могла научиться в процессе тонкой настройки извлекать факты. Наиболее известным примером такого набора данных является Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al., 2016]. Набор данных SQuAD содержит более 100 000 вопросов к статьям Wikipedia. Вопросно-ответные пары составлялись пользователями краудсорсинговой платформы. Отрывки текстов из Wikipedia, к которым задавались вопросы, не ограничены конкретными предметными областями. Набор данных универсален и может использоваться для обучения или настройки моделей-кодировщиков с архитектурой Transformer и любых других систем для понимания чтения.

Часто для построения диалоговых систем исследователи и разработчики используют модели-декодировщики с архитектурой Transformer. Они подходят для реализации генеративного подхода к построению диалоговых систем. Генеративный подход отличается от экстрактивного как принципом работы, так и сферой применения готовых моделей. В настоящем исследовании под генеративным подходом к построению вопросно-ответных систем мы подразумеваем традиционное моделирование языка с использованием возможности переноса знаний на новый домен без дообучения (англ., *zero-shot learning*) или построением фреймов для оформления реплик диалога. Рассмотрим особенности генеративных диалоговых систем подробнее.

Самой известной моделью-декодировщиком с архитектурой Transformer является Generative Pre-trained Transformer (GPT). Обучение первой версии GPT проходило в два этапа. Первый этап заключался в построении традиционной языковой модели методом обучения без учителя (англ., *unsupervised* *pre-training*) [Radford, Narasimhan, 2018, с. 3]. Традиционная языковая модель учится предсказывать следующий токен для заданной пользователем последовательности. Обученная модель принимает на вход пользовательский префикс, который она должна продолжить. Часто эту процедуру называют генерацией текста, а подход к построению подобных систем — генеративным.

Для обучения таких моделей требуются значительные объемы обучающих данных. Первая версия GPT обучалась на двух наборах: BookCorpus и 1B Word Benchmark. Первый набор содержит порядка 7 000 неопубликованных художественных книг различных жанров. Второй набор содержит новостные тексты объемом один миллиард слов [Radford, Narasimhan, 2018, с. 4].

Второй этап обучения GPT представлял собой процесс тонкой настройки, адаптации языковой модели к конкретным задачам методом обучения с учителем (англ., *supervised* *fine-tuning*). Если на первом этапе для обучения модели использовались «сырые» тексты без каких-либо меток, то на втором этапе GPT получала на вход только размеченные данные. Модель настраивалась под решение множества задач. Например, GPT «училась» классифицировать тексты. Также модель использовалась для выявления отношений между предпосылкой и гипотезой: цель GPT заключалась в том, чтобы подтвердить, опровергнуть или выявить отсутствие связи между двумя предложениями. Кроме того, модель настраивалась для сравнения текстов: целью GPT было выявление сходств между заданными предложениями. Наконец, GPT «училась» отвечать на вопросы по тексту, выбирая один вариант ответа из нескольких возможных [Radford, Narasimhan, 2018, с. 4].

Годом позже появилась усовершенствованная версия GPT-2, которая обучалась как традиционная языковая модель на наборе данных WebText. Набор WebText собирался разработчиками обновленной системы. Он содержит 45 миллионов ссылок, опубликованных на платформе Reddit. Данные для обучения GPT-2 извлекались с HTML-страниц, представленных в WebText [Radford et al., 2019, с. 4].

Последняя версия GPT-3 обучалась с использованием нескольких наборов данных. Например, разработчики включили в выборку отфильтрованные данные набора Common Crawl объемом 410 миллиардов токенов. Фильтрация этого набора была необходима, так как он собирался автоматически, и значительная его часть содержала сгенерированные или бессмысленные тексты. Кроме того, использовалась обновленная версия WebText объемом 19 миллиардов токенов, два набора данных художественной литературы общим объемом 67 миллардов токенов и тексты из Wikipedia объемом 3 миллиарда токенов. Разные наборы данных имели свой вес, например, тексты, полученные с помощью WebText, были приоритетнее литературных произведений [Brown et al., 2020, с. 8-9].

Можно утверждать, что система GPT-3 [Brown et al., 2020] проходит тест Тьюринга. Это означает, что действия GPT-3 с высокой точностью имитируют поведение человека. Такую интерпретацию теста Тьюринга приводил Джон Сёрль в своей статье по философии искусственного интеллекта «Сознание, мозг и программы» [Searle, 1980, с. 419].

Имитирующее поведение GPT-3 выражается в ее выдачах, реакциях на запросы и реплики пользователя и другие действия. Идентификаторами того, что GPT-3 проходит тест Тьюринга, является связность генерируемого моделью текста, уместные ответы на реплики пользователя, порождаемые с учетом контекста диалога и стиля письма собеседника-человека. Об этих способностях свидетельствуют возможности программного обеспечения, созданного с помощью GPT-3 [GPT-3 Demo, 2022].

Вычислительные мощности, необходимые для работы GPT-3, пока не предоставляются облачными сервисами и недоступны большинству пользователей. Разработчики GPT-3 (компания OpenAI) открыли доступ к интерфейсу программирования приложений этой системы, чтобы любой пользователь смог испытать новую модель [OpenAI API, 2022]. На данный момент использование GPT-3 недоступно в Российской Федерации (на Рисунке 1 представлен скриншот, подтверждающий эту информацию). Поскольку первая версия GPT считается устаревшей, а новая GPT-3 недоступна для жителей РФ, Глава 3 настоящего исследования «Подходы к построению системы» содержит описание генеративных экспериментов только с системой GPT-2.

Задача традиционного моделирования языка, заложенная во все версии GPT, по сути сводится к предсказанию следующего токена (одного слова или сочетания символов). Такие модели создают вероятностное пространство: они не моделируют структуру текста, не извлекают семантические отношения и не выстраивают синтаксические деревья. Тем не менее, вероятностное распределение — основа алгоритма генеративных систем — наделяет модели способностью обнаруживать аналогии, повторы, шаблоны построения текстов или каких-либо других типов данных.

Отметим, что GPT была адаптирована для генерации изображений [Chen et al., 2020], решения математических [Cobbe et al., 2021] и других задач. Явление, которое описывает универсальность генеративных систем — находить связи, проводить аналогии и переносить накопленные знания о предмете на специфические задачи — называется мета-обучением (англ., *meta-learning*).

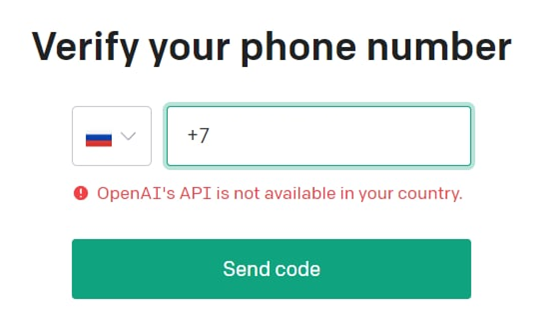


Рисунок 1. Интерфейс программирования приложений OpenAI для работы с GPT-3 недоступен для жителей Российской Федерации в 2022 году

Мета-обучение позволяет разработать модель, подходящую для решения одновременно нескольких задач, без дообучения или тонкой настройки [Brown et al., 2020, с. 4]. Разработчики GPT выделяют три сценария мета-обучения в зависимости от количества образцов (демонстрационных вариантов решения различных задач), которые модель получает на вход в процессе обучения. В первом случае модель получает от десяти до ста демонстрационных вариантов (англ., *few-shot learning*). Во втором случае — один образец (англ., *one-shot learning*). В третьем случае модель не получает ни одного примера решения задачи (англ, *zero-shot learning*) [Brown et al., 2020, с. 5]. Именно этот тип мета-обучения генеративных систем интересует нас в контексте настоящего исследования.

Способность генеративных систем решать новые задачи и обрабатывать домены, которые не были представлены в обучающей выборке, (англ, *zero-shot learning*) позволяет использовать GPT в качестве диалоговой системы. Например, чтобы использовать GPT как вопросно-ответную модель, достаточно ввести вопрос на естественном языке, например: *«Сколько планет в Солнечной системе?»* Вопросительные предложения обладают характерными лингвистическими особенностями, такими как использование частотных вопросительных слов и определенной структуры высказывания. Поэтому система способна распознавать языковые шаблоны и генерировать ответы. Это происходит за счет zero-shot learning, так как специально выстроенные вопросно-ответные пары не использовались для обучения GPT (аналогичные эксперименты проводились для исследования преимуществ one- и few-shot learning [Brown et al., 2020, с. 14]).

Тем не менее тонкая настройка может понадобиться при использовании таких моделей, если мы хотим построить систему с закрытым доменом. Для этого необходимо подготовить набор данных в виде реплик диалога (типа *«вопрос — ответ»* или *«реплика — ответ на реплику»*). В качестве обучающих данных можно использовать комментарии из вопросно-ответных сервисов (например, Quora или Ответы Mail.ru) или киносценарии. Наше решение этой задачи представлено в Главах 2 «Материал исследования» и 3 «Подходы к построению системы».

Методы применения генеративных моделей к построению диалоговых систем не ограничиваются использованием мета-обучения. Другим возможным вариантом может быть создание фреймов диалоговых реплик. Примером такой архитектуры служит система DialoGPT, которая представляет собой расширение GPT-2 [Radford et al., 2019]. Система DialoGPT использует задачу генерации текста для моделирования диалогов с множеством реплик. Основное отличие DialoGPT от GPT-2 заключается в использовании данных социальной сети Reddit для обучения (разработчики извлекали из Reddit дискуссии в диалоговом формате). Гипотеза исследователей заключалась в том, что в ходе обучения на таких данных DialoGPT сможет моделировать совместное распределение токенов в разговоре [Zhang et al., с. 272].

Одним из интересных примеров использования моделей-автокодировщиков с архитектурой Transformer для построения диалоговых систем является разработка чат-бота о новой коронавирусной инфекции COVID-19 с помощью BART [Lewis et al., 2019]. Для этой цели разработчики использовали набор Wizard of Wikipedia. Набор Wizard of Wikipedia состоит из вопросов пользователя («ученика») и ответов человека-эксперта («мудреца»), у которого есть доступ к программному обеспечению для извлечения информации из базы знаний Wikipedia [De Bruyn et al., 2020, с. 2].

Разработчики диалоговой системы о COVID-19 адаптировали архитектуру BART для решения своей задачи. Модели-автокодировщики состоят из элементов для кодирования и декодирования информации. Кодирующие элементы в адаптированной модели BART принимали на вход пару ключ-значение для обработки и обнаружения связи между историей диалога с пользователем (значение) и информации из базы знаний (ключа). Декодировщик получал на вход обработанные кодировщиками данные. Подобно GPT, декодировщик генерировал тексты, то есть ответы на реплики пользователя на естественном языке.

Итак, в разделе 1.1 Главы 1 «Развитие диалоговых систем» мы рассмотрели модели и технологии, которые оказали влияние на нашу работу. Нам удалось пронаблюдать, как разработки 60—90-х годов XX века стали источником вдохновения исследователей XXI века. Первые алгоритмы искусственного интеллекта, усовершенствованные и усиленные за счет появления новых вычислительных мощностей, заиграли новыми красками в XXI веке — многие ученые нашли нетривиальные способы применения старых моделей и добились в этом успеха.

Кроме того, мы рассмотрели возможности современных систем на примере моделей с архитектурой Transformer. Мы сделали обзор основных понятий, используемых в данном исследовании: трансферное обучение (перенос знаний), мета-обучение, экстрактивный и генеративный подход к построению диалоговых систем, модели-кодировщики, декодировщики и автокодировщики. В следующем разделе мы рассмотрим базовые типы задач, которые могут использоваться для обучения моделей разговорного искусственного интеллекта (англ., *conversational* *artificial* *intelligence*).

## 1.3. Базовые модули диалоговых систем

Сфера разработки интеллектуальных диалоговых систем называется разговорным искусственным интеллектом (англ., *conversational* *artificial* *intelligence*). Разработки в этой области объединяют достижения ученых в сферах обработки естественного языка и машинного обучения. Автор исследования имеет опыт построения диалоговых систем, поэтому не лишним будет начать данный раздел работы с некоторых наблюдений из личной практики.

Системы разговорного искусственного интеллекта состоят из нескольких модулей. Например, любая система должна включать алгоритмы обработки запросов, которые могут использоваться для анализа истории диалога и последних реплик пользователя. Если диалоговая система имеет цель информировать пользователя, то, скорее всего, она также будет иметь доступ к базе знаний или данных и использовать алгоритмы извлечения информации (англ, *information* *retrieval*). Также любая модель нуждается в модуле, отвечающем за генерацию выходных данных, то есть ответа на реплику пользователя. Такой модуль определяет форму и содержание ответа, его модальность (например, система может выводить только текст, а может — текст, снабженный изображением и аудиоматериалом). В данном разделе мы рассмотрим реализацию этих модулей, базовых элементов диалоговой системы, на примере алгоритмов, используемых в настоящем исследовании.

Выше мы упоминали о том, что информационная система может быть оснащена алгоритмом извлечения информации, который позволяет модели производить запрос к базе данных или знаний. Разработкой и исследованием возможностей систем с таким модулем занимаются специалисты в двух областях искусственного интеллекта — обработке естественного языка (англ., *natural* *language processing*) и извлечении информации (англ., *information* *retrieval*). Рассмотрим пример работы такой системы на стыке двух областей.

Вопросно-ответные системы, которые производят запрос к базам знаний (англ., *knowledge base question answering*), обычно решают пять основных задач: анализ вопроса пользователя, картирование входных данных, разрешение неоднозначности, составление запроса к базе знаний и извлечение информации. Анализ вопроса пользователя — задача преимущественно лингвистическая, то есть относящаяся к обработке естественного языка. Система производит синтаксический разбор высказывания с целью извлечь информацию, требуемую пользователем. Для этого модель может провести анализ типа вопроса, извлечь именованные сущности, установить отношения между единицами высказывания, например, выделить меронимы (слова, обозначающие составные части какого-либо объекта) и холонимы (слова, обозначающие целую часть для каких-либо составляющих) [Diefenbach et al., 2018, с. 535-536]. В терминах описанных выше модулей, этот алгоритм относится к элементу обработки запроса пользователя.

Картирование пользовательского запроса заключается в соотнесении единиц, выделенных системой в ходе лингвистического анализа, и имеющимися в базе знаниями. Например, система ищет в базе единицы, названия которых совпадают с выделенными именованными сущностями. Выявленные единицы не всегда однозначны (например, именованная сущность «Москва» может обозначать название города, реки, гостиницы, судна, песни, художественного или документального фильма, а также иметь другие значения), поэтому следующим этапом становится разрешение неоднозначности, за которым следует составление запроса на соответствующем языке, например, SPARQL.

Составленный запрос используется для извлечения информации из базы знаний [Diefenbach et al., 2018, с. 535-537]. В терминах наших модулей, описанные этапы представляют собой комплексный элемент для извлечения информации и вывода ответа (система использует язык запроса и выводит информацию из базы знаний).

Четыре этапа работы описанного алгоритма вопросно-ответной системы с базой знаний относятся к области извлечения информации, и только один связан с лингвистикой. Мы можем утверждать, что системы, в основе которых лежит работа с базами знаний, связаны с лингвистикой и обработкой языковых данных косвенным образом. В настоящем исследовании мы фокусируемся на проблемах обработки естественного языка и лингвистической составляющей работы диалоговых систем, поэтому мы находим избыточным включение моделей с базами знаний в наши эксперименты. Тем не менее, изучение таких моделей необходимо для понимания принципов работы экстрактивных алгоритмов, извлекающих информацию не из базы знаний, а из текстов.

Рисунок 2 представляет упрощенную схему работы экстрактивного алгоритма для вопросно-ответных систем, суть которого заключается в извлечении ответа на вопрос из контекста. Экстрактивные модели не делают запрос к базе данных напрямую, однако, как и системы с базами знаний, они содержат все, описанные нами выше, модули систем разговорного искусственного интеллекта. Рассмотрим ход работы экстрактивных систем.

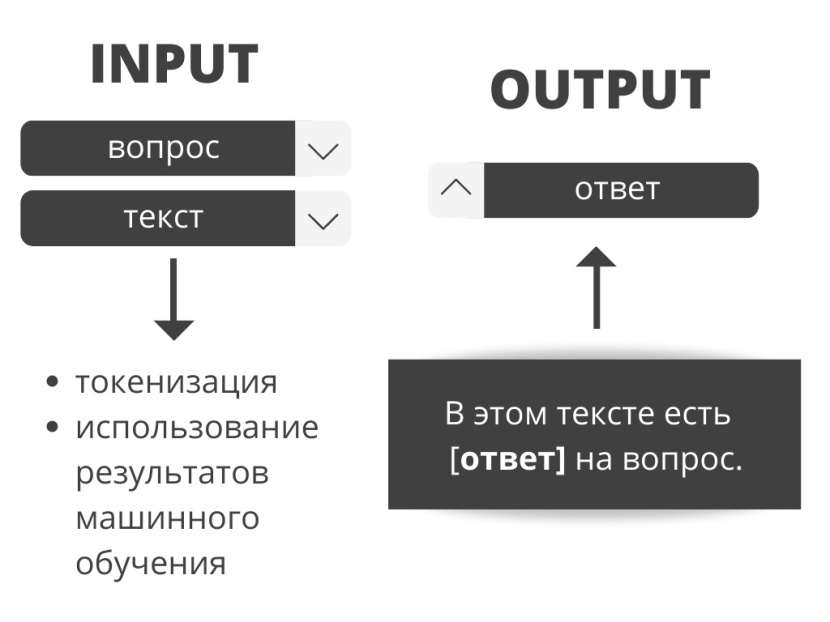


Рисунок 2. Упрощенная схема работы экстрактивных систем

В отличие от моделей с базами знаний экстрактивные алгоритмы не делают запросов к неким массивам данных, они лишь используют знания (например, веса обученной нейронной сети) для извлечения ответа на вопрос по заданному тексту. Как показывает Рисунок 2, на вход такие модели получают два элемента: вопрос и текст, из которого нужно извлечь информацию. Модуль обработки входных данных обычно ограничен токенизацией вопроса пользователя и текста с последующим преобразованием токенов к виду векторов для нейросетевой обработки [Gao et al., 2019, с. 173].

Модуль извлечения информации в экстрактивных системах представляет собой процесс поиска вектора ответа. В процессе обучения нейросети модель учится сопоставлять тексты, вопросы и готовые ответы. Веса обученной нейронной сети по сути представляют набор признаков, по которым система может идентифицировать ответ (а точнее, его векторное представление), то есть тот отрывок текста, который может ответить на вопрос пользователя.

Для извлечения ответа модель использует веса обученной нейросети, а также вектора вопроса и текста, полученные на предыдущем этапе работы в модуле обработки запроса. Реализация алгоритма поиска ответа зависит от особенностей архитектуры системы. Например, модель может использовать ранжирование векторов, сопоставляя вопрос с возможными отрывками текста, полученного на входе. Важную роль играет токенизация текста. Она определяет вид отрывков, которые модель может использовать в качестве ответа [Gao et al., 2019, с. 173].

Модуль вывода в экстрактивных системах состоит из элемента, который декодирует вектор найденного моделью ответа и приводит его к виду текста на естественном языке. Ответ на вопрос в таких системах представляет собой отрывок из текста, полученного моделью на входе. Следовательно, процесс вывода сводится к картированию вектора ответа к его положению в заданном тексте [Gao et al., 2019, с. 173].

Частным случаем реализации экстрактивного подхода является решение задачи машинного понимания прочитанного текста (англ., *machine reading comprehension*). Классическим образцом решения этой задачи является использование материалов Стенфордского вопросно-ответного набора данных SQuAD [Rajpurkar et al., 2016] в качестве обучающих данных для нейросетевой модели. Модель получает на вход вектор вопроса и вектор текста . Задача модели заключается в том, чтобы определить расположение ответа на вопрос в тексте .

Ответ состоит из двух частей. Часть обозначает расположение первого элемента ответа в тексте, а часть — расположение последнего [Gao et al., 2019, с. 173]. Чтобы получить и , мы представляем текст в виде набора порядковых чисел, каждый из которых обозначает положение символа в тексте. Например, для текста *«да, это так»* мы выделили ответ *«это»*. Мы кодируем текст *«да, это так»* последовательностью [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], в которой слово *«это»* зашифровано номерами 3, 4 и 5. Следовательно .

Обратимся к генеративному подходу к построению вопросно-ответных систем (англ., *generative question answering*). Одним из вариантов использования генеративного подхода к построению вопросно-ответных систем является применение мета-обучения к традиционному моделированию языка. Возможности мета-обучения обсуждались нами в разделе 1.2 «Современные модели и трансферное обучение» настоящей главы. Для построения такой системы можно использовать предварительно обученную генеративную модель, например, GPT-2 [Radford et al., 2019], и набор данных с вопросно-ответными парами (киносценарии или линии диалогов из социальных сетей).

Традиционное моделирование языка — задача, подходящая для моделей машинного обучения без учителя. Задача системы заключается в том, чтобы смоделировать распределение токенов в последовательностях (предложениях) разной длины . Распространенным решением этой задачи (активно использующимся с 80-х годов XX века) является факторизация совместного распределения символов через произведение условной вероятности [Radford et al., 2019, с. 2]:

Входными данными для генеративных систем служат префиксы, то есть последовательности, которые вводит пользователь и которые модель должна продолжить. Если генеративная модель используется в качестве основы для вопросно-ответной системы, то ожидается, что пользователь будет вводить в качестве префиксов вопросы в полной форме (например, *«Что обозначает аббревиатура РАС?»*) или запросы, аналогичные тем, что используются для работы с информационно-поисковыми системами (например, *«аутизм особенности дети»*). Модуль для обработки запроса в генеративной системе будет использоваться для токенизации префикса и преобразования его к виду векторов.

Мета-обучение обеспечивает работу модуля извлечения ответа на вопрос в генеративных системах. «Запоминая» закономерности распределения токенов, модель накапливает своего рода знания. Например, тексты специфических предметных областей будет отличать наличие соответствующих терминов. Наличие в тексте (в префиксе или в вопросе пользователя) такой лексики будет своего рода флажком для генеративной модели. Распознав характерные слова, модель сможет генерировать тематический текст или ответ на вопрос. Свою «память» модель будет использовать в качестве базы знаний.

Генеративные модели, как и экстративные, используют векторные представления для моделирования вероятностного распределения. Метод токенизации может различаться, однако в приоритете обычно — деление на подслова (англ., *subword tokenization*). Например, модель GPT и ее разновидности используют кодирование диграмм, или пар байтов (англ., *byte pair encoding*) [Radford et al., 2019, с. 4]. Метод, изначально предназначенный для реализации алгоритма сжатия информации [Gage, 1994], нашел свое применение в области токенизации данных для машинного перевода, а теперь используется повсеместно в обработке естественного языка. Суть метода в том, что наиболее частотные пары байтов (диграммы) кодируются новым символом. Операция повторяется несколько раз, пока мы не получаем сжатый кодированный текст. Метод позволяет избегать проблем со словами, не зафиксированными в обучающем словаре (англ., *out-of-vocabulary words*), и считается более гибким, чем пословная или посимвольная сегментация [Sennrich et al., 2015, с. 2].

Поскольку такие модели, как GPT, используют подсловную (англ., *subword*) токенизацию текста, они анализируют частотные элементы слов, а не слова целиком или отдельные символы. Ожидаемым поведением модели будет создание «нейросетевых окказионализмов». Примером может служить слово *«морники»*, которое сгенерировала система компании Яндекс как перевод английского *«morning»* [Блог Яндекса, 2017]; слово имеет морфологическую структуру, характерную для русского языка, однако в словарях не зафиксировано. Кроме того, вероятно искажение фактов, потому что одно и то же слово или даже часть предложения может встретиться в неограниченном количестве контекстов в разном значении.

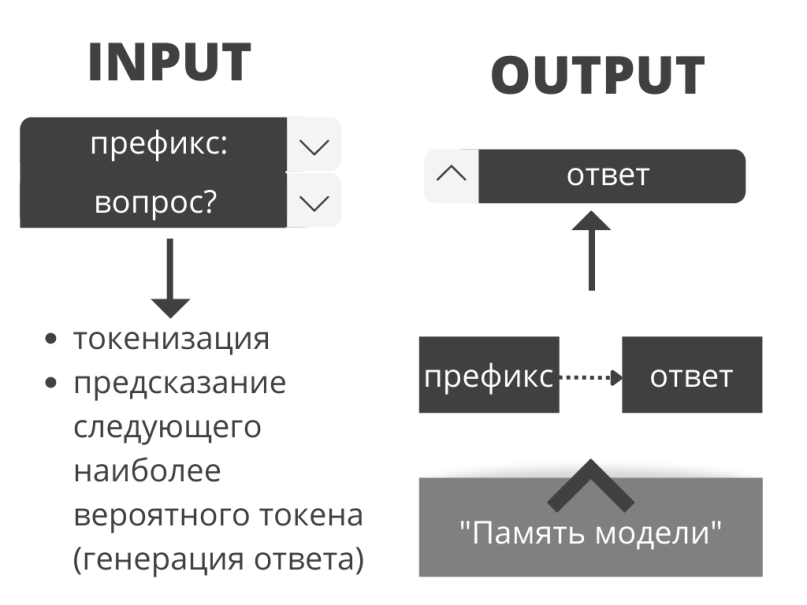


Рисунок 3. Упрощенная схема работы генеративных систем

Модулем вывода в генеративной системе выступает алгоритм-декодировщик. Рисунок 3 иллюстрирует работу генеративной модели, адаптированной под вопросно-ответную систему. На изображении представлен префикс, концепция работы традиционного моделирования языка и вывод ответа. Архитектура моделей, выбранных нами для реализации экстрактивного и генеративного подходов, описаны в разделе 1.4 «Архитектура Transformer».

## 1.4. Архитектура Transformer

Изначально разработанная для решения задачи машинного перевода, на сегодняшний день архитектура Transformer считается универсальным, легким для обучения и одним из самых производительных инструментов для решения широкого спектра задач обработки естественного языка [Wold et al., 2020, с. 38]. Обычная, или «ванильная» (англ., *vanilla*), версия архитектуры, описанная в статье 2017 года [Vaswani et al., 2017], состояла из кодировщика и декодировщика, как и предшествующая ей популярная модель для машинного перевода Sequence-to-Sequence [Sutskever et al., 2014]. Рассмотрим базовые элементы Transformer.

Кодировщик Transformer представляет собой стек из шести идентичных слоев. Каждый слой кодировщика состоит из двух элементов: механизма многоголового внутреннего внимания (англ., *multi-head self-attention mechanism*) и нейронной сети с прямой связью (тип нейронной сети без циклов; считается наиболее простым). Декодировщик модели Transformer также представляет собой стек из шести слоев, однако в нем каждый слой состоит из трех элементов. Помимо внимания и нейронной сети, слои декодировщика включают в себя дополнительный элемент для воспроизведения результатов работы многоголового механизма, то есть для обработки выходных данных стека-кодировщика [Vaswani et al., 2017, с. 2-3].

Чтобы понять, что представляет собой механизм многоголового внутреннего внимания, рассмотрим термин внимание (англ., *attention*) в контексте разработки систем искусственного интеллекта. Внимание — это механизм, который позволяет модели фокусироваться на отдельных токенах из запроса пользователя или обучающей выборки. Механизм внимания позволяет модели искать связанные друг с другом слова (например, находить перевод для заданного слова в параллельном корпусе) и присваивать более высокие значения весов тем парам слов, которые с наибольшей вероятностью связаны между собой.

Разработчики Transformer предлагают описывать математическую функцию механизма внимания в терминах науки об извлечении информации: запрос, ключ и значение. Они предлагают новый тип механизма внимания — масштабированное скалярное произведение (англ., *scaled dot-product attention*). На вход такая модель получает векторы запроса и ключа размерности , а также вектор значения размерности Скалярное произведение векторов запроса , ключа и значения делится на квадратный корень размерности ключа и запроса . При этом вектор ключа транспонируется (иначе он может совпасть с вектором значения ). Далее применяется функция , которая позволяет преобразовать полученные значения для последующей обработки [Vaswani et al., 2017, с. 3-4]:

Понятия «запрос», «ключ» и «значения» достаточно условны в контексте обработки естественного языка. Это любые токены, связи между которыми пытается найти модель. В системах извлечения информации, например, в системе поиска по каталогу видеоматериалов, запросом будет выступать непосредственно запрос пользователя (например, *«смешные видео котиков»*), ключом — набор меток для уточнения поиска (например, *«короткие видео, видео для детей»*), а значением — список релевантных видео.

Стандартный механизм внимания в системах машинного перевода ищет связи между токенами исходного и целевого языков. Механизм внутреннего внимания (англ., *self-attention*), предложенный разработчиками Transformer, предполагает также поиск связей между элементами одного (например, исходного) языка. Такая модель способна устанавливать связи между местоимениями и существительными, которые оно заменяет.

Улучшенный алгоритм внимания называется многоголовым (англ., *multi-head*), потому что он выполняет операцию расчета скалярного произведения несколько раз. Количество операций при этом равно количеству голов обработки [Vaswani et al., 2017, с. 4-5]. Каждый раз система рассчитывает веса для разных элементов. Например, системе дано предложение: *«Кошка перебежала дорогу, потому что она испугалась».* Модель с двумя головами сначала просчитает силу связи между словами *«кошка»* и *«она»*, а затем повторит то же самое для лексем *«дорога»* и *«она».*

Итак, мы рассмотрели основные задачи, которые могут использоваться для обучения вопросно-ответных моделей, типы наборов данных, необходимые для построения таких систем, а также устройство архитектуры, используемой в настоящем исследовании. В следующих главах мы опишем процесс создания набора данных для исследования, а также наши эксперименты и проанализируем собранный материал, чтобы составить набор методических рекомендаций по разработке интеллектуальных систем для поддержки инклюзии.

# Глава 2. Материал исследования

Материалом исследования послужил набор данных, который мы собрали и опубликовали на платформе Figshare [Firsanova, 2020]. Figshare — это открытый репозиторий для распространения научно-исследовательских материалов, таких как изображения, наборы данных, аудио и видео. Данные доступны для скачивания всем желающим. Набор регулярно обновляется и дополняется каждые три месяца.

Разработка материала исследования представляет собой индивидуальный проект, то есть такие процессы, как собирание материала, компиляция набора, обработка и публикация данных выполняются нами самостоятельно. На данный момент для пользователей доступно пять версий набора, которые были получены в результате реализации подхода к построению диалоговых систем, в основе которого лежит работа с данными. Описание разновидностей набора и наши предпосылки к их созданию представлены в разделе 3.2 «Работа с данными».

## 2.1. Источник материала

Источником материала исследования стали тексты, опубликованные на информационном сайте о расстройствах аутистического спектра Autistic City. Данный ресурс расположен по адресу <https://www.aspergers.ru/>*.* Использование материалов сайта согласовано с администрацией ресурса в личной переписке. В соответствии с информацией на портале, ресурс представляет собой коллекцию переводов онлайн-публикаций и статей, рассказов и эссе людей с аутизмом, оригинальных исследований и других материалов, связанных с расстройствами аутистического спектра.

Некоторые из материалов сайта Autistic City созданы людьми с синдромом Аспергера, высокофункциональным аутизмом (в этом случае интеллект сохранен). Принимая во внимание этот факт, одиннадцатилетнюю историю портала, а также его популярность, мы считаем, что материалы, представленные на сайте Autistic City, могут служить ценным и надежным источником для построения систем для поддержки инклюзии.

Рисунок 4 демонстрирует тематическое разнообразие текстов, использованных для построения набора данных. Каждый текст на портале Autistic City снабжен набором тегов, например, *«самодиагностика», «личный опыт», «мероприятия»*. Темы для нашего исследования определялись в соответствии с наборами тегов. Выделенные темы представляют обобщенную информацию о содержании текстов сайта. Эта информация приводится в качестве иллюстрации нашей выборки для того, чтобы было удобнее ориентироваться в материале.

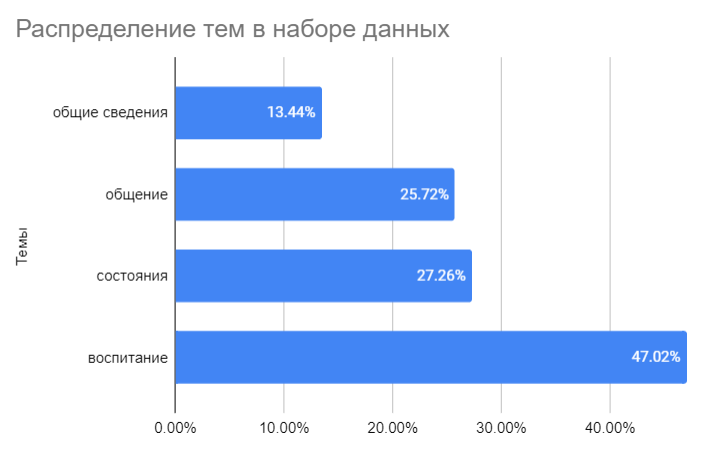


Рисунок 4. Тематическое распределение текстов в наборе данных

Как показывает диаграмма на Изображении 4, почти половину текстов составляют материалы о воспитании ребенка в спектре аутизма. Эти тексты содержат советы для родителей об особенностях обучения и физического воспитания ребенка в спектре, а также информацию о профилактике типичных для аутизма состояний, таких как, нарушения сна или сенсорная перегрузка. Около 30% текстов описывают состояния, характерные как для детей, так и для взрослых в спектре аутизма (тревожность, паническая атака, сенсорная перегрузка) и предлагают некоторые решения по их преодолению. Приблизительно такую же долю текстов составляют материалы о коммуникации с людьми в спектре, например, о том, как заводить отношения и дружбу. Менее 15% текстов содержат общие сведения об аутизме — определение расстройства, его характеристики. Имплицитно эта информация содержится во всех текстах, однако в эти 15% вошли только познавательные тексты без советов и предложений.

Тексты для набора данных отбирались вручную на основании нескольких критериев. В приоритете были материалы, описывающие международный или российский опыт людей с расстройствами аутистического спектра. Тексты, переведенные с иностранного языка и описывающие опыт жизни людей с аутизмом, не актуальный для России, нами не использовались.

Мы не можем предсказать, как пользователь будет реагировать на ситуации, когда модель выводит тексты от первого лица. Поэтому мы решили избежать сценариев, когда система может «выучить» такие языковые шаблоны, а именно отказались от использования текстов от первого лица, например, от некоторых эссе с описанием личного опыта. Также при составлении набора данных мы не использовали тексты, описывающие психиатрические методики, так как извлечение отрывков из материалов на медицинскую тему чревато искажением информации и порождением ложных фактов.

## 2.2. Структура набора данных

Структура набора данных, используемого в настоящей работе [Firsanova, 2020], разрабатывалась нами по образцу второй версии Стенфордского вопросно-ответного набора данных (Stanford Question Answering Dataset 2.0, SQuAD 2.0) [Rajpurkar et al., 2018], первая версия которого уже упоминалась нами в разделе 1.2 «Современные модели и трансферное обучение». Набор SQuAD 2.0 содержит короткие тексты, вопросы к ним и ответы (метки для извлечения выдержек из текстов, которые отвечают на соответствующий вопрос), а также метки соответствия вопроса содержанию текста, которых не было в первой версии SQuAD [Rajpurkar et al., 2016]. Рассмотрим структуру нашего набора и его отличия от стенфордского.

Наш набор данных представляет собой объект JavaScript Object Notation (JSON), легкий для чтения человеком и удобный для обработки на языке программирования Python. Объект является словарем (англ., *dictionary*; на Рисунке 5 этот тип данных обозначен пометой *dict*), который включает два базовых элемента: (1) набор вопросно-ответных пар в форме словаря (*qa\_pairs* на Рисунке 5) и (2) абзацы для машинного чтения строкового типа (*context* на Рисунке 5; строковый тип обозначен пометой *str*, англ., *string*). Базовые элементы выделены на Рисунке 5 красным цветом.

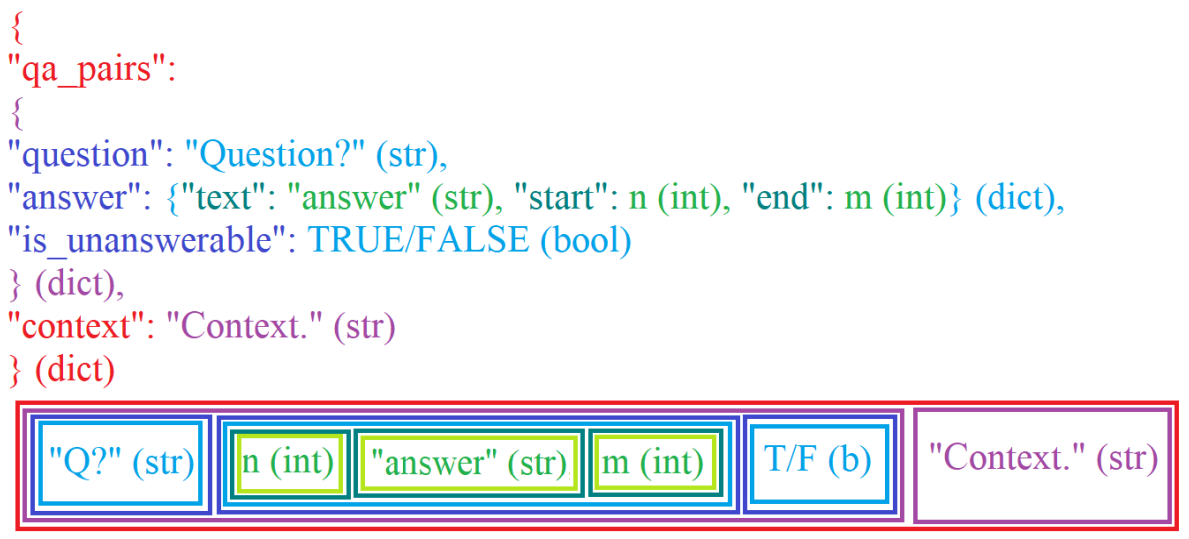


Рисунок 5. Структура набора данных [Firsanova, 2021 (C)]

Фиолетовым цветом на Рисунке 5 выделены элементы подсловаря *qa\_pairs*, который мы использовали для представления вопросно-ответных пар в наборе. Подсловарь *qa\_pairs* состоит из трех пар типа «ключ — значение». Первый элемент подсловаря содержит вопрос строкового типа (*question* на Рисунке 5). Второй элемент представляет собой еще один словарь, который содержит ответ на вопрос строкового типа, а также пометы его границ в абзаце для чтения *context*. Границы ответа обозначены двумя целыми числами — порядковыми номерами первого и последнего символа соответствующего отрывка в тексте для чтения (*answer* на Рисунке 5, целочисленный тип обозначен пометой *int*, англ., *integer*). Необходимость добавления помет начала и конца ответа обоснована нами в разделе 1.3 «Базовые модули диалоговых систем».

Третьим элементом является метка соответствия вопроса теме текста (*is\_unanswerable* на Рисунке 5), обозначенная булевой переменной, то есть принимающая одно из двух значений — *«истина»* (англ., *true*) или *«ложь»* (англ., *false*). Булевы переменные обозначены на Рисунке 5 пометой *bool* (англ., *Boolean*)*.* Эта часть нашего набора данных существенно отличается от аналогичного элемента во второй версии стенфордского набора [Rajpurkar et al., 2018]. В стенфордском наборе под вопросами, на которые невозможно ответить, подразумевались релевантные единицы, соответствующие теме текста. Однако напрямую информация для ответа на такие вопросы в текстах не представлена. На них можно дать лишь приблизительный ответ.

В отличие от разработчиков стенфордского набора, мы намеренно снабдили 5% наших данных нерелевантными вопросами, которые не соответствуют теме текста. Наша идея заключалась в том, чтобы «запутать» модель, научить ее выделять признаки «отвлекающих» вопросов, не относящихся к предметной области «расстройства аутистического спектра». Пользователи чат-ботов склонны начинать игру с моделями, делать запросы с целью развлечения, а не получения информации. Такие сценарии нежелательны в работе социально-медицинских систем, поскольку их цель — информировать без искажения фактов, а не развлекать.

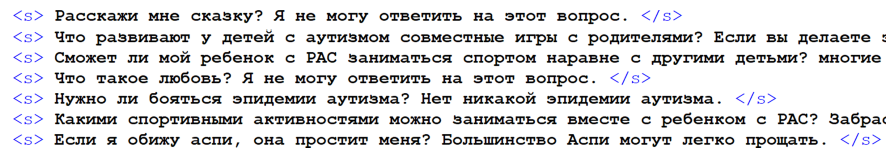


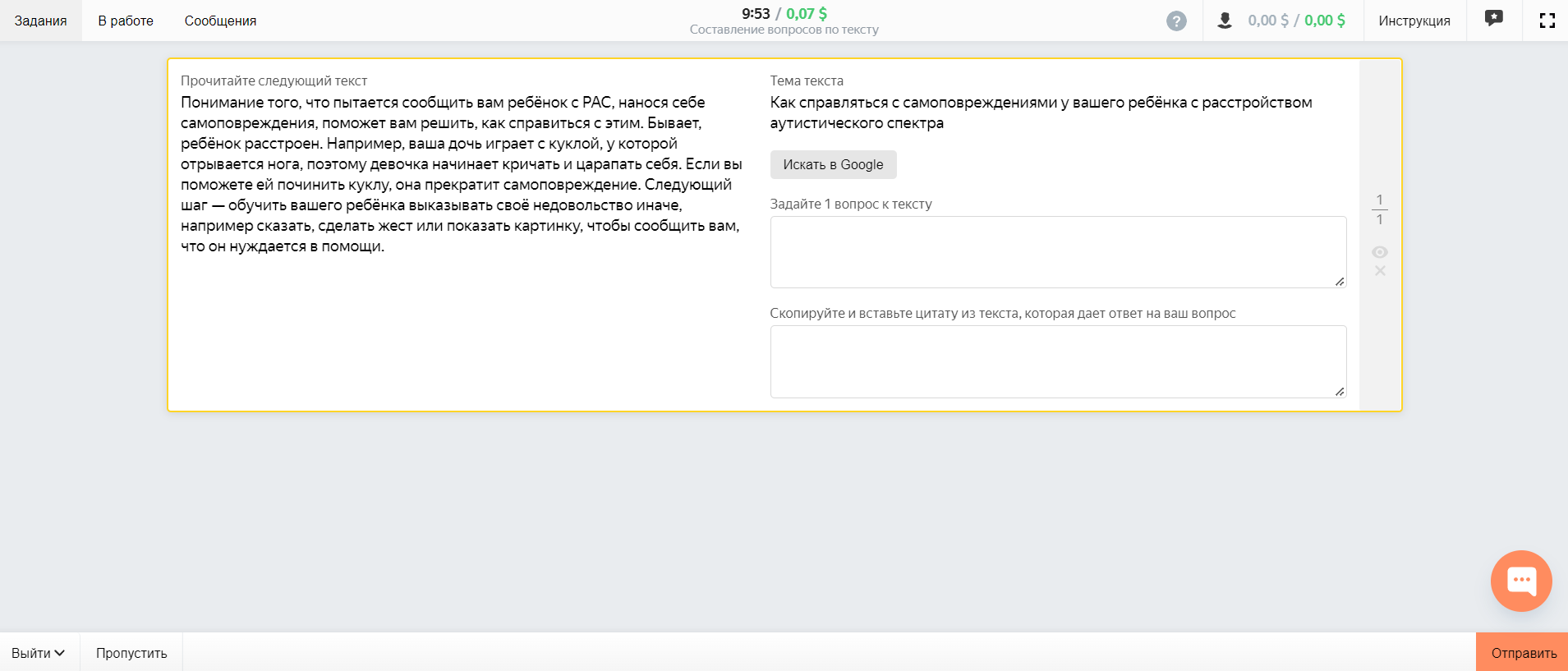
Рисунок 6. Структура набора данных, адаптированная для обучения генеративных систем

Для обучения генеративной модели мы привели структуру набора к строковому типу вида «вопрос — ответ». Иллюстрация измененного формата приведена на Рисунке 6. Из набора были извлечены вопросно-ответные пары без меток и контекстов. Мы разделили вопросно-ответные пары метками начала и конца предложения *<s>* и *</s>* соответственно. Вопросы с истинным значением метки *is\_unanswerable* мы снабдили ответом-шаблоном *«Я не могу ответить на этот вопрос».*

Образец набора данных представлен в Приложении А. Первую тысячу вопросно-ответных пар для набора данных мы составили самостоятельно. Затем мы запустили краудсорсинг. Информация о деталях реализации краудсорсингового проекта приведена во введении к настоящей работе. В следующем разделе мы рассмотрим процесс краудсорсинга, приведем наши наблюдения и текущую статистику набора данных.

## 2.3. Краудсорсинг

Сбор данных методом краудсорсинга осуществлялся на платформе Яндекс.Толока. Образец интерфейса платформы приведен на Рисунке 7. Тексты заданий для краудсорсинга представлены в Приложении Б. Исполнителям краудсорсинговой платформы предлагалось выполнить два задания, целью каждого из них являлся сбор данных для составления обучающих наборов для диалоговых систем, которые смогут найти применение в сфере поддержки инклюзии людей с расстройствами аутистического спектра.

 Рисунок 7. Интерфейс платформы для краудсорсинга и образец одного из заданий

Прежде чем запустить краудсорсинговую разметку, мы самостоятельно выполнили собственные задания, составили 1134 вопросно-ответных пары для набора данных. Такое тестирование заданий позволило внести некоторые изменения в инструкции для исполнителей краудсорсинга и запустить разметку, принимая во внимание личный опыт. Мы вывели критерии отбора текстов для разметки, которые описаны в разделе 2.1 «Источник материала», и сформулировали правила решения заданий, которые приведены в Приложении Б.

В первом задании разметчикам предлагалось составить вопросы к коротким текстам о расстройствах аутистического спектра и найти ответы в тех же отрывках. Максимальный объем текстов для исполнителей составлял 512 символов, поскольку наш набор создавался для моделей подобных BERT [Devlin et al., 2019], которые в экстрактивных диалоговых системах обычно принимают на вход последовательности именно такой длины. Детали настройки и использования моделей приведены в разделе 3.1 «Работа с моделью». Мы вручную сокращали тексты для разметки до нужной длины таким образом, чтобы отрывки оставались осмысленными и информативными. Пример отрывка приведен на Рисунке 7 в блоке «Прочитайте следующий текст».

Второе задание предназначалось для составления нерелевантных пользовательских запросов, не соответствующих нашей предметной области, то есть созданию материала для 5% набора данных с истинным значением метки *is\_unanswerable*. Значение этой метки объясняется в разделе 2.2 «Структура набора данных». Разметчикам предлагалось составить 10 вопросительных предложений на такие темы, как «искусственный интеллект», «приветствие», «прощание», «знакомство», «чат-боты» и другие. Мы выбрали перечисленные темы исходя из личного опыта взаимодействия с чат-ботами, наблюдениями блоггеров [FAQ Bot, 2020] и существующей документации для диалоговых систем [Microsoft, 2022].

Интерфейс задания создавался в конструкторе платформы Яндекс.Толока. Конструктор позволяет заказчикам прописывать конфигурации в формате JavaScript Object Notation (JSON). В первом задании входные данные содержали тексты для составления вопросов, заголовки или темы (например, *«нарушения сна у детей в спектре аутизма»*), а также кнопку для поиска в интернете с заранее прописанным поисковым запросом (например, *«сон дети аутизм»*).

Мы обеспечили интерфейс задания заголовками и кнопками поиска для того, чтобы исполнители могли ориентироваться в терминологии текстов об аутизме и в один клик получить дополнительную информацию. Выходные данные (так в конфигурации платформы Яндекс.Толока называются поля для решений исполнителей) включали строки для ввода составленного вопроса и найденного ответа.

Во втором задании входные данные содержали список тем для составления вопросов, одинаковый для всех исполнителей. Выходные данные этого задания включали десять строк для ввода составленных вопросов. Все данные краудсорсинга, а также результаты разметки исполнителей хранились в файлах с расширением Tab Separated Value (TSV).

На данный момент набор данных содержит 577 текстов для машинного чтения и 3314 вопросно-ответных пар. Набор, описанный в Главе 2, динамический: мы меняем его структуру по результатам экспериментов и постоянно увеличиваем объем. Сбор материала для обучающего набора продолжается. Расширение базы данных запланировано нами и на следующий год, поскольку работа вызвала общественный резонанс. Например, набор данных победил в конкурсе на лучший датасет сообщества Open Data Science по итогам 2020 года [Data Ёлка 2020, 2020].

# Глава 3. Подходы к построению системы

Набор данных, подробно описанный в Главе 2 настоящего исследования и в одной из наших статей [Firsanova, 2021 (C)], использовался для проведения экспериментов по реализации двух базовых подходов к построению систем искусственного интеллекта и диалоговых моделей в частности. Мы использовали терминологию, подробно описанную Эндрю Ыном в его докладе для мероприятия DeepLearning.AI [Ng, 2021]. Рассмотрим два подхода к построению интеллектуальных систем, которые он описал в своем выступлении.

Первый подход предполагает разработку алгоритма и настройку моделей машинного обучения (англ., *model-centric approach*). Это сугубо математический подход, в котором данные используются в качестве материала для расчета производительности модели. В основе второго подхода — преобразование обучающих данных (англ., *data-centric approach*). Выбранная модель при этом не изменяется. Допускаются минимально необходимые манипуляции, например, настройка параметров обучения. Суть этого подхода заключается во множественных операциях со структурой, дизайном и объемом обучающих данных [Ng, 2021].

По словам Эндрю Ына, 99% современных исследований посвящены реализации первого, математического, подхода. Однако фактически 80% работы с моделями машинного обучения представляют собой обработку данных, поэтому использование второго подхода (обработка данных) часто позволяет добиться повышения производительности более чем на 6% в сравнении с математическим подходом. К минусам второго подхода относится недостаточность методологического аппарата ввиду слабой изученности. Также для реализации этого подхода необходимо иметь хотя бы одну испытанную и доказавшую свою надежность модель.

Мы решили последовательно применить оба подхода к построению систем. Работа проводилась в два этапа. На первом этапе мы настроили несколько моделей с помощью математического подхода и выбрали наиболее производительные модификации, а на втором — провели несколько экспериментов с обучающими данными.

## 3.1. Работа с моделью

Для математических экспериментов в области разработки моделей искусственного интеллекта мы выбрали построение вопросно-ответных систем с использованием экстрактивного и генеративного подходов. Обоснование нашего выбора и особенности экстрактивного и генеративного алгоритмов описаны в разделе 1.3 «Базовые модули диалоговых систем». Рисунки 2 и 3 иллюстрируют схему реализации этих подходов при построении вопросно-ответных систем.

Для осуществления генеративного и экстрактивного подхода мы применяли методы трансферного обучения. Концепция трансферного обучения описана в разделе 1.2 «Современные модели и трансферное обучение». Мы использовали только предобученные модели с архитектурой Transformer, которая описана в разделе 1.4 «Архитектура Transformer». В работе применялись следующие техники трансферного обучения:

1. тонкая настройка: настройка параметров обучения (англ., *fine-tuning*);
2. дообучение модели с использованием новых наборов данных;
3. добавление нового слоя для решения последующей (англ., *downstream*) задачи к первоначальной структуре нейросетевой модели (только для экстрактивного подхода);
4. преобразование формальной структуры входных данных (только для генеративного подхода).

Обучение моделей производилось в коллаборативной среде программирования Google Colab с использованием графического процессора (англ., *graphics processing unit*) NVIDIA Tesla K80, удаленный доступ к которому мы получили безвозмездно в рамках бесплатных возможностей платформы Colab. Графический процессор позволяет производить энергоемкие математические операции, с которыми может не справиться процессор среднестатистического персонального компьютера. Например, на тонкую настройку и дообучение одной модели машинного мы тратили от 15 до 30 минут в зависимости от количества параметров и объема обучающей выборки.

Для создания программного кода мы использовали инструменты репозитория HuggingFace [Wolf et al., 2020]. Эта платформа представляет собой площадку для энтузиастов в сфере разработки моделей с архитектурой Transformer. На сайте HuggingFace можно найти обучающие материалы для исследователей, наборы данных, примеры готовых решений таких популярных задач, как анализ тональности текста, а также коллекцию инструментов для предварительной обработки данных (например, токенизации), обучения и оценки моделей. Любой разработчик может внести свой вклад в сообщество и загрузить на сайт собственную модель.

Наш программный код написан на языке Python 3.7. Код представлен в нашем репозитории https://github.com/vifirsanova/ASD-QA [GitHub, 2021]. Значения метрик оценки моделей в описании репозитория могут отличаться от результатов работы систем, описанных в данном исследовании. Причина заключается в том, что в ходе исследования мы провели множество экспериментов с разными значениями параметров обучения моделей.

Кроме того, системы машинного обучения отличает нестабильность, вызванная большим количеством параметров и связей между нейронами. Повлиять на результат могут также особенности работы алгоритмов, например, механизмы случайной генерации чисел. Все это приводит к тому, что значения метрик оценки будут различаться, даже если одну и ту же модель обучить несколько раз, не изменяя параметры.

Рассмотрим программное обеспечение системы. Для предварительной обработки данных мы использовали библиотеки JSON и Scikit-Learn. Библиотека JSON необходима для работы с форматом JavaScript Object Notation, в котором хранился наш набор данных (набор описан в Главе 2 «Материал исследования»). Библиотека Scikit-Learn использовалась для разделения набора данных на три выборки: обучающую (70% от общего количества вопросно-ответных пар), проверочную (15%) и тестовую (также 15%).

Для токенизации данных и их преобразования к виду векторов мы использовали токенизаторы из библиотеки HuggingFace. Для сборки данных, подготовленных для машинной обработки, мы использовали библиотеку PyTorch. Предварительно обученные модели загружались из репозитория HuggingFace. Обучение производилось с помощью PyTorch на графических процессорах. Для извлечения результатов использовалась библиотека PyTorch. Скрипты для подсчета метрик оценки производительности систем мы прописывали вручную. Для обучения генеративной модели дополнительно было проведено несколько экспериментов с использованием пакета gpt\_2\_simple [Woolf, 2019] для упрощенного и ускоренного дообучения GPT-2.

В Таблице 1 приведены конфигурации выбранных для исследования моделей. Для реализации экстрактивного подхода мы выбрали несколько систем. Во-первых, использовался BERT [Devlin et al., 2019]. Эта модель подробно описана в разделе 1.2 «Современные модели и трансферное обучение».

Во-вторых, использовалась дистиллированная, то есть облегченная версия BERT под названием DistilBERT. Дистилляцией знаний в машинном обучении называют процесс разработки компактных моделей, производительность которых (измеряемая метриками оценки) не будет уступать структурно аналогичным, но требующим больших вычислительных мощностей системам. В процессе дистилляции модель учится повторять «поведение» существующей системы, которая занимает больший объем памяти компьютера [Sanh et al., 2019, с. 2].

В-третьих, мы использовали кросс-лингвистическую модель XLM-RoBERTa, которая обучалась на материале текстов на ста языках [Conneau et al., 2019, с. 1]. В основе XLM-RoBERTa — архитектура, аналогичная BERT. Мета-обучение, описанное в разделе 1.2 «Современные модели и трансферное обучение», помогает кросс-лингвистическим моделям анализировать структуры типологически различных языков и таким образом добиваться больших значений метрик оценки производительности, чем аналогичные системы, обученные для одного языка и дообученные в процессе трансфера знаний для другого.

Таблица 1. Конфигурации моделей [Firsanova, 2022]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | BERT | DistilBERT | XLM-RoBERTa | GPT-2 |
| Значение исключения (англ., *dropout*) | 0.1 | | | |
| Функция активации | GELU | | | |
| Количество скрытых слоев | 12 | 6 | 12 | 12 — версия HuggingFace  24 — версия gpt\_2\_simple |
| Количество эмбеддингов | 512 | | | 768 |
| Количество голов внимания | 12 | | | 12 — версия HuggingFace  20 — версия gpt\_2\_simple |
| Размер обучающего словаря | 30 522 | | | 50 257 |
| Количество параметров | 110 миллионов | 66 миллионов | 125 миллионов | 117 миллионов — HuggingFace,  774 миллиона — gpt\_2\_simple |

В качестве генеративной модели мы использовали две версии GPT-2 [Radford et al., 2019]. Первая версия обучалась с помощью библиотеки HuggingFace. Для этой модели мы изменили конфигурации в соответствии с доступными нами вычислительными мощностями. Вторая версия обучалась с помощью пакета gpt\_2\_simple. Этот пакет для упрощенного использования GPT-2 позволил нам для сравнения протестировать более производительный вариант генеративной модели. Влияние характеристик перечисленных нами моделей на производительность экстрактивных и генеративных диалоговых систем приведено в разделе 3.3 «Оценка системы» настоящей главы.

Рассмотрим выбранные нами конфигурации моделей. Значения конфигураций были выбраны нами в результате множественных экспериментов, которые мы описывали в своих статьях [Firsanova, 2022]. Для каждой системы мы задали значение дропаута (или исключения) равное 0.1. Дропаут — это механизм для предотвращения переобучения, то есть такой ситуации, при которой модель выучила «наизусть» данные обучающей выборки и более не способна переносить знания на новые материалы. Искусственным образом в процессе обучения нейросети исключается определенная доля нейронов (в случае наших экспериментов эта доля составила 0.1), за счет чего создаются новые сети (с исключениями). В конце обработки модель использует усредненный результат обучения каждой из полученных таким образом нейросетей [Коротков, Шамурина, 2020, с. 61].

В качестве функции активации мы выбрали Gaussian error linear units (GELU) [Hendrycks, Gimpel, 2016]. Функция GELU имеет отрицательный коэффициент и небольшую выпуклость, что делает ее более покатой, чем не менее популярная в машинном обучении Rectifier linear unit (ReLU) [Ramachandran et al., 2017, с. 7]. Эта особенность делает GELU устойчивой к отрицательным значениям, то есть к выдачам модели машинного обучения, которые можно интерпретировать следующим образом: «Выбранные данные N с наименьшей вероятностью относятся к классу X». Сегодня GELU считается передовой в обработке естественного языка.

Кроме того, мы настраивали количество скрытых слоев нейронной сети (блоков обработки данных) и эмбеддингов (языковых признаков, используемых моделью). Также настраивалось количество голов внимания (определение этого термина дано в разделе 1.4 «Архитектура Transformer» Главы 1), размер обучающего словаря (количество n-грамм, полученных в результате токенизации) и количество параметров нейросетевой модели, то есть ее размер.

Таблица 2. Настраиваемые параметры моделей, описаны в статье [Firsanova, 2021 (D)]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | BERT | DistilBERT | XLM-RoBERTa | GPT-2 |
| Время обучения | 20 | 10 | 10 | 30 |
| Количество эпох | 10 | 20 | 10 | 30 |
| Скорость обучения | 3е-5 | 1е-5 | 3е-5 | 1е-4 |
| Размер батча | 1 | | | |
| Температура |  | | | 0.7 |
| Top K |  | | | 40 |

Для перечисленных моделей мы провели процедуру тонкой настройки с выбором оптимального времени обучения, количества эпох, скорости обучения, а также размера блоков данных, подаваемых на каждом шаге обучения (размером батча). Для генеративных моделей подбирались также оптимальные значения температуры (доля случайных выдач; чем выше значение температуры, тем разнообразнее генерируемый текст) и параметра Top K (количество слов, которое модель анализирует на каждом шаге обучения). Результаты тонкой настройки представлены в Таблице 2. Модели дообучались на нашем наборе данных, описанном в Главе 2 «Материал исследования».

Модели для реализации экстрактивного подхода были снабжены дополнительным («верхним») слоем для извлечения информации из текста. Выбранные нами экстрактивные модели предназначены для решения двух задач: классификации предложений и маскированного моделирования языка. Напрямую, без модификаций, извлекать ответы на вопросы такие модели не могут (хотя, по нашему мнению, маскированное моделирование языка имплицитно дает моделям знания для решения этой задачи). Однако добавление слоя, который картирует позиции токенов начала и конца ответа на вопрос в тексте из обучающей выборки на выученные нейронной сетью признаки, решает эту проблему.

Для реализации генеративного подхода мы преобразовали структуру обучающей выборки к виду строк типа «вопрос — ответ» (иллюстрация приведена на Рисунке 6 в Главе 2 «Материал исследования»). Входные данные для генеративной модели требовали только вопроса, который выполнял функцию префикса для генерации текста, в то время как экстрактивные модели требовали также наличие контекста для извлечения информации.

В ходе работы с моделями мы изучили содержание репозиториев с кодом наших базовых моделей: BERT [Devlin et al., 2019], XLM-RoBERTa [Conneau et al., 2019], DistilBERT [Sanh et al., 2019] и GPT-2 [Radford et al., 2019]. Разработчики архитектур, в основе которых лежит кодировщик BERT (BERT, XLM-RoBERTa и DistilBERT), кроме первоначальных версий моделей публикуют также некоторые модификации, различающиеся по размеру (по количеству параметров и объему обучающих данных) и разнообразию языков, с которыми программа теоретически может справиться. В репозитории HuggingFace [Wolf et al., 2020] можно найти адаптации перечисленных моделей под конкретные задачи или определенные наборы языков.

Приведем обоснование выбора модификаций для экстрактивных моделей. Исходя из наших потребностей, мы отказались от тестирования больших (англ., *large*) и малых (англ., *small*) версий моделей с кодировщиком BERT. Ввиду небольшого объема доступных нам обучающих данных (то есть объема набора, описанного в Главе 2 «Материал исследования»), мы не ожидали значительного влияния размера модели на ее производительность.

Кроме того, тестирование этого критерия не входило в наши планы, так как в центре нашего внимания — исследование влияния лингвистических особенностей текста на производительность интеллектуальных систем. Вместо этого мы акцентировали внимание на языковом разнообразии доступных моделей с кодировщиком BERT. Для этого мы сравнили производительность моделей, настроенных на работу с русскоязычным, многоязычным и кросс-лингвистическим материалом.

В качестве русскоязычной модели мы использовали ruBERT — BERT для русского языка, модель из репозитория HuggingFace. В качестве многоязычных модификаций мы использовали mBERT и mDistilBERT — версии BERT и DistilBERT, обученные на полиязыковом материале, они представлены разработчиками моделей. В качестве кросс-лингвистической версии мы использовали XLM-RoBERTa без модификаций, так как эта модель изначально разрабатывалась для кросс-лингвистической обработки текстов.

Поскольку наши эксперименты проводились до презентации русскоязычной версии GPT-2 [SmartMarket, 2022], у нас не было возможности исследовать влияние лингвистического разнообразия обучающих данных на производительность генеративной модели. В связи с этим, мы решили восполнить недостаточность изучения влияния размера модели на производительность систем и провели несколько экспериментов с GPT-2 с разным количеством параметров обучения (117 и 774 миллиона параметров).

Итак, у нас была возможность изучить влияние языкового разнообразия моделей на производительность систем на примере экстрактивных архитектур. Влияние размера моделей на производительность мы проверяли на генеративных архитектурах. Мы находим избыточным исследовать влияние двух критериев (размер и языковое разнообразие) на генеративные и экстрактивные системы одновременно, потому что в основе всех типов моделей в наших экспериментах лежит одна базовая архитектура — Transformer. В конечном итоге, все наши опыты сводятся к исследованию Transformer (ознакомиться подробнее с этой архитектурой можно в разделе 1.4 «Архитектура Transformer»).

## 3.2. Работа с данными

Следующим этапом нашей работы было преобразование обучающего набора данных с целью повышения производительности экстрактивных и генеративных моделей. В результате экспериментов создано пять версий набора данных, описанного в Главе 2 «Материал исследования». Все модификации в открытом доступе расположены в репозитории проекта на платформе FigShare [Firsanova, 2020]. Рассмотрим каждую модификацию набора и предпосылки к ее созданию.

Оригинальная версия набора данных проиллюстрирована в Приложении А. Каждому вопросу в первоначальной версии набора соответствует только один ответ. Средняя длина вопроса составляет 7 слов, или 47 символов. Средняя длина ответа — 16 слов, или 111 символов. Скрипты для подсчета статистики набора данных приведены в нашем репозитории https://github.com/vifirsanova/ASD-QA.

Первая модификация нашего набора допускает наличие нескольких ответов на один вопрос. Стенфордский вопросно-ответный набор [Rajpurkar et al., 2016], который послужил источником вдохновения для данной работы, также содержал несколько вариантов ответа. Ситуации, когда один вопрос может иметь несколько ответов, возникают в случаях, когда в тексте содержится перечисление или части предложения соединены сочинительной связью.

Например, в нашем наборе данных есть текст об особенностях фокуса у людей в спектре аутизма, в котором содержится следующее предложение: *«Все дело в том, что мне трудно воспринимать картину в целом, и вместо этого я постоянно сосредотачиваюсь на деталях…»*. Эта фраза может использоваться в качестве ответа на вопрос *«Как проявляется гиперфокус у людей в спектре аутизма?»,* по крайней мере, тремя способами: (1) *«трудно воспринимать картину в целом»*; (2) *«постоянно сосредотачиваюсь на деталях»*; (3) *«трудно воспринимать картину в целом, и вместо этого я постоянно сосредотачиваюсь на деталях»*.

Некоторые ответы содержат наречия со значением усиления и обороты, которые не несут смысловой нагрузки, но способствуют лучшему усвоению информации. Такие элементы можно опустить без потери смысла. Например, высказывание *«постоянно сосредотачиваюсь на деталях»* допускает преобразование к виду *«сосредотачиваюсь на деталях»*, при этом смысл текста не меняется. Так один вопрос может иметь два варианта ответа: развернутый и сокращенный.

Мы просмотрели порядка четырехсот вопросно-ответных пар (треть набора данных, составленного до запуска краудсорсинга) и добавили новые варианты ответов (не более трех, чтобы не перегружать систему) в соответствии с приведенными выше фактами. Модификацию с несколькими вариантами ответа мы назвали *multiple* (в переводе с английского *«множественный»*), под этим названием ее можно найти в нашем профиле на FigShare https://doi.org/10.6084/m9.figshare.13295831.

Вторую модификацию мы разработали на базе первой. В парах набора *multiple* с несколькими ответами мы выбирали самые короткие по количеству символов тексты и переносили их в новую выборку. Мы также просмотрели другие вопросно-ответные пары и сократили ответы там, где возможно. При этом каждый вопрос, как и в оригинальной версии набора данных, предполагал наличие только одного ответа. Средняя длина ответов в укороченной версии набора сократилась при этом с 16 до 14 слов, или со 111 до 92 символов.

Мы обработали треть первоначальной версии набора, созданной до запуска краудсорсинга. В профиле на FigShare вторая модификация представлена под названием *short* (переводится с английского как «короткий», в данном случае подразумевается «укороченный», то есть *shortened*).

Модификация *short* создана для того, чтобы узнать, как средняя длина ответа в обучающей выборке влияет на производительность модели. Мы предположили, что модель будет лучше справляться с короткими ответами из выборки, поскольку они содержат конкретную информацию, которую, по аналогии с поиском ключевых слов и тезисов, обнаруживать в тексте проще, в отличие от полных предложений с избыточными конструкциями. В то же время технически длинные ответы угадывать проще. Чем больше символов в тексте соответствует гипотетическому ответу, тем выше вероятность случайного угадывания правильного варианта.

Третья модификация набора создана, чтобы выяснить, как размер обучающей выборки влияет на производительность интеллектуальной системы. Размер обучающей выборки прямо пропорционален эффективности языковой модели. Об этом свидетельствует, например, эволюция модели GPT [Radford et al., 2018]. По мере увеличения количества параметров и объемов данных, заложенных в модель, GPT-2 [Radford et al., 2019] и GPT-3 [Brown et al., 2020] в значительной мере опережали предшествующие варианты как в значениях метрик оценки, так и в количестве решаемых задач. Увеличение размера выборки позволяет модели в процессе обучения подробно проанализировать структуру обучающих текстов, накопить знания о лингвистических признаках, характерных для данной предметной области.

Впрочем, увеличение объема выборки может повлечь за собой новые трудности, такие как генерация необъективной информации (англ., *bias*), вызванная неким социокультурным смещением. Например, в предметной области «расстройства аутистического спектра» сдвиг может быть вызван большим количеством текстов, в котором нейротипичные люди противопоставляются аутичным. В результате обучения на таких текстах модель может генерировать высказывания, которые отражают необъективную оценку поведения обеих групп людей.

Чтобы узнать, как отразиться изменение объема выборки на значениях метрик производительности, мы случайным образом исключили 50% вопросно-ответных пар из оригинальной выборки. Третью модификацию можно найти в нашем профиле FigShare под названием *half-sized* (название указывает на то, что размер выборки был сокращен вдвое).

Четвертая модификация затрагивает структуру набора данных. Разработчики стенфордского набора данных создали новую версию своего продукта [Rajpurkar et al., 2018], снабженную вопросами, на которые в текстах для машинного чтения ответов не было. Эта версия была разработана для адекватной оценки новых моделей, которые приблизились к человеческой производительности согласно метрикам. Объективно о таком качестве работы моделей мы говорить не можем, что указывает на несовершенство обучающих и проверочных выборок и мотивирует исследователей усложнять существующие наборы аналогичными методами.

Наш набор также содержит вопросы, которые система должна учиться игнорировать. Однако наша задача относится к сложной для обработки закрытой предметной области — «расстройства аутистического спектра». Поэтому ожидать производительность на уровне человеческой не имеет смысла. Тем не менее мы решили проверить, в какой мере наличие побочных вопросов усложняет задачу для экстрактивных и генеративных моделей. Для этого мы удалили из выборки все нерелевантные вопросы. В профиле FigShare новая модификация получила название *no\_impossible* (название указывает на отсутствие нерелевантных вопросов).

Описанные в настоящем разделе эксперименты мы представили на конференции NeurIPS 2021 [Firsanova, 2021 (A)]. В следующем разделе мы проанализируем результаты наших опытов. Мы представим значения выбранных метрик оценки в таблице, рассмотрим результаты человеческой оценки, а также постараемся проинтерпретировать работу исследуемых моделей с точки зрения науки о языке.

## 3.3. Оценка системы

Автоматическая оценка производительности вопросно-ответных систем является нетривиальной задачей. Использование таких широко распространенных в машинном обучении метрик, как полнота (англ., *recall*), точность (англ., *precision*) и F1-мера (англ., *F1-score*), предполагает наличие некого критерия для классификации данных, необходимого для выделения ложноположительных (англ., *false positives*) и ложноотрицательных (англ., *false negatives*) выдач моделей. Мы использовали решение, предложенное разработчиками стенфордского набора данных, которое описано в скрипте для оценивания обученных с помощью этого набора моделей [SQuAD Explorer, 2022].

Формулы (3)—(5) показывают, что изменениям подверглись точность и полнота . В качестве истинно положительных значений выступали токены, которые система предсказала верно, то есть элементы, встретившиеся, как в проверочном наборе, так и в выборке сгенерированных моделью ответов (обозначены как ). Ложноположительные значения были представлены в виде разницы между всеми предсказанными токенами и верно угаданными токенами. Ложноотрицательные значения представлялись как разница между токенами проверочной выборки и верно угаданными токенами. При этом -мера не подвергалась изменениям, производился расчет гармонического среднего.

Также мы провели ручную оценку производительности обученных систем по собственной методике. Необходимость в ручной оценке возникла в связи с тем, что исследуемые системы могут в будущем найти применение в социальной и медицинской сфере. Разработки в таких областях требуют особенной тщательности при тестировании. Автоматические метрики оценки далеки от совершенства: в процессе обучения система учится не решать задачу, а подстраиваться под заданные стандарты.

Некоторые выдачи обученной модели могут представлять собой набор удачно угаданных токенов, а не связный ответ на вопрос. Кроме того, некоторые вопросы в качестве ответа могут допускать варианты, не учтенные разработчиком набора данных. Такие случаи автоматическая система оценки может игнорировать. В связи с этим некоторые модели будут «недооценены».

### 3.3.1. Автоматические метрики оценки

Рассмотрим результаты работы моделей, обученных с использованием двух подходов, описанных в разделах 3.1 «Работа с моделью» и 3.2 «Работа с данными». Таблица 3 представляет значения метрик автоматической оценки моделей — полноты, точности и F1-меры. В левой части таблицы приведены результаты оценки моделей, настроенных в ходе работы с алгоритмами. В правой части представлены результаты настроенной XLM-RoBERTa после преобразования обучающего набора (в таблице название модели сокращено до XLM-R для удобства представления данных).

Оценка моделей производилась с помощью кода на основе скрипта разработчиков стенфордского вопросно-ответного набора [SQuAD Explorer, 2022]. Код можно найти в репозитории настоящего исследования https://github.com/vifirsanova/ASD-QA [GitHub, 2021].

Таблица 3. Результаты автоматической оценки [Firsanova, 2021 (A)]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Работа с моделью | | | | Работа с данными (на примере XLM-R) | | | |
| Модель | Точность | Полнота | F1 | Набор | Точность | Полнота | F1 |
| mBERT | 0.42 | 0.25 | 0.31 | multiple | 0.39 | 0.36 | 0.37 |
| ruBERT | 0.45 | 0.28 | 0.35 | short | 0.37 | 0.29 | 0.31 |
| mDBERT | 0.51 | 0.24 | 0.33 | no\_impos | 0.44 | 0.40 | 0.42 |
| XLM-R | 0.39 | 0.36 | 0.37 | half-sized | 0.72 | 0.04 | 0.07 |
| GPT-2 117 | 0.67 | 0.32 | 0.43 |  | | | |
| GPT-2 774 | 0.78 | 0.41 | 0.54 |

Среди экстрактивных моделей наиболее эффективной (в соответствии с метрикой F1-меры) оказалась кросс-лингвистическая модель XLM-RoBERTa. Мы выделили две причины такой высокой производительности этой архитектуры. Во-первых, XLM-RoBERTa является единственной моделью в нашей выборке, которая в процессе предварительного обучения рассчитывала веса признаков типологически различных языков. Благодаря мета-обучению модель XLM-RoBERTa обладает не только универсальными, но и специфическими знаниями о лингвистических структурах. Эта особенность позволяет нам использовать кросс-лингвистическую модель для решения задач на русском языке, относительно более сложном для обработки, чем, например, английский с простой морфологией и строгим порядком слов.

Многоязычные модели, такие как mBERT и mDistilBERT (в Таблице 3 модель mDistilBERT обозначена как mDBERT), также могут обладать такими знаниями. Однако следует учитывать, что эти модели заучивали языковые структуры в процессе дообучения, то есть их знания не являются базовыми, основополагающими в работе их алгоритмов.

Во-вторых, модель XLM-RoBERTa «больше» своих экстрактивных соперниц. Вторая часть названия модели XLM-RoBERTa указывается на ее робастность (англ., *robustness*), то есть устойчивость и производительность. В Таблице 1 приведено количество параметров каждой из выбранных нами моделей. Мы видим, что XLM-RoBERTa опережает по этому критерию все другие выбранные нами экстрактивные модели. Также она превосходит по количеству параметров малую версию генеративной GPT-2. Опыт разработки в сфере машинного обучения показывает, что количество параметров интеллектуальной модели прямо пропорционально ее производительности [Sutskever, 2019]. Наши опыты и результаты XLM-RoBERTa подтверждают заявления ученых.

Мы сравнили производительность русскоязычных и многоязычных моделей на примере двух версий BERT. Модификация, обозначенная в Таблице 3 как ruBERT, представляет собой модель BERT, дообученную на материалах русского языка. Значения F1-меры показывают преимущество лингвоспецифической модели: она на 0.04 единицы превосходит другие системы.

Этот результат позволяет нам сделать вывод о том, что выбор в пользу дообученных моделей для конкретного языка оправдан. Такая модель может содержать знания не только о структуре, но и о словарном составе анализируемого языка. Мы предполагаем, чем более конкретными знаниями о предметной области и языке обладает модель, тем выше будет ее производительность при решении специфических задач языковой обработки.

На примере экстрактивных моделей мы убедились в том, что количество параметров обеспечивает преимущество модели. Значения метрик оценки генеративных моделей — GPT-2 со 117 и 774 миллионами параметров — подтверждают это. Большая версия GPT-2 опережает малую не только по количеству параметров, но и по количеству скрытых слоев и голов внимания. Эти характеристики позволяют назвать большую GPT-2 более глубокой моделью, что подтверждается ее производительностью.

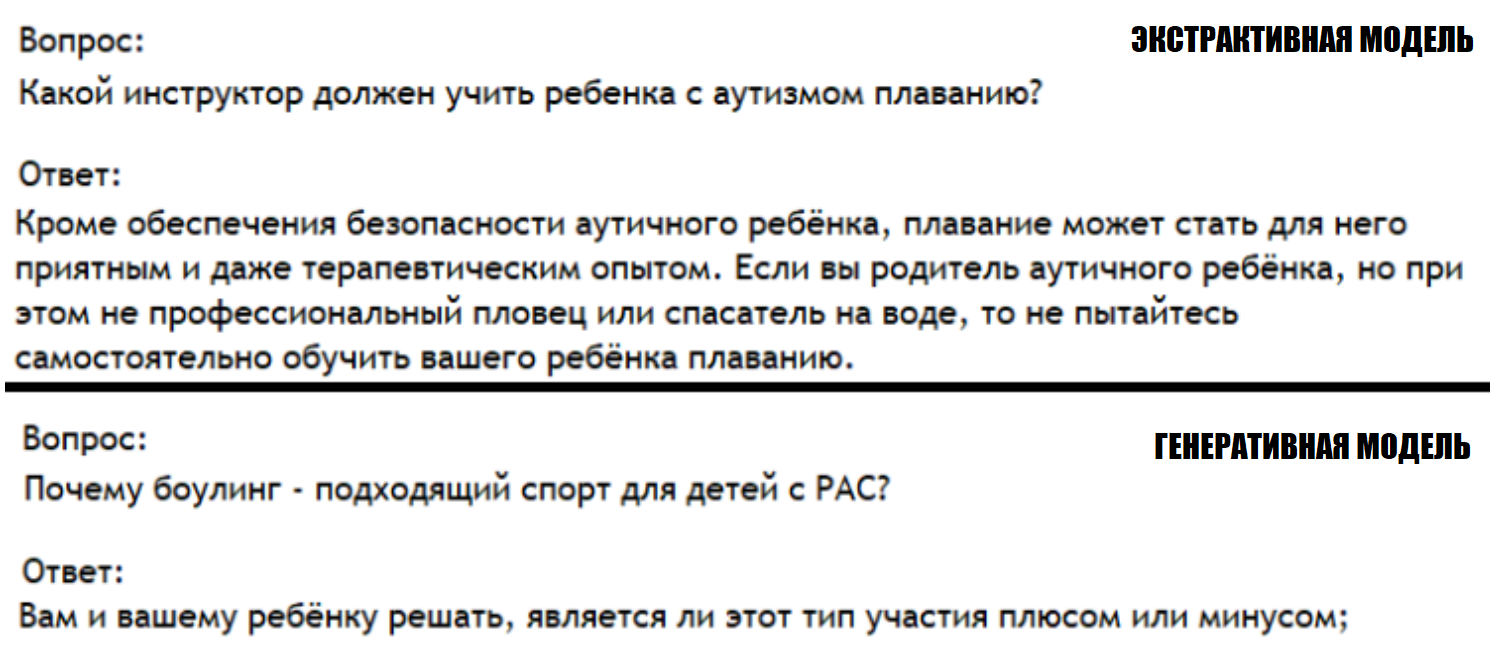


Рисунок 8. Образцы выдач генеративной и экстрактивной моделей

Значение F1-меры GPT-2 с 774 миллионами параметров больше на 0.09 единиц, чем у аналогичной модели со 117 миллионами параметров, что делает ее самой производительной в нашем исследовании. Однако практика показывает, что выдачи генеративных моделей не следует использовать в задачах реального мира. Рисунок 8 показывает образцы выдач экстрактивной и генеративной моделей. Мы видим, что генеративная модель склонна искажать информацию и порождать ответы на не связанные с вопросом пользователя темы.

Эффективность использования подхода, в основе которого лежит работа с данными, мы проверяли на модели XLM-RoBERTa. Результаты представлены в Таблице 3 в группе колонок «Работа с данными». При выборе модели для проверки этого подхода мы стремились найти баланс между максимально возможной производительностью модели и скоростью ее обучения, так как нам предстояло каждый раз заново дообучать выбранную модификацию на новом наборе данных.

Поскольку среднее время обучения генеративных моделей в нашем эксперименте оказалось почти втрое больше времени обучения экстрактивных модификаций, мы остановились на использовании наиболее производительной архитектуры с кодировщиком BERT — XLM-RoBERTa. Наши наборы описаны в разделе 3.2 «Работа с данными» (в Таблице 3 название набора *no\_impossible* сокращено до *no\_impos* для удобства визуального представления информации).

Использование модификации набора данных с несколькими вариантами ответа *multiple* не повлияло на производительность модели. Мы предполагали, что наличие альтернатив сделает задачу поиска ответа на вопрос несколько проще и приведет к более высоким значениям метрик оценки, однако ошиблись. Мы сделали вывод о том, что при составлении набора данных для нашей задачи достаточно остановиться на формате, в котором одному вопросу соответствует только один ответ, чтобы не тратить ресурсы на избыточную работу.

Набор с укороченными ответами *short* усложнил задачу для XLM-RoBERTa. Это подтверждает нашу гипотезу о том, что вероятность угадывания ответа системой прямо пропорциональна длине образцов из проверочной выборки. В то же время этот факт опровергает наше предположение о том, что наборы коротких ответов позволяют модели отыскивать ключевую информацию (например, именованные сущности, которые могут выступать в роли идентификатора местоположения ответа) и тем самым расширять базу знаний модели. Если нашей целью является наиболее точное воспроизведение образцов из проверочной выборки, предпочтение при составлении ответов обучающего набора стоит отдавать более длинным последовательностям.

Наибольших значений метрик оценки мы добились с набором без нерелевантных вопросов *no\_impossible*. Этот результат был предсказуем, поскольку набор нерелевантных вопросов мы составляли для того, чтобы усложнить задачу, научить модель распознавать пользовательские запросы, которые следует игнорировать. При использовании упрощенного набора производительность модели увеличилась на 0.04 единицы, согласно значениям F1-меры. Однако это сужает спектр возможностей модели. Мы не сможем предсказать реакцию обученного таким образом чат-бота на запросы, не соответствующие теме расстройств аутистического спектра.

Наконец, результаты работы модели, обученной на наборе *half-sized* (50% оригинального набора данных), показали, что размер обучающей выборки играет решающую роль в разработке диалоговых систем с нестандартным доменом. Модель показала высокую точность в сравнении с другими модификациями, однако полнота оказалась критически низкой — 0.04 единицы. Причина заключается в том, что большую часть ответов модель пропустила, то есть ее выдача была пустой.

Показатели F1-меры для других модификаций в среднем оказались в 5 раз больше значений модели, обученной на сокращенном вдвое наборе. Этот вывод представляет собой количественное доказательство необходимости регулярного расширения доступного нам набора данных.

### 3.3.2. Ручная оценка

Результаты автоматической оценкиразработанныхнамимодификаций показали, что генеративные системы являются более эффективным инструментом для разработки вопросно-ответных систем о расстройствах аутистического спектра. Однако на практике генеративные модели не пользуются популярностью среди разработчиков диалоговых систем. Память таких моделей весьма условна. Модель не обращается к существующему в неком цифровом пространстве хранилищу, но генерирует информацию, опираясь на выведенные ею в процессе обучения закономерности. Такой механизм ненадежен, он часто приводит к генерации ложных фактов.

Мы решили провести эксперимент по ручной оценке выдач генеративной и экстрактивной систем. Для этого мы составили анкету, образец которой представлен в Приложении В. Информантам предлагалось прочитать несколько ответов на вопросы и оценить их по пятибалльной шкале. При желании можно было прокомментировать свою оценку.

В опросе приняли участие 107 человек. Анкетирование производилось с помощью Google Forms. Вопросно-ответные пары представляли собой десять случайных выдач GPT-2 со 117 миллионами параметров и столько же выдач XLM-RoBERTa. Обе модели были дообучены на модификации набора данных *no\_impossible.*

Таблица 4. Результаты ручной оценки [Firsanova, 2021 (D)]

|  |  |
| --- | --- |
| Базовая модель | Оценка |
| GPT-2 117 (генеративный подход) | 46.00 |
| XLM-RoBERTa (экстрактивный подход) | 71.65 |

Мы подсчитали оценки респондентов и для наглядности перевели результаты в стобалльную систему. Оценки представлены в Таблице 4. Мы видим, что экстрактивный подход в значительной мере превосходит генеративный. Это противоречит значениям автоматической оценки. Опишем комментарии информантов.

По мнению участников эксперимента, большая часть ответов экстрактивной системы соответствует теме вопроса. Выдачи не нарушают нормы русского языка. При этом некоторые результаты содержат избыточное количество слов, что делает тексты тяжелыми для восприятия. Информанты перечислили недостатки метода на основе извлечения информации. Например, некоторые ответы показались респондентам логичными, но неблагозвучными. Такие выдачи нуждались в перефразировании или изменении порядка слов, что исключено в работе с моделью, которая настроена на дословное цитирование текста. Рисунок 9 иллюстрирует образец выдачи, который, на наш взгляд, обладает характеристиками, обозначенными информантами.

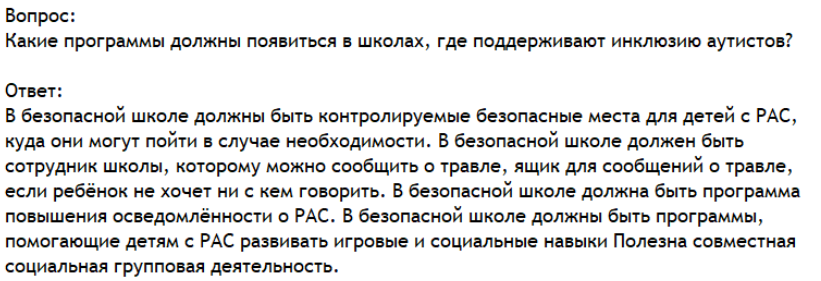


Рисунок 9. Типовой образец работы экстрактивной системы

Участники эксперимента считают выдачи генеративной системы неполными и вводящими в заблуждение. Многие ответы, по их мнению, не соответствуют темам, затронутым в вопросах, и звучат «размыто». Некоторые информанты характеризовали выдачи генеративной системы как бессмысленные. Рисунок 10 иллюстрирует одну из выдач, обладающую перечисленными свойствами.

Проведенный опыт позволяет сделать вывод о необходимости ручной оценки при разработке систем, которые могут найти применение в инклюзивной среде. Автоматические системы оценки не отражают реальное состояние системы. Это может привести к неправильному выбору базовой архитектуры для разработки программного обеспечения.

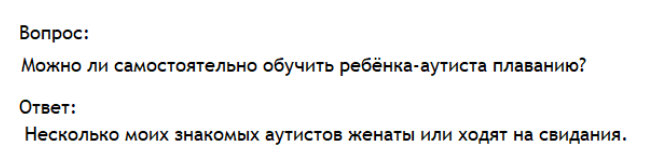


Рисунок 10. Типовой образец работы генеративной системы

Тем не менее оценивать каждую из разработанных модификаций вручную может быть ресурснозатратно, поэтому, на наш взгляд, оптимальным решением будет использование двухступенчатой оценки. На первом этапе мы предлагаем использовать автоматические системы оценки для отладки модификаций и выбора наиболее производительной модели. На втором этапе можно производить ручную оценку для выбора оптимальной системы, которая войдет в основу конечного продукта, например, чат-бота.

## 3.4. Влияние лингвистических особенностей материала

В процессе технической разработки архитектуры для диалоговой системы мы выделили три языковых признака, которые оказывают наибольшее влияние на работу интеллектуальных моделей. Проанализируем каждый из этих признаков, чтобы составить список методологических рекомендаций для разработчиков диалоговых систем, которые в будущем могут послужить инструментом для поддержки инклюзии людей с расстройствами аутистического спектра.

В качестве первого признака мы выделили средства интенсификации высказывания, то есть слова и выражения, не несущие информационной нагрузки, однако способствующие лучшему восприятию текста. К таким единицам мы относим усилительные наречия и частицы *(«очень», «сильно», «абсолютно», «даже»* и другие). Рисунок 11 иллюстрирует влияние средств интенсификации на работу экстрактивных и генеративных систем.

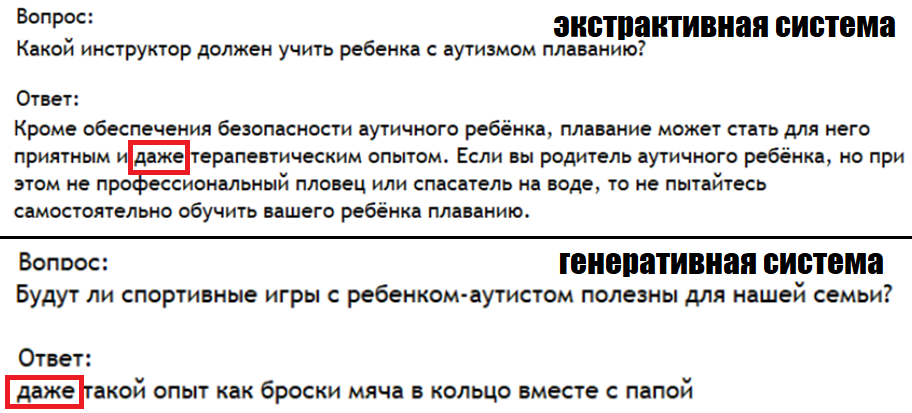


Рисунок 11. Влияние средств интенсификации на работу моделей

Мы проанализировали 20 случайных выдач наших систем и предположили, что оба типа моделей анализируют средства интенсификации в связке с правым (в случае генеративных моделей), либо с правым и левым (в случае экстрактивных моделей) контекстами. Модели используют эти слова в качестве индикаторов (или признаков) местоположения ответа на вопрос в текстах обучающей выборки. Наличие средств интенсификации в обучающей выборке окажет скорее благотворное влияние на производительность модели, вне зависимости от ее типа.

Однако в генеративных моделях средства интенсификации могут стать причиной проявления недостатков однонаправленных систем. Частица «*даже*» в примере на Рисунке 11 стоит на первом месте в выдаче. Предположительно, это связано с тем, что генеративная система в процессе обучения анализирует только правый контекст. Если генерация ответа в данном случае действительно «опиралась» на поиск средства интенсификации, то есть частицы «*даже*», то это объясняет генерацию нехарактерного для грамматики русского языка порядка слов. Это значит, что у разработчика вопросно-ответной системы в приоритете должно быть использование двунаправленных систем, таких как BERT.

Второй признак — наличие перечисления или сочинительной связи в предложениях. Рисунок 12 иллюстрирует образец ответа с использованием перечисления на примере работы экстрактивной системы. По результатам ручной оценки работы наших моделей, ответы систем, которые содержали перечисление каких-либо признаков, свойств или действий, оценивались информантами наиболее высоко. В среднем, порядка 60% респондентов оценивали такие выдачи на 4-5 баллов по пятибалльной шкале и характеризовали их как «исчерпывающие», «актуальные» и «грамматически корректные».

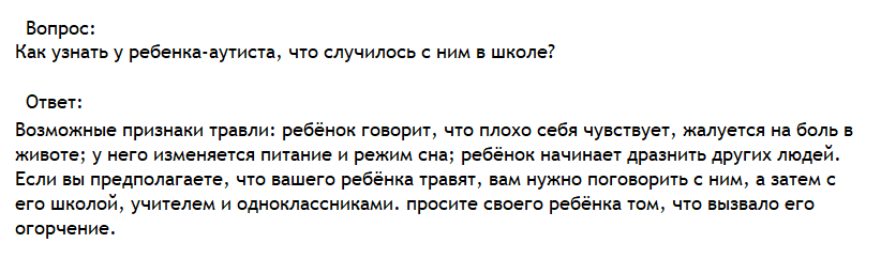


Рисунок 12. Образец ответа с перечислением

Ответы с перечислением или сочинительной связью в предложениях легко идентифицируются как экстрактивными, так и генеративными моделями, поскольку содержат характерную синтаксическую структуру. Архитектура Transformer, которую мы использовали для обучения наших моделей, справляется с последовательностями достаточной для анализа синтаксиса длины. Например, генеративные модели при генерации ответа учитывали контексты до 40 токенов (под токенами подразумеваются би- или триграммы, полученные методом подсловной токенизации).

Кроме того, наши модели учитывали порядок слов. Мы можем утверждать, что системы использовали наличие перечислений и сочинительной связи в предложениях в качестве ориентира для поиска ответов на вопросы. Также тексты с перечислениями охватывают большое количество тем, они могут использоваться для ответа на широкий спектр вопросов. Мы считаем использование предложений с перечислением одним из наиболее удачных решений в составлении текстов для обучающей выборки.

Третий признак — совпадение структуры вопросов и ответов. При составлении материалов для обучающего набора особое внимание уделялось структуре вопросов. Согласно комментариям наших информантов, которые оценивали системы вручную, некоторые вопросы из первоначальной (пробной) обучающей выборки были составлены некорректно. В частности, было отмечено, что ряд вопросов, например, *«Много ли мифов об аутизме существует?»*, *«Как узнать у ребенка-аутиста, что случилось с ним в школе?»* и другие,лишены конкретики, то есть в них не выражен определенный пользовательский запрос. Поэтому автоматическая генерация ответа на такие вопросы затруднительна.

Чтобы решить эту проблему, мы составили список требований для разметки набора данных. Требования заключаются в следующем:

1. соблюдение синтаксической структуры вопроса (наличие вопросительного слова, корректный порядок слов, соблюдение правил пунктуации);
2. наличие терминов и выражений, обозначающих тему текста (например, *«аутизм», «синдром Аспергера»*)
3. наличие идентификатора пользовательского запроса (например, указание конкретной проблемы, с которой могут столкнуться люди в спектре аутизма или их родные: *«травля в школе», «отказ от пищи»* и другие).

Итак, мы рассмотрели проведенные эксперименты по разработке генеративных и экстрактивных моделей, которые могут использоваться в качестве основы для построения вопросно-ответной системы о расстройствах аутистического спектра. Мы изучили некоторые аспекты разработки набора данных для такой системы с точки зрения технической реализации процесса, выяснили, как на производительность моделей влияет лингвистическая составляющая обучающей выборки. Итогом нашей работы стал комплекс методологических рекомендаций, который включает в себя основные выводы по каждому из этапов проделанной нами работы. Список рекомендаций представлен в Приложении Г.

# Заключение

Наше исследование представляет собой обзор существующих подходов к построению диалоговых систем для поддержки инклюзии людей в спектре аутизма. В основе исследования — эксперименты по разработке вопросно-ответных моделей с архитектурой Transformer. В ходе работы мы опирались на два распространенных подхода к разработке систем искусственного интеллекта.

В основе первого подхода лежит работа с математическими моделями, алгоритмами. В процессе его реализации мы экспериментировали с параметрами обучения моделей, исследовали возможности разных архитектур. Нам удалось сравнить характеристики генеративных и экстрактивных диалоговых систем.

Экстрактивные модели извлекают ответ из заданного контекста посредством поиска начала и конца отрывка, удовлетворяющего информационную потребность пользователя. Генеративные модели генерируют ответ на вопрос без опоры на контекст, учитывая лишь накопленные в процессе обучения знания. Мы выяснили, что экстрактивные модели проявляют себя как более предсказуемые и точные, чем их генеративные аналоги.

Разработка экстрактивных диалоговых систем представляет собой трудоемкий процесс, требующий доработки существующих архитектур посредством создания дополнительных слоев нейросетей, а также составления наборов данных, включающих не только вопросно-ответные пары, но и тексты для машинного понимания прочитанного, и метаданные с метками начала и конца ответа в контексте.

Разработка генеративных систем не требует дополнительных усилий — такие модели используют возможности мета-обучения, в частности, в процессе обучения они выводят некие языковые законы, применимые для решения новых задач без дообучения или изменения архитектуры. При этом обучение таких моделей не требует текстов для машинного чтения или особых меток. Системе достаточно принять на вход набор вопросно-ответных пар.

Однако при разработке систем для поддержки инклюзии в приоритете — точность и предсказуемость системы, а не эффективность или трудоемкость ее разработки. Наши эксперименты показали, что генеративные модели в процессе порождения текстов способны искажать информацию, заменяя темы из вопроса на другие, вероятно, более простые для машинной интерпретации. Такое «поведение» машины может привести к порождению ложных фактов и дезинформации. Мы пришли к выводу о том, что использование генеративных систем в реальной жизни возможно только в качестве эксперимента. При этом экстрактивные системы, на наш взгляд, следует использовать на практике.

В основе второго базового подхода в нашем исследовании лежит работа с данными, то есть с материалом для обучения интеллектуальных моделей. Наше исследование началось с работы над набором данных, предназначенным для построения информационных диалоговых систем о расстройствах аутистического спектра. Мы выбрали надежный информационный интернет-ресурс Autistic City, расположенный по адресу https://aspergers.ru, согласовали использование материалов с указанного сайта с его администрацией и собрали набор статей для обучающей выборки.

Структура обучающей выборки разрабатывалась по образцу стенфордского вопросно-ответного набора данных для машинного понимания прочитанного. Выборка включает наборы текстов для машинного чтения и вопросно-ответные пары. Ответы на вопросы снабжены метками, обозначающими расположение соответствующего отрывка в тексте, то есть порядковыми номерами первого и последнего символа ответа.

Вопросы имеют показатели релевантности теме текста, они могут соответствовать или не соответствовать теме текста. Нерелевантные вопросы составляют 5% выборки и включают типовые запросы, которые пользователь может вводить в систему для развлечения. Наша идея заключается в том, что модель должна научиться игнорировать интерактивные вопросы. Диалоговая система для поддержки инклюзии должна информировать, а не развлекать пользователя.

Описанный набор данных для машинного обучения составлен нами самостоятельно. На поздних этапах разработки мы привлекли краудсорсинг. Исполнителям было поручено задавать вопросы к тексту, как релевантные, так и нерелевантные. Для составления релевантных вопросов исполнителям следовало ознакомиться с текстами для машинного чтения. Для нерелевантных вопросов им предлагалось вдохновиться списком тем, популярных среди пользователей чат-ботов. Список тем мы вывели на основе личного опыта разработки и взаимодействия с чат-ботами, руководствуясь опытом коллег, блоггеров в сфере IT и ученых.

В ходе реализации подхода к машинному обучению, в основе которого лежит работа с данными, мы провели несколько экспериментов по изменению структуры набора. Мы увеличили количество ответов, уменьшили их длину, изменили число вопросно-ответных пар и дизайн набора (удалили нерелевантные вопросы). В результате мы выяснили, что использование более простой структуры (например, набора без нерелевантных вопросов), увеличение средней длины ответа и количества вопросно-ответных пар приводят к наилучшим результатам.

Использование простой структуры набора данных несколько упрощает задачу, что, с одной стороны, ограничивает возможности модели машинного обучения, а с другой — делает выдачи системы более предсказуемыми. Поскольку при разработке диалоговых систем для поддержки инклюзии важна безопасность модели, проявляющаяся в том числе в предсказуемости ее выдач, мы решили остановиться на выборе в пользу простых структур.

Увеличение средней длины ответа также облегчает задачу, поскольку обеспечивает более высокую вероятность случайного угадывания. Увеличение количества вопросно-ответных пар позволяет модели проанализировать большее количество связей и признаков, а также охватить большее количество тематических кластеров. Это обеспечивает более высокую гибкость модели, делает систему устойчивой к неожиданным вопросам пользователя.

Основанием для наших выводов по экспериментам с реализацией различных подходов к построению диалоговых систем послужили результаты автоматической и ручной оценки интеллектуальных моделей, а также итоги лингвистического анализа материала исследования. Автоматическая оценка систем производилась с применением распространенных в машинном обучении метрик — полноты, точности и F1-меры. Ручная оценка проводилась по методике, разработанной нами для данного исследования. Фокус-группе было предложено оценить набор случайных выдач обученных моделей по пятибалльной шкале и обосновать свое решение. Лингвистический анализ состоял в выделении языковых признаков, которые оказали наибольшее влияние на качество выдач моделей в соответствии с результатами ручной и автоматической оценки.

Основным выводом этой части работы стало выявление необходимости ручной оценки систем для поддержки инклюзии, поскольку результаты автоматической оценки в нашем эксперименте не совпадали с мнением фокус-группы. Значения метрик автоматической оценки показали, что наиболее производительным типом систем оказались генеративные вопросно-ответные модели. Однако респонденты выявили значительное количество отклонений от темы и высокую вероятность генерации ложных фактов в работе таких моделей. По результатам ручной оценки и лингвистического анализа наиболее надежным типом систем были признаны экстрактивные модели.

Стоит принять во внимание тот факт, что ручная оценка представляет собой ресурснозатратный процесс, требующий вычитки выходных данных моделей, поиска респондентов, составления и анализа анкет. Мы пришли к выводу о том, что гибридный подход будет оптимальным решением проблемы оценки диалоговых систем для поддержки инклюзии.

Предлагаемый гибридный подход состоит из двух этапов. На первом этапе мы оцениваем исследуемую выборку модификаций моделей и архитектур автоматически и выбираем наиболее производительные варианты. На втором этапе мы генерируем набор выдач с использованием каждой из выбранных на первом этапе моделей и предлагаем фокус-группе оценить несколько случайных выдач.

Мы не проводили исследование оптимального количества образцов выдач для ручной оценки, так как в центре нашего внимания было исследование подходов к разработке моделей, а не их отладки или оценки (для этого необходимо проведение новых полноценных исследований). Мы предполагаем, что для ручной оценки достаточно составить от 10 до 20 выдач на каждый тип модели. При использовании большего количества выдач существует риск того, что респонденты испугаются объема текстов, которые им предстоит оценить. В этом случае вместо оценки системы мы можем получить результаты случайного угадывания. В данном исследовании мы ограничились десятью образцами на каждую модель и получили активную обратную связь — респонденты пожелали продолжить обсуждение результатов работы в личной переписке с автором исследования.

Целью нашей работы было составление методологических рекомендаций для разработчиков систем для поддержки инклюзии людей с расстройствами аутистического спектра. Нам удалось достичь этой цели. Со списком рекомендаций можно ознакомиться в Приложении Г «Методологические рекомендации».

Методологические рекомендации опираются на выводы, описанные нами в конце каждого раздела практической части нашей работы. При составлении рекомендаций мы сформировали три тематических блока, каждый из которых описывает один из базовых этапов разработки диалоговых систем: составление набора данных, непосредственно разработка и оценка системы. Это распространенные этапы разработки систем искусственного интеллекта многих типов, однако наши рекомендации касаются только создания диалоговых систем для поддержки инклюзии.

Рекомендации затрагивают технические аспекты разработки таких систем и включают советы по грамотному распределению ресурсов (время, человеческий ресурс, энергоемкость). Также мы обращаем внимание на необходимость дополнительных консультаций со специалистами (например, психиатрами) и представителями целевой аудитории (например, людьми с аутизмом и их родителями). Кроме того, мы перечисляем лингвистические особенности, которые следует учитывать при разработке систем. Рассмотрим список рекомендаций подробнее.

Рекомендации, касающиеся процесса составления набора данных, включают пять пунктов, которые описывают предпочтительные особенности структуры и наполнения набора. Составляя рекомендации по разработке структуры, мы учитывали такие особенности, как количество ответов на один вопрос, средняя длина ответа, наличие и разнообразие специальных меток в наборе (дизайн набора) и его размер (количество вопросно-ответных пар). При составлении лингвистических рекомендаций мы учитывали влияние средств интенсификации, перечислений и сочинительных конструкций на производительность моделей.

Рекомендации включают три пункта о выборе архитектуры и настройке моделей. Мы комментируем выбор предобученных моделей с точки зрения влияния языковых особенностей данных для предварительного обучения на производительность системы при решении последующих задач после тонкой настройки. Также мы освещаем проблему работы с энергоемкими моделями и особенности выбора подходящей архитектуры с технической точки зрения. Например, мы описываем преимущества и недостатки одно- и двунаправленных моделей на основе собственного опыта, полученного в ходе нашей работы.

Советы по оценке систем включают два пункта. В первом мы предупреждаем разработчиков диалоговых систем о недостатках использования исключительно автоматических методов оценки. Мы рекомендуем больше внимания уделять ручной оценке таких систем, учитывать то обстоятельство, что недоработанные модели могут порождать ложные факты и создавать прецеденты для неблагоприятных ситуаций в инклюзивном сообществе. Во втором пункте мы советуем разработчикам искать методы сочетания ручной и автоматической оценки для экономии ресурсов при создании интеллектуальных систем.

В начале исследования мы выдвинули ряд гипотез о результатах работы наших систем. Мы размышляли о том, какие из наших выводов могут повлиять на состав методологических рекомендаций. Большое значение в нашей работе уделено исследованию преимуществ и недостатков генеративного и экстрактивного подходов к построению диалоговых систем.

Одна из гипотез заключалась в том, что генеративный подход может представлять опасность при разработке систем для поддержки инклюзии. Гипотеза основывалась на знаниях о том, что значительное влияние на работу генеративных алгоритмов имеет фактор случайной генерации токенов. Мы предположили, что эта особенность генеративных систем может привести к тому, что ответы на вопросы пользователей, сгенерированные моделями будут тематически отклоняться от пользовательского запроса. Гипотеза подтвердилась. Результаты ручной оценки генеративных моделей и комментарии респондентов-оценщиков показали, что такие системы часто порождают связные и грамматические корректные тексты, которые абсолютно не связаны с темами, обозначенными пользователями в их запросах.

Поэтому мы считаем использование генеративных систем небезопасным в инклюзивной среде. Однако в перспективе мы планируем провести исследование методов контроля таких моделей. На данный момент мы предполагаем, что построение тематических кластеров может стать одним из решений проблемы генеративных алгоритмов. Контролируемая генеративная система сначала будет распознавать тему из вопроса пользователя и только потом генерировать ответ на вопрос, используя веса модификаций, дообученных на материале конкретных тем.

Следующая гипотеза связана с работой экстрактивных систем. Мы предположили, что экстрактивные модели позволят достичь наивысших значений метрик автоматической оценки производительности, однако это не подтвердилось. Несмотря на то, что такие модели учатся извлекать точные цитаты из текста, что, как кажется, должно обеспечивать высокий процент угадывания ответов на вопросы пользователя, в случаях, когда экстрактивным моделям не удается найти подходящий ответ, они генерируют пустую выдачу или пару токенов без информационной нагрузки (например, частотные биграммы, такие как *«по»*).

Тем не менее принцип работы таких систем обеспечивает надежность и высокую точность интерпретации текстов из обучающей выборки моделью, что подтверждается результатами ручной оценки. Наши респонденты присваивали оценки «4» и «5» по пятибалльной шкале экстрактивным системам значительно чаще, чем их генеративным аналогам.

При этом экстрактивные системы требуют использования более сложных по структуре наборов данных, которые должны включать не только вопросно-ответные пары, но и контексты, а также метки, указывающие расположение ответа на вопрос в соответствующих отрывках текстов. Проблему составления набора данных удалось решить с помощью краудсорсинга.

Третья гипотеза касалась подхода к построению систем машинного обучения, в основе которого лежит работа с данными. Мы предполагали, что преобразование структуры и дизайна набора в большей степени повлияет на работу экстрактивных, нежели генеративных систем. Результаты лингвистического анализа материала исследования показали, что языковой состав и структура данных, на которых обучается модель, в равной степени влияют на производительность и особенности выдач систем вне зависимости от архитектуры. Гипотеза не подтвердилась.

Мы пришли к выводу о том, что при создании языковых моделей работе с данными следует уделять больше внимания, чем настройке алгоритма. Это касается преобразований обучающей выборки, увеличения объема наборов данных, экспериментов с составом метаданных и других опытов. Эта мысль созвучна последним идеям Эндрю Ына, который в 2021 году предложил в глобальном научном обществе проводить новые семинары и соревнования по искусственному интеллекту, в основе которого лежит работа с данными.

Мы размышляли над тем, как достичь баланса между игровой формой подачи материала и качественным информированием пользователя. Чтобы разобраться в этом вопросе, мы взяли несколько интервью у представителей некоммерческих организаций, таких как «Перспективы» (Санкт-Петербург), «Семейная усадьба» и «Равные возможности» (Иркутск), а также у представителей сообществ, которые занимаются защитой и обеспечением равных прав для различных категорий населения.

В результате мы получили более глубокое представление о том, с какими проблемами сталкиваются люди с расстройствами аутистического спектра и их близкие в повседневной жизни. Мы постарались понять, как может воспринимать мир человек в спектре аутизма. Нам удалось сформировать приблизительный список тем, которые в будущем следует осветить в наборе данных и структурировать доступный материал. Мы планируем продолжить работу над нашим проектом для поддержки людей в спектре аутизма в следующем году.

Как обеспечить качественную подачу такого сложного для интерпретации материала и при этом разработать систему, интересную для разных целевых аудиторий? Можно предположить, что экстрактивные алгоритмы должны работать в ансамбле с генеративными моделями. Генеративные языковые модели проявили себя как ненадежные, «склонные» к генерации ложных фактов, однако мы считаем, что модели для генерации изображений могут использоваться для иллюстрирования выдач диалоговых систем. Если целевой аудиторией нашей диалоговой системы станут люди в спектре, сопровождение текстов иллюстрациями будет важным элементом обеспечения доступности информации. Кроме того, иллюстрированный материал найдет отклик в сердцах молодых нейротипичных людей, которые сегодня с большим удовольствием проводят время в социальных сетях с визуальным контентом.

В заключение подчеркнем, что основной вопрос, которым мы задавались в начале исследования — как разработать систему, которая не принесет вреда — остается открытым для дискуссии. Сегодня искусственный интеллект, как наука, индустриальное течение и область философии, неразрывно связан с такими понятиями, как этика, открытость данных и борьба с ложной информацией.

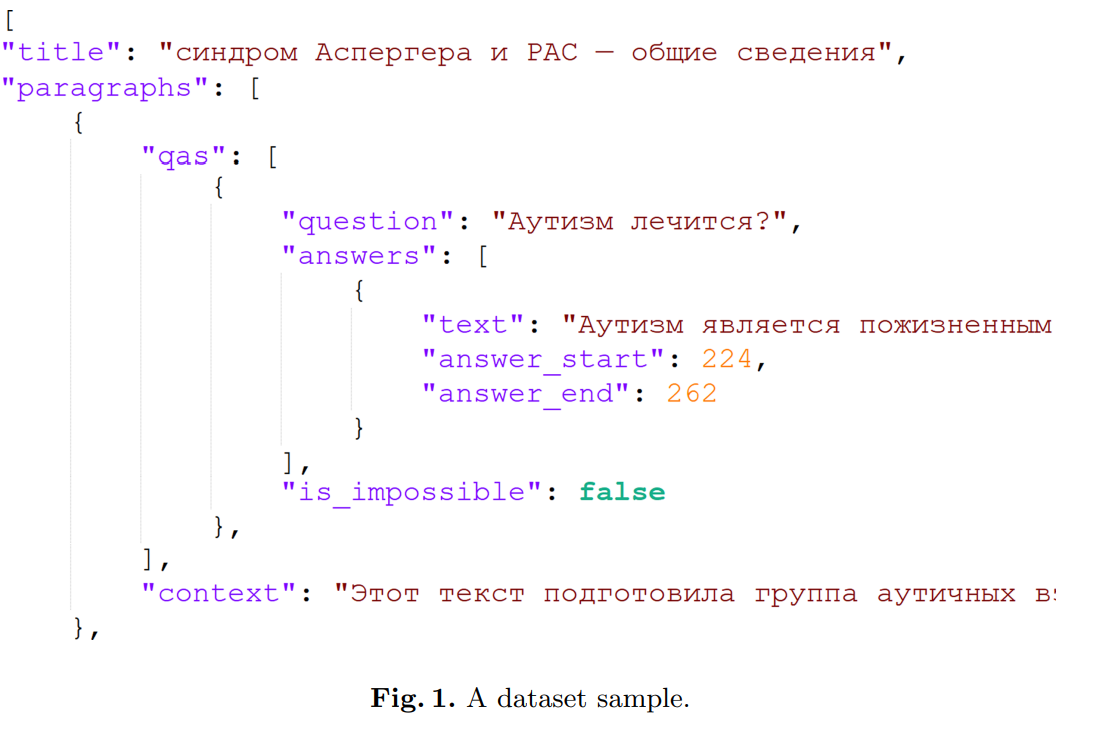
Мы верим, что средства искусственного интеллекта действительно могут послужить кирпичиком для выстраивания равного, инклюзивного общества, однако следует помнить, что современные интеллектуальные системы далеки от совершенства. Они плохо поддаются контролю, порой генерируют пугающие выдачи и могут интерпретировать материал, провоцирующий и усиливающий социальное неравенство. Тем не менее в настоящей работе мы постарались рассмотреть некоторые возможности применения таких систем для поддержки инклюзии и получили обратную связь, которая позволила убедиться в том, что проведенная работа не была напрасна.

# Список источников

1. Академия Яндекса [Электронный ресурс] / Как устроена работа голосовых помощников. URL: https://academy.yandex.ru/posts/kak-ustroena-rabota-golosovykh-pomoschnikov (дата обращения: 10.05.2022)
2. Аль-Кайси А. Н., Архангельская А. Л., Руденко-Моргун О. И. Интеллектуальный голосовой помощник Алиса на уроках русского языка как иностранного (уровень А1) [Текст] //Филологические науки. Вопросы теории и практики. – 2019. – Т. 12. – №. 2.
3. Беломутов, Г. В. Разработка навыка на платформе Яндекс. Диалоги для управления устройствами умного дома [Текст] / Г. В. Беломутов, И. А. Мурашко // Научные Горизонты. – 2020. – № 5. – С. 255–259.
4. Белоусов А. И., Ткачев С. Б. Дискретная математика [Текст]. – 2015.
5. Блог Яндекса [Электронный ресурс] / Как победить морников: Яндекс запустил гибридную систему перевода. URL: https://yandex.ru/blog/company/kak-pobedit-mornikov-yandeks-zapustil-gibridnuyu-sistemu-perevoda (дата обращения: 12.05.2022)
6. Виртуальные ассистенты Салют [Электронный ресурс] / Алиса – голосовой помощник от компании Яндекс. URL: https://yandex.ru/alice?ysclid=l30i4o5z68 (дата обращения: 10.05.2022)
7. Дубельщиков А. А., Тутова Н. В. Навыки Яндекс. Алиса: от идеи до реализации [Текст] //Телекоммуникации и информационные технологии. – 2020. – Т. 7. – №. 2. – С. 92-97.
8. Информационный документ, Всемирный экономический форум, 2019. [Электронный ресурс] / Четвертая промышленная революция Целевые ориентиры развития промышленных технологий и инноваций. URL: https://www3.weforum.org/docs/WEF\_%D0%A7%D0%B5%D1%82%D0%B2%D0%B5%D1%80%D1%82%D0%B0%D1%8F\_%D0%BF%D1%80%D0%BE%D0%BC%D1%8B%D1%88%D0%BB%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B5%D0%B2%D0%BE%D0%BB%D1%8E%D1%86%D0%B8%D1%8F.pdf (дата обращения: 10.05.2022)
9. Коротков И. А., Шамурина А. И. Некоторые методы увеличения эффективности сверточной нейронной сети [Текст] //Информационные технологии XXI века. – 2020. – С. 58-65.
10. Маруся — голосовой помощник от VK [Электронный ресурс] / Mail.ru. URL: https://marusia.mail.ru/ (дата обращения: 10.05.2022)
11. Маруся | Быстрый старт | VK для разработчиков [Электронный ресурс] / Создание скилла Маруси. URL: https://vk.com/dev/marusia\_skill\_docs (дата обращения: 10.05.2022)
12. Ратников М. О., Чафонова А. Г. Лингвометодический потенциал виртуальных голосовых помощников (на примере системы «Алиса» компании «Яндекс») [Текст] //Вестник Воронежского государственного университета. Серия: Проблемы высшего образования. – 2019. – №. 4. – С. 64-66.
13. СберДевайсы — СберБанк [Электронный ресурс] / SberBank. URL: https://salute.sber.ru/ (дата обращения: 10.05.2022)
14. Яндекс.Авто [Электронный ресурс] / Яндекс.Авто — бортовой компьютер. URL: https://auto.yandex (дата обращения: 10.05.2022)
15. Adiwardana D. et al. Towards a human-like open-domain chatbot [Текст] //arXiv preprint arXiv:2001.09977. – 2020.
16. Autism Spectrum Disorder QA [Электронный ресурс] / Open Data Science. URL: https://ods.ai/projects/asd\_qa?ysclid=l30ie5l9l5 (дата обращения: 10.05.2022)
17. Autistic City [Электронный ресурс] / Проект поддержки людей с синдромом Аспергера и аутизмом. URL: https://www.aspergers.ru (дата обращения: 10.05.2022)
18. Beveridge M., Milward D. Combining task descriptions and ontological knowledge for adaptive dialogue [Текст] //International Conference on Text, Speech and Dialogue. – Springer, Berlin, Heidelberg, 2003. – С. 341-348.
19. Bickmore T. W., Giorgino T. Some Novel Aspects of Health Communication from a Dialogue Systems Perspective [Текст] //AAAI Technical Report (4). – 2004. – С. 5-12.
20. Blinov P. et al. RuMedBench: A Russian Medical Language Understanding Benchmark [Текст] //arXiv e-prints. – 2022. – С. arXiv: 2201.06499.
21. Brown T. et al. Language models are few-shot learners [Текст] //Advances in neural information processing systems. – 2020. – Т. 33. – С. 1877-1901.
22. Caswell I., Liang B. Recent advances in Google Translate [Текст] //Google AI Blog: The latest from Google Research. – 2020.
23. Chen M. et al. Generative pretraining from pixels [Текст] //International Conference on Machine Learning. – PMLR, 2020. – С. 1691-1703.
24. Cobbe K. et al. Training verifiers to solve math word problems [Текст] //arXiv preprint arXiv:2110.14168. – 2021.
25. Compton R. et al. MEDCOD: A Medically-Accurate, Emotive, Diverse, and Controllable Dialog System [Текст] //Machine Learning for Health. – PMLR, 2021. – С. 110-129.
26. Conneau A. et al. Unsupervised Cross-lingual Representation Learning at Scale [Текст] //Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. – 2020. – С. 8440-8451.
27. Cooper A., Ireland D. Designing a chat-bot for non-verbal children on the autism spectrum [Текст] //Stud Health Technol Inform. – 2018. – Т. 252. – С. 63-68.
28. Core techniques of question answering systems over knowledge bases: a survey [Текст] //Knowledge and Information systems. – 2018. – Т. 55. – №. 3. – С. 529-569.
29. Danilov G. et al. Detection of muscle weakness in medical texts using natural language processing [Текст] //Digital Personalized Health and Medicine. – IOS press, 2020. – С. 163-167.
30. Data Ёлка 2020 [Электронный ресурс] / ODS Best Project Award. YouTube. URL: https://www.youtube.com/watch?v=TPDMRitnRXI&list=PLTlO6nV\_TaGAZu5OlPdY6hEIZn0xhVIZ6&index=10 (дата обращения: 12.05.2022)
31. Devakunchari R., Rishabh A., Eshita A. A Survey of Chatbot Design Techniques [Текст] // International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 8958. – 2019. – T. 8. – №. 2S2.
32. Divya S. et al. A self-diagnosis medical chatbot using artificial intelligence [Текст] //Journal of Web Development and Web Designing. – 2018. – Т. 3. – №. 1. – С. 1-7.
33. FAQ Bot [Электронный ресурс] / Chit-chat in FAQ Bot. URL: https://www.faqbot.ai/post/chit-chat-in-faq-bot (дата обращения: 12.05.2022)
34. Firsanova V. Question Answering Systems and Inclusion: Pros and Cons [Текст] //«Интернет и современное общество»(Internet and Modern Society–IMS): Международная конференция. – RWTH Aahen University, 2021 (B).
35. Firsanova V. The Advantages of Human Evaluation of Sociomedical Question Answering Systems [Текст] //International Journal of Open Information Technologies. – 2021. – Т. 9. – №. 12. – С. 53-59 (D).
36. Firsanova V. The Description of The Autism Spectrum Disorder Question Answering Dataset [Текст] // Материалы студенческой сессии международной конференции Диалог 2021 (C).
37. Firsanova V. Transformer Models for Question Answering on Autism Spectrum Disorder QA Dataset [Текст] //International Conference on Digital Transformation and Global Society. – Springer, Cham, 2021. – С. 122-133.
38. Firsanova V. Two Approaches to Building Dialogue Systems for People on the Spectrum [Текст] //Conference on Neural Information Processing Systems. – 2021 (А).
39. Firsanova, V. Autism Spectrum Disorder and Asperger Syndrome Question Answering Dataset. – 2020 [Электронный ресурс] / FigShare. Dataset. URL: https://doi.org/10.6084/m9.figshare.13295831 (дата обращения: 12.05.2022)
40. Firsanova, V. Supporting the Inclusion of People with Asperger Syndrome: Building a Customizable Chatbot with Transfer Learning [Текст] //AINL: Artificial Intelligence and Natural Language Conference. Workshop on Human-AI Interaction. 7. 2021 (E).
41. Gage P. A new algorithm for data compression [Текст] //C Users Journal. – 1994. – Т. 12. – №. 2. – С. 23-38.
42. Gao J. et al. Neural approaches to conversational AI [Текст] //Foundations and trends® in information retrieval. – 2019. – Т. 13. – №. 2-3. – С. 127-298.
43. GitHub [Электронный ресурс] / GitHub - vifirsanova ASD-QA. URL: https://github.com/vifirsanova/ASD-QA (дата обращения: 12.05.2022)
44. GPT-3 Demo [Электронный ресурс] / GPT-3 Demo: 300+ GPT-3 Examples, Demos, Apps. URL: https://gpt3demo.com/ (дата обращения: 12.05.2022)
45. Guinn C. I., Hubal R. C. Augmented transition networks (ATNs) for dialog control: A longitudinal study [Текст] //Computational Intelligence. – 2006. – С. 395-400.
46. Hendrycks D., Gimpel K. Gaussian error linear units (gelus) [Текст] //arXiv preprint arXiv:1606.08415. – 2016.
47. Homma T., Atlas L. E., Marks R. J. An artificial neural network for spatiotemporal: application to phoneme classification [Текст] //Proceedings of the 1987 International Conference on Neural Information Processing Systems. – 1987. – С. 31-40.
48. Hua M., Raley R. Playing With Unicorns: AI Dungeon and Citizen NLP [Текст] //DHQ: Digital Humanities Quarterly. – 2020. – Т. 14. – №. 4.
49. Kenton J. D. M. W. C., Toutanova L. K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Текст] //Proceedings of NAACL-HLT. – 2019. – С. 4171-4186.
50. Keselj V. Speech and Language Processing [Текст]. Daniel Jurafsky and James H. Martin (Stanford University and University of Colorado at Boulder) Pearson Prentice Hall, 2009, xxxi+ 988 pp; hardbound, ISBN 978-0-13-187321-6. – 2009.
51. Lewis M. et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension [Текст] //Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. – 2020. – С. 7871-7880.
52. Li X. et al. A General Chinese Chatbot based on Deep Learning and Its’ Application for Children with ASD [Текст] //International Journal of Machine Learning and Computing. – 2020. – Т. 10. – №. 4. – С. 1-10.
53. Madhu D. et al. A novel approach for medical assistance using trained chatbot [Текст] //2017 international conference on inventive communication and computational technologies (ICICCT). – IEEE, 2017. – С. 243-246.
54. Microsoft [Электронный ресурс] / Документация по QnA Maker. URL: https://docs.microsoft.com/ru-ru/azure/cognitive-services/qnamaker/
55. Models - Hugging Face [Электронный ресурс] / Hugging Face. URL: https://huggingface.co/models (дата обращения: 12.05.2022)
56. Möller T. et al. COVID-QA: A Question Answering Dataset for COVID-19 [Текст] //Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020. – 2020.
57. OpenAI API [Электронный ресурс] / OpenAI's API provides access to GPT-3. URL: https://openai.com/api/ (дата обращения: 12.05.2022)
58. Pendergrass A. et al. Inclusive scientific meetings: Where to start [Текст] //500 Women Scientists. – 2019.
59. Radford A. et al. Improving language understanding by generative pre-training [Текст] //Open AI. – 2018.
60. Radford A. et al. Language models are unsupervised multitask learners [Текст] //OpenAI blog. – 2019. – Т. 1. – №. 8. – С. 9.
61. Raffel C. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer [Текст] //Journal of Machine Learning Research. – 2020. – Т. 21. – С. 1-67.
62. Rajpurkar P. et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text [Текст] //Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. – 2016. – С. 2383-2392.
63. Rajpurkar P., Jia R., Liang P. Know What You Don’t Know: Unanswerable Questions for SQuAD [Текст] //Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). – 2018. – С. 784-789.
64. Ramachandran P., Zoph B., Le Q. V. Searching for activation functions [Текст] //arXiv preprint arXiv:1710.05941. – 2017.
65. Ruder S. et al. Transfer learning in natural language processing [Текст] //Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials. – 2019. – С. 15-18.
66. Sanh V. et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter [Текст] //arXiv preprint arXiv:1910.01108. – 2019.
67. Searle J. R. Minds, brains, and programs [Текст] //Behavioral and brain sciences. – 1980. – Т. 3. – №. 3. – С. 417-424.
68. Sennrich R., Haddow B., Birch A. Neural Machine Translation of Rare Words with Subword Units [Текст] //Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). – 2016. – С. 1715-1725.
69. SmartMarket [Электронный ресурс] / RuGPT-3 — AI-модель для написания текстов. URL: https://developers.sber.ru/portal/products/rugpt-3?ysclid=l339daa39&attempt=1 (дата обращения: 12.05.2022)
70. Soufyane A., Abdelhakim B. A., Ahmed M. B. An intelligent chatbot using NLP and TF-IDF algorithm for text understanding applied to the medical field [Текст] //Emerging Trends in ICT for Sustainable Development. – Springer, Cham, 2021. – С. 3-10.
71. SQuAD Explorer [Электронный ресурс] / The Stanford Question Answering Dataset. URL: https://rajpurkar.github.io › SQuAD-explorer (дата обращения: 12.05.2022)
72. Su H. et al. Moviechats: Chat like humans in a closed domain [Текст] //Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). – 2020. – С. 6605-6619.
73. Sutskever I. – 2019 [Электронный ресурс] / GPT-2. Matroid. Presented at the Matroid Scaled Machine Learning Conference 2019. YouTube. URL: https://www.youtube.com/watch?v=T0I88NhR\_9M&ab\_channel=Matroid (дата обращения: 12.05.2022)
74. Sutskever I., Vinyals O., Le Q. V. Sequence to sequence learning with neural networks [Текст] //Advances in neural information processing systems. – 2014. – Т. 27. – С. 3104-3112.
75. Vaswani A. et al. Attention is all you need [Текст] //Advances in neural information processing systems. – 2017. – Т. 30. – С. 6000–6010.
76. Vatian A. et al. Comparative analysis of approaches to building medical dialog systems in Russian [Текст] //International Conference on Intelligent Data Engineering and Automated Learning. – Springer, Cham, 2019. – С. 175-183.
77. Wallace R. S. The anatomy of ALICE In: Parsing the Turing Test [Текст]. R. Epstein, G. Roberts, and G. Beber. – 2009. – С. 181–210.
78. Wang L. L. et al. CORD-19: The COVID-19 Open Research Dataset [Текст] //Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020. – 2020.
79. Weizenbaum J. Eliza—a computer program for the study of natural language communication between man and machine [Текст] //Communications of the ACM. – 1983. – Т. 26. – №. 1. – С. 23-28.
80. Winograd T. Understanding natural language [Текст] //Cognitive psychology. – 1972. – Т. 3. – №. 1. – С. 1-191.
81. Wolf T. et al. Transformers: State-of-the-art natural language processing [Текст] //Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. – 2020. – С. 38-45.
82. Woods W. A. Progress in natural language understanding: an application to lunar geology [Текст] //Proceedings of the June 4-8, 1973, national computer conference and exposition. – 1973. – С. 441-450.
83. Woods W. A. Transition network grammars for natural language analysis [Текст] //Communications of the ACM. – 1970. – Т. 13. – №. 10. – С. 591-606.CHOMSKY, N. Aspects of the Theory of Syntaz. MIT Press, Cambridge, Mass., 1965.
84. Woolf M. How to make custom AI-Generated text with GPT-2 [Текст] //Max Woolf’s Blog. – 2019.
85. Xu P. et al. Attention-guided Generative Models for Extractive Question Answering [Текст] //arXiv e-prints. – 2021. – С. arXiv: 2110.06393.
86. Yoon K. Convolutional Neural Networks for Sentence Classification [Текст] //Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). – 2014. – С. 1746-1751.
87. Zhang Y. et al. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation [Текст] //Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. – 2020. – С. 270-278. Diefenbach D. et al.

# Приложения

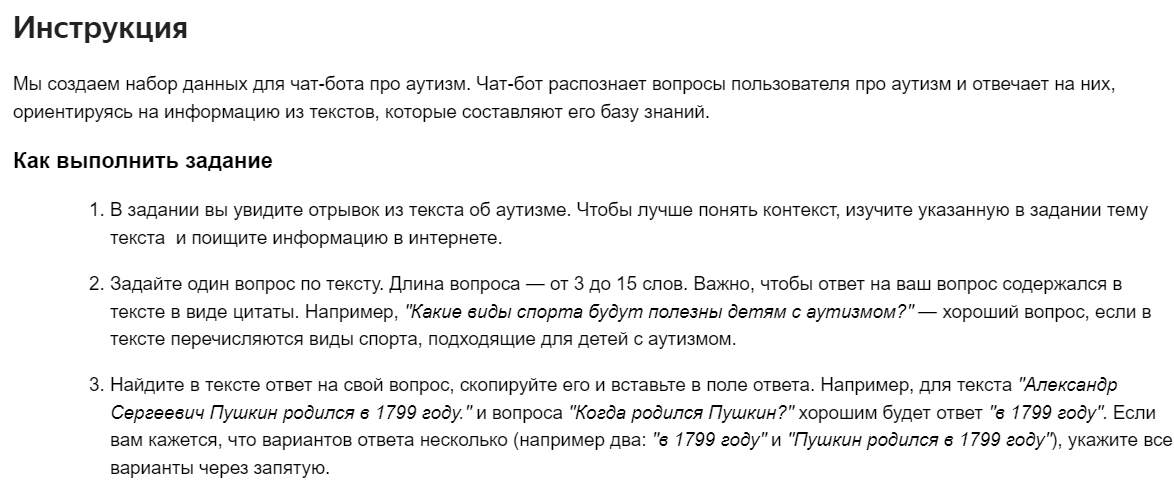
# А. Образец набора данных



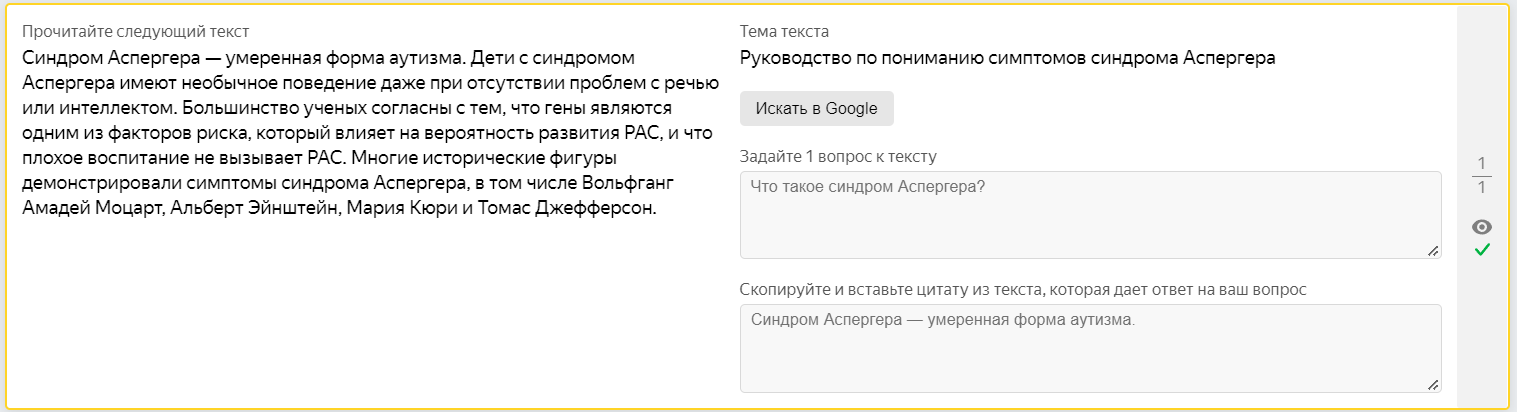
# Б. Тексты заданий для краудсорсинга

## 1. Задание на составление вопросов по тексту

#### 1. 1. Инструкция на сервисе Яндекс.Толока

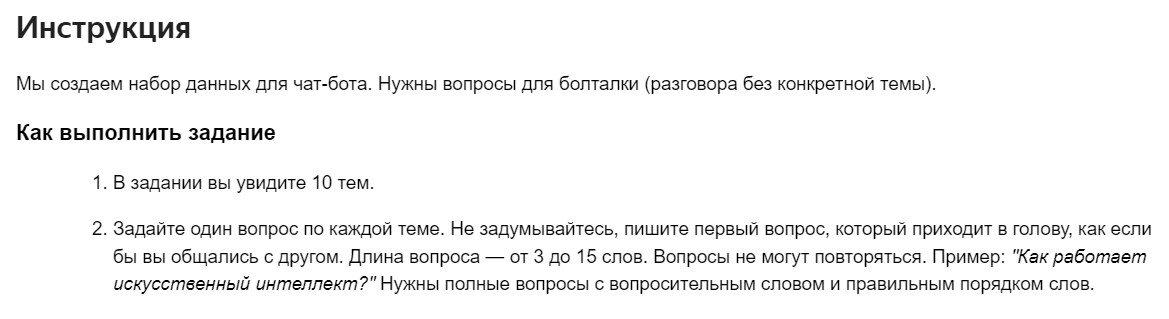


#### 1.2. Иллюстрация интерфейса с решенным заданием

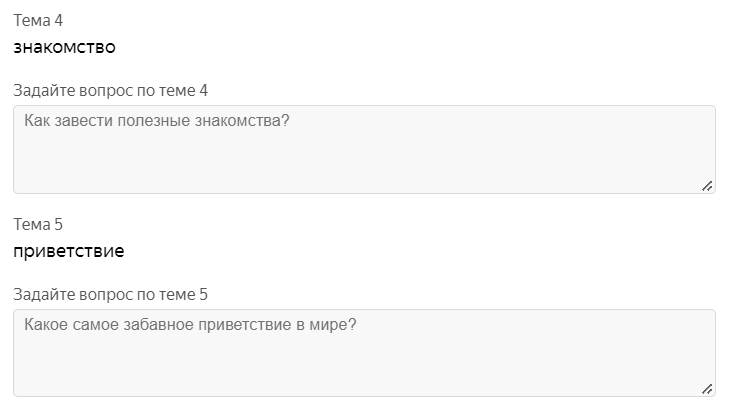
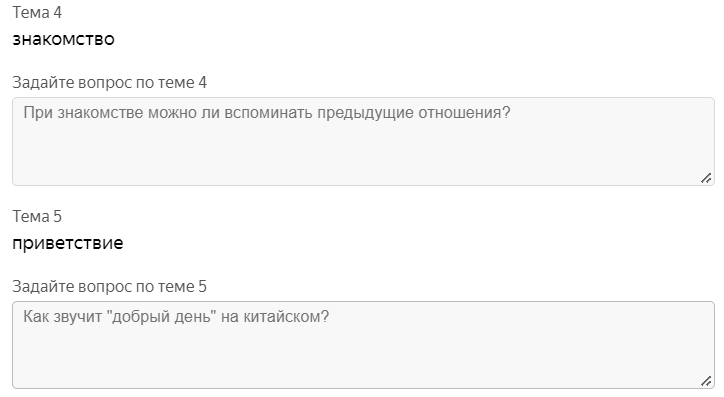


## 2. Задание на составление нерелевантных вопросов

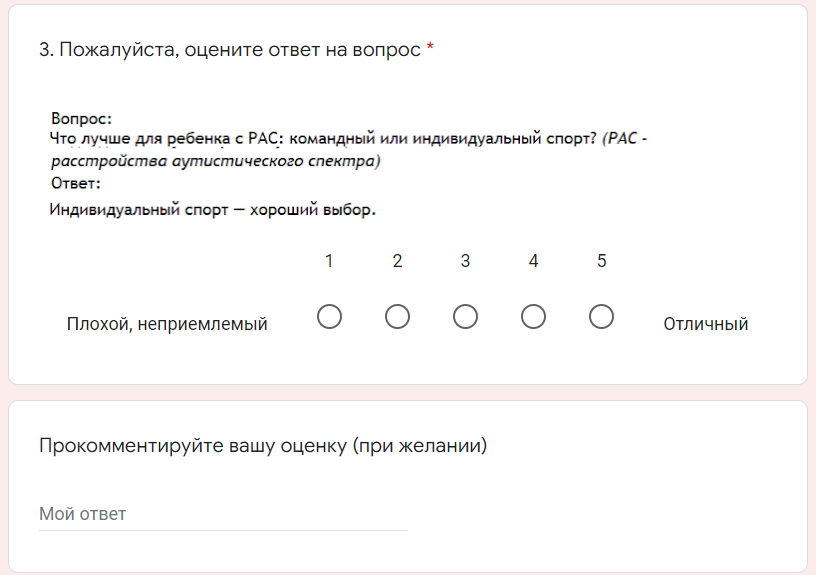
#### 2. 1. Инструкция на сервисе Яндекс.Толока



#### 2. 2. Иллюстрация интерфейса с решенным заданием

# В. Образец анкеты для ручной оценки



# Г. Методологические рекомендации

## Рекомендации по составлению набора данных

### Одному вопросу должен соответствовать один ответ

Практика показывает, что наличие нескольких вариантов ответа на один вопрос в обучающей выборке, не улучшает показатели метрик оценки интеллектуальных моделей. Время, которое вы могли бы впустую потратить на составление нескольких вариантов ответа, лучше использовать на пополнение набора данных новыми вопросно-ответными парами.

### Не стоит жестко ограничивать длину ответа

Длина ответа должна составлять, в идеале, не менее 10-15 слов. Интеллектуальным моделям проще угадывать более длинные последовательности. Пополняя обучающую выборку длинными ответами, вы, с одной стороны, упрощаете задачу, которую должна решить модель, с другой стороны, обеспечиваете более высокую производительность и, как следствие, безопасность, если речь идет о системах для поддержки инклюзии.

### Отдавайте предпочтение наборам данных с простой структурой

Диалоговые системы для поддержки инклюзии предполагают работу с малоресурсными и сложными для обработки доменами, такими как «аутизм» или «доступная среда для людей с инвалидностью». Если, вы хотите построить более производительную и при этом безопасную для пользователей систему, выбирайте наборы данных для решения простых задач. Сложные структуры (например, наличие в наборе данных специальных «шумов», которые модель должна учиться игнорировать) открывают новые возможности для вашей диалоговой системы, однако при разработке таких моделей будьте готовы к низким значениям метрик оценки и высоким рискам.

### Размер обучающего набора прямо пропорционален производительности модели

При составлении материалов для обучающего набора постарайтесь охватить как можно больше аспектов выбранной предметной области. Например, составляя набор советов для родителей ребенка с аутизмом, проконсультируйтесь с психиатрами и представителями вашей целевой аудитории (родителями детей в спектре аутизма). Узнайте, какие темы следует осветить. Ваш набор может пополниться текстами о видах спорта для детей с аутизмом, воспитании особенных детей, материалами о походе к врачу и другой информацией.

Проводить первые опыты на наборе данных, который находится в разработке, – нормально. Так вы сможете внести коррективы в структуру набора на первых этапах разработки и сэкономить ресурс на поздних.

Размер набора непосредственно влияет на производительность вашей системы. Чем больше данных получит модель в процессе обучения, тем больше языковых связей и признаков она сможет выделить, тем надежнее будет ее работа.

### Средства интенсификации, перечисления и сочинительные предложения – ваши друзья

Средства усиления, например, наречия и частицы с соответствующим значением («очень», «даже» и другие), предложения с перечислениями действий или явлений, а также сочинительные связи выступают для модели своего рода индикаторами структуры текстов. Современные модели машинного обучения учитывают при анализе частотные конструкции, знаки препинания и порядок слов в текстах. Эти особенности модели используют для того, чтобы научиться ориентироваться в материале. Чем больше структур сможет считать модель, тем выше будет ее производительность. Если вы используете краудсорсинг для составления набора данных, пропишите лингвистические требования для исполнителей.

## Рекомендации по разработке системы

### При выборе предварительно обученной модели обращайте внимание на язык

Если вы используете предобученные модели (трансферное обучение), выбирайте версии, адаптированные под ваш язык. Например, если вы работаете с BERT, моделью, первоначально подходящей для английского языка, попробуйте также настроить его модификации, дообученные на русскоязычном материале (например, ruBERT).

Не стоит недооценивать возможности многоязычных и кросс-лингвистических моделей. Как показывает практика, многие из них имеют аналогичный уровень производительности. Кроме того, среди многоязычных и кросс-лингвистических моделей часто можно найти робастные модификации, превосходящие по производительности модели, настроенные для одного конкретного языка. К таким моделям относится, например, XLM-RoBERTa.

### Обращайте внимание на размер модели

Не бойтесь использовать модели с большим количеством параметров, например, T-5 или OPT. Это энергоемкие модели, для которых требуется специальное оборудование, например, графический процессор GPU, однако вы можете использовать такие коллаборативные платформы, как Google Colab, Yandex DataSphere, AWS, Kaggle, DeepNote и другие облачные сервисы.

### Двунаправленные архитектуры и экстрактивные системы в приоритете

Как показывает практика, двунаправленные языковые модели (например, BERT), настроенные для извлечения, а не генерации ответа на вопрос, проявляют себя как более надежные и устойчивые, чем однонаправленные генеративные архитектуры (например, GPT). Несмотря на то, что разработка генеративной вопросно-ответной системы представляет собой менее трудоемкий процесс, так как не требует добавления новых слоев к нейронной сети или разработки специальных классификаторов, результаты работы таких моделей плохо предсказуемы. Это приводит к высоким рискам использования генеративных архитектур при разработке систем для поддержки инклюзии.

## Рекомендации по оценке системы

### Результаты автоматической оценки могут быть обманчивы

Следует помнить о том, что любая модель в первую очередь учится достигать определенных значений метрик оценки, а не решать задачу. В результатах машинного обучения имеет место случайное угадывание.

### Сочетайте ручную и автоматическую оценку

Ручная оценка системы представляет собой ресурснозатратный процесс вне зависимости от выбранной вами методики. Однако вы можете сочетать методы ручной и автоматической оценки. Например, если вы выбираете между несколькими архитектурами, используйте автоматическую оценку для настройки своих моделей, выберите несколько случайных выдач настроенных модификаций (оптимальное количество составляет от 10 до 20) и оцените их с привлечением фокус-группы.